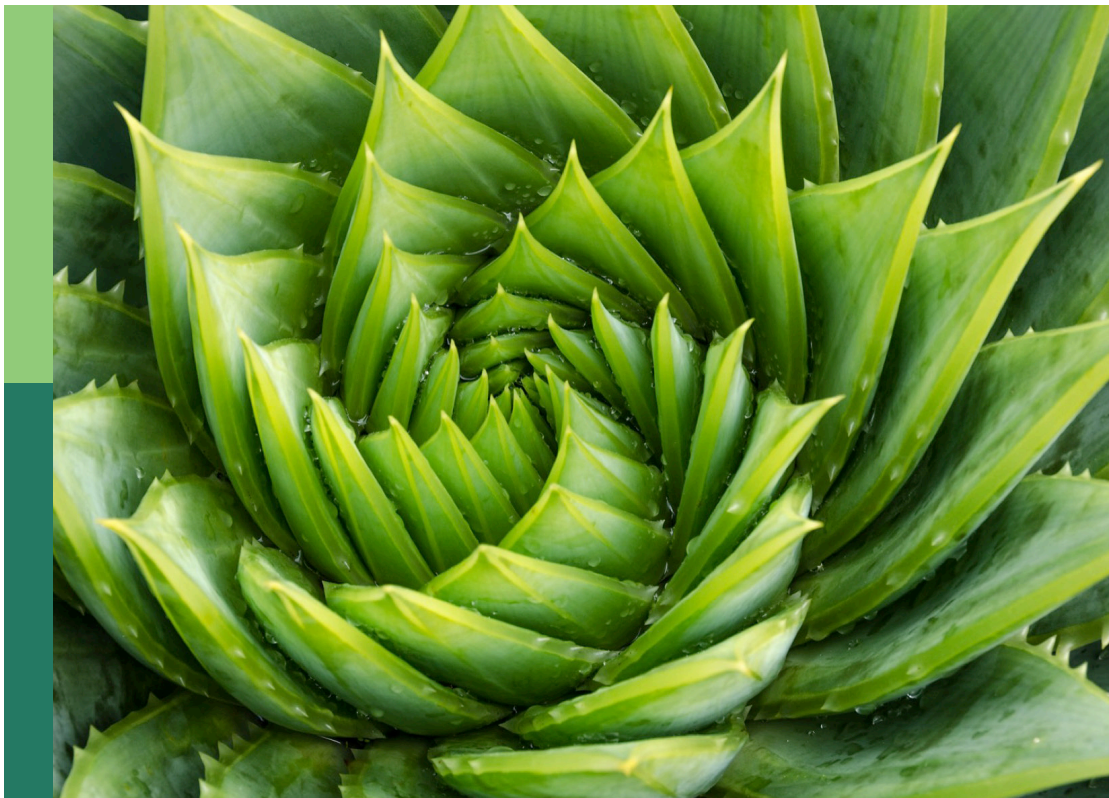
frontiers | Research Topics

IoT, UAV, BCI empowered deep learning models in precision agriculture

Edited by
José Dias Pereira, Liangliang Yang and Jian Lian

Coordinated by
Dengchao Feng

Published in
Frontiers in Plant Science



ISSN 1664-8714
ISBN 978-2-8325-4892-9
DOI 10.3389/978-2-8325-4892-9

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

IoT, UAV, BCI empowered deep learning models in precision agriculture

Topic editors
José Dias Pereira — Instituto Politecnico de Setubal (IPS), Portugal
Liangliang Yang — Kitami Institute of Technology, Japan
Jian Lian — Shandong Management University, China

Topic coordinator
Dengchao Feng — Shandong Police college, China

Citation
Dias Pereira, J., Yang, L., Lian, J., Feng, D., eds. (2024). *IoT, UAV, BCI empowered deep learning models in precision agriculture*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-4892-9

Table of contents

frontiers | Frontiers in Plant Science

TYPE Editorial
PUBLISHED 01 May 2024
DOI 10.3389/fpls.2024.1399753

OPEN ACCESS

EDITED AND REVIEWED BY
Roger Deal,
Emory University, United States

*CORRESPONDENCE
José Dias Pereira
dias.pereira@estsetubal.ips.pt

RECEIVED 12 March 2024
ACCEPTED 08 April 2024
PUBLISHED 01 May 2024

CITATION
Lian J and Dias Pereira J (2024) Editorial: IoT, UAV, BCI empowered deep learning models in precision agriculture.
Front. Plant Sci. 15:1399753.
doi: 10.3389/fpls.2024.1399753

Editorial: IoT, UAV, BCI empowered deep learning models in precision agriculture

Jian Lian[1] and José Dias Pereira[2]*

[1]School of Intelligence Engineering, Shandong Management University, Jinan, China, [2]Instituto Politécnico de Setúbal, Escola Superior de Tecnologia de Setúbal, Setúbal, Portugal

KEYWORDS

Internet of the Things, Unmanned Aerial Vehicle, deep learning, machine vision applications, precision agriculture, crop monitoring, plant disease recognition and classification

Editorial on the Research Topic
IoT, UAV, BCI empowered deep learning models in precision agriculture

Introduction

This Research Topic focuses the recent development in the Internet of Things and deep learning algorithms, including convolutional neural networks, transformer, and diffusion models, for precision agriculture in field and specialty crops. The 15 accepted papers include original research and review articles focusing novel deep learning algorithms, architectures, and applications of various instruments combined with the Internet of Things (IoT) and others advanced devices.

Research Topic coverage

We collected two reviews and thirteen research papers on the Research Topic focused by this Research Topic. The authors of the accepted publications presented articles that cover mainly of the following topics: deep learning models for precision agriculture; deep learning, BCI, and UAV-based crop monitoring; plant disease recognition and classification; UAV and deep learning for plant species detection and classification; deep learning and the BCI-empowered UAV applications for precision agriculture and optimization for deep learning algorithms in Precision Agriculture. There was a total of 37 submitted papers, 15 were accepted and 22 were rejected, that means an acceptance rate around 40%. This editorial discusses AI advancements in categorization, segmentation, detection, monitoring, and route planning that are influencing agriculture globally.

Image classification

Leaf disease classification needs improvement for precision agriculture applications. Han and Guo propose a new method for diagnosing leaf diseases in ligneous plants using an enhanced vision transformer model. The suggested method uses a multi-head attention

module to record pictures and class context. Additionally, the multi-layer perceptron module was used. The proposed deep model is trained using 22 types of ligneous leaf disease photos from a public dataset. The suggested model's training time is reduced via transfer learning. Identification of apple leaf diseases is critical for apple production. Propose a new attention strategy to help apple tree growers spot leaf diseases. Cheng and Li present a novel deep learning network based on MobileNet v3 and its methodology. Our network outperformed EfficientNet-B0, ResNet-34, and DenseNet-121 in recognizing apple leaf diseases with a remarkable accuracy of 98.7% on a private dataset. This model also outperforms existing models in accuracy, recall, and f1-score while keeping MobileNet's fewer parameters and computational efficiency. High-throughput crop monitoring using remotely recorded pictures and deep learning has improved crop health monitoring. Nampally et al. conduct studies on maize crops using various water treatments in a controlled setting. They capture crop data from tillering to heading using a multispectral camera on a UAV. A CNN model was presented with a flexible convolutional layer to learn and extract rich spatial and spectral characteristics. A weighted attention-based bi-directional long short-term memory network processes these features to deal with how they depend on time and order. Aggregated spatial-spectral-temporal Characteristics forecast water stress. To enable more efficient identification of plant diseases and pests, Guan et al. designed a novel network architecture based on EfficientNetV2. The experiments demonstrate that training this model using a dynamic learning rate decay strategy can improve the accuracy of plant disease and pest identification. Transfer learning is incorporated into the training process. After being trained using the dynamic learning rate decay strategy, the model achieves an accuracy of 99.80% on the Plant Village plant disease and pest dataset.

Image segmentation

Fine ripeness identification can improve strawberry harvest management by providing more precise crop information. Accordingly, Tang et al. offer a technique for recognizing strawberry ripeness in the field. The approach has three steps: after adding self-calibrated convolutions to the Mask R-CNN backbone network to boost model performance, the model extracts the strawberry target from the picture. In the second step, region segmentation divides the strawberry target into four sub-regions and extracts color features. The final step classifies and visualizes strawberry ripeness using color feature values. SVM classifiers provide the best strawberry ripeness classification effect. Classification outperforms manual feature extraction and AlexNet, ResNet18 models. Strawberry improved planting management decisions may be made accurately using this strategy. Precision field segmentation using satellite data is a major difficulty in sugarcane yield prediction and crop management. Yuan et al. propose DSCA-PSPNet using a modified ResNet34 and pyramid scene parsing network with new modules. The proposed sugarcane field feature representation is preferable since it can respond to spatial and channel-wise information.

Object detection

Jia et al. present an improved YOLOX_m approach for effective green fruit recognition in complicated orchard situations. First, the model uses the CSPDarkNet backbone network to extract three effective feature layers at various sizes from the input picture. These effective feature layers are fed into the feature fusion pyramid network for enhanced feature extraction, which combines feature information from different scales. The Atrous spatial pyramid pooling module increases the receptive field and the network's ability to obtain multi-scale contextual information. For classification and regression prediction, the head prediction network receives the fused features. Varifocal loss also reduces the influence of an imbalanced positive and negative sample distribution to improve accuracy. Khan et al. explore the use of edge computing devices to improve the accuracy of deep learning models for agricultural applications, while taking into account resource restrictions. Example data came from the publicly accessible Plant Village dataset of healthy and sick leaves for 14 crop species and 6 disease categories. The MobileNetV3-small model achieved 99.50% accuracy in leaf classification. Quantization-based post-training optimization lowered model parameters from 1.5 million to 0.9 million while retaining 99.50% accuracy. The final ONNX model allows deployment on mobile devices and other platforms. It provides a cost-effective way to deploy accurate deep-learning models in agriculture. Vello et al. studied the usefulness of image-based phenotyping using fluorescent and visible light pictures to measure and classify Camelina seeds. They created SeedML, a user-friendly online service that uses phenomics platforms with fluorescent and visible light cameras to detect Camelina seeds from high-salt plants. This gateway can improve quality control, detect stress signs, and track agricultural productivity trends with high throughput. This study may aid climate crisis research and agri-food quality control tool deployment. Mbouembe proposes SBCS-YOLOv5s, an effective tomato identification method. SBCS-YOLOv5s adds SE, BiFPN, CARAFE, and Soft-NMS modules to improve model feature expression. Modelling channel-wise interactions and adaptive re-calibration of feature maps enable the SE attention module to catch essential information and enhance model feature extraction. The SE module's adaptive re-calibration may also increase model resilience to environmental changes. Next, an efficient, weighted bidirectional feature pyramid network replaced the PANet multi-scale feature fusion network. Third, the neck network replaces the upsampling operator with CARAFE. Better feature maps with more semantic information result from this approach. CARAFE's spatial detail enhancement helps the model distinguish closely placed fruits. Finally, the Soft-NMS method replaced the Non-Maximum-Suppression (NMS) approach to better identify occluded and overlapping fruits. Soft-NMS's continuous weighting approach makes it better at managing little and big fruits in images. Yang

et al. used a multi-sensor fusion and CNN to identify moisture content in agricultural goods during drying in real time. This work designed a multi-sensor data collection platform and created a CNN prediction model using raw load, air velocity, temperature, tray position data and material weight data. In the model performance comparison, the CNN prediction model had the best prediction effect. Validation trials demonstrated that the detection system satisfied online moisture content detection criteria for agricultural product drying. This work allows online detection of various agricultural product drying indicators. Popescu et al. discuss neural network-based emerging agricultural trends for detecting hazardous insects and pests. Using a systematic review, this technology's pros and cons and researchers' methods for improving it are discussed. This review examines pest detection using neural networks, pest databases, current software, and unique modified architectures. Multiple research publications from 2015 to 2022 were analyzed, with fresh patterns analyzed between 2020 and 2022. Molina-Rotger et al. study the use of random forest and support vector machine algorithms to detect and classify olive flies in a Raspberry Pi B+-based electronic trap. Combining the two approaches improves classification accuracy with a limited training data set.

Monitoring

For ecological fruit production, orchard monitoring is an essential study and practice. Popescu et al. discuss recent advances in orchard monitoring, focusing on neural networks, UAVs, and practical applications. Papers on complicated issues found by combining field keywords were chosen and examined. The study focused on 2017–2022 studies on neural networks and UAVs in orchard monitoring and productivity assessment. UAV trajectories and flights in the orchard were emphasized due to their intricacy. The structure and implementation of the newest neural network systems utilized in such applications, databases, software, and performance are studied. To make recommendations for researchers and end users, the new concepts and their implementations were surveyed in concrete applications.

Path planning

Zhang et al. offer an enhanced local route planning technique for an artificial potential field, including an elliptic repulsion potential field as the border potential field. The potential field function solves unreachable objectives and local minima by using an enhanced variable polynomial and a distance factor. The scope of the repulsion potential field is changed to an ellipse, and a fruit tree boundary potential field is added, which reduces environmental potential field complexity, allows the robot to avoid obstacles without crossing the fruit tree boundary, and improves its safety when working independently.

All of the 15 accepted papers include advances and novelties in the different topics covered by the Research Topic "*IoT, UAV, BCI empowered deep learning models in precision agriculture*". The editors are pleased to present this collection of articles to the precision agriculture research area and others related areas, and they hope that it will help researchers' advances in the future.

Author contributions

JL: Writing – original draft. JD: Writing – original draft.

Acknowledgments

The authors would like to express their sincere gratitude to Instituto Politécnico de Setúbal, ESTSetúbal, Campus do IPS, Estefanilha, Edifício Sede, 2910-761 Setúbal, Portugal and School of Intelligence Engineering, Shandong Management University, Jinan, China, for all received support.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be considered as a potential conflict of interest.

Publisher's note

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 08 May 2023
DOI 10.3389/fpls.2023.1187734

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Weiyang Chen,
Qufu Normal University, China
Bin Yang,
Huaiyin Normal University, China

*CORRESPONDENCE
Weikuan Jia
jwk_1982@163.com
Xinting Ge
xintingge@163.com

RECEIVED 16 March 2023
ACCEPTED 05 April 2023
PUBLISHED 08 May 2023

CITATION
Jia W, Xu Y, Lu Y, Yin X, Pan N, Jiang R and Ge X (2023) An accurate green fruits detection method based on optimized YOLOX-m.
Front. Plant Sci. 14:1187734.
doi: 10.3389/fpls.2023.1187734

An accurate green fruits detection method based on optimized YOLOX-m

Weikuan Jia[1,2*], Ying Xu[1], Yuqi Lu[1], Xiang Yin[3], Ningning Pan[1], Ru Jiang[1] and Xinting Ge[1,4*]

[1]School of Information Science and Engineering, Shandong Normal University, Jinan, China, [2]School of Information Science and Engineering, Zaozhuang University, Zaozhuang, China, [3]School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo, Shandong, China, [4]School of Medical Imaging, Xuzhou Medical University, Xuzhou, China

Fruit detection and recognition has an important impact on fruit and vegetable harvesting, yield prediction and growth information monitoring in the automation process of modern agriculture, and the actual complex environment of orchards poses some challenges for accurate fruit detection. In order to achieve accurate detection of green fruits in complex orchard environments, this paper proposes an accurate object detection method for green fruits based on optimized YOLOX_m. First, the model extracts features from the input image using the CSPDarkNet backbone network to obtain three effective feature layers at different scales. Then, these effective feature layers are fed into the feature fusion pyramid network for enhanced feature extraction, which combines feature information from different scales, and in this process, the Atrous spatial pyramid pooling (ASPP) module is used to increase the receptive field and enhance the network's ability to obtain multi-scale contextual information. Finally, the fused features are fed into the head prediction network for classification prediction and regression prediction. In addition, Varifocal loss is used to mitigate the negative impact of unbalanced distribution of positive and negative samples to obtain higher precision. The experimental results show that the model in this paper has improved on both apple and persimmon datasets, with the average precision (AP) reaching 64.3% and 74.7%, respectively. Compared with other models commonly used for detection, the model approach in this study has a higher average precision and has improved in other performance metrics, which can provide a reference for the detection of other fruits and vegetables.

KEYWORDS

green fruits, YOLOX_m, Atrous spatial pyramid pooling, varifocal loss, object detection (OD)

1 Introduction

In the world, the annual consumption of fruits in all countries is huge and has been showing an increasing trend, so the production and planting area have been expanding in recent years, which requires a lot of human resources. In order to reduce labor costs, the production and management of modern agriculture is gradually developing in the direction

of automation. In recent years, computer vision technology is gradually being applied to modern agriculture because of its role in vision systems for agricultural automation equipment, such as pest and disease identification and detection (He et al., 2013; Fuentes et al., 2017; Johnson et al., 2021), automated harvesting of fruits and vegetables (Jia et al., 2020; Wang et al., 2022; Tang et al., 2023), crop growth information monitoring and yield estimation (Apolo-Apolo et al., 2020; Li et al., 2020), and so on. The precision of the vision system detection determines the efficiency of the automated equipment, and the complexity of the modern orchard environment makes its ability to accurately detect the target fruit dependent on a variety of factors, such as the angle of light, weather conditions, and the overlap of shading between fruits, etc. In addition, the color of most immature fruits is green, so the research on the detection of green fruits is important for the subsequent operation of fruits, such as yield estimation and fruit harvesting, etc., but the similar color of immature green fruits and leaves will cause the boundary to be more difficult to distinguish, which will also have an impact on the precise detection of fruits. These problems have attracted the attention of many domestic and international scholars, who have carried out some relevant research and achieved some results.

Traditional machine learning plays an important role in the field of computer vision, and many results have been achieved in machine learning detection research in agricultural fruit detection. Linker (Linker et al., 2012) proposed a green apple recognition model based on fruit characterization information with a correct detection rate close to 95%. Wu (Wu et al., 2020) proposed a fruit point cloud segmentation method combining color and 3D geometric features, where local descriptors were used to obtain candidate regions and global descriptors were used to obtain the final segmentation results. Wang (Wang et al., 2021) proposed a new kernel density clustering (KDC) to better realize the accurate identification of green apples. Tian (Tian Y. et al., 2019) proposed a fruit localization algorithm based on image depth information, which fits the detection region by introducing a segmentation algorithm to locate the center and radius of the apple circle, respectively, through the gradient information obtained from the depth apple image and the corresponding RGB spatial information. Moallem (Moallem et al., 2017) used the multilayer perceptron (MLP) and k-nearest neighbors (KNN) to classify the apples with 92.5% and 89.2% recognition rates for the extracted features, respectively. Traditional machine learning for agricultural fruit detection is relatively well established, but the limitations of machine learning also limit the speed and precision of object detection.

In recent years, with the rapid development of deep learning and convolutional networks, they have eliminated some of the limitations and complex operations of traditional machine learning. Computer vision has also shifted its research focus to deep learning and convolutional networks, and has been widely used in many fields. At present, research on vision systems for agricultural automation equipment has also focused on deep learning models, and some results have been achieved. Sun (Sun et al., 2022) proposed a balanced feature pyramid for small apple detection, which achieved an average detection accuracy of 35.6% for small targets on the Pascal VOC benchmark with good generalization performance. Wang (Wang and He, 2019) proposed an apple object detection and recognition algorithm based on R-FCN. The model uses ResNet-44 as the backbone network, which improves the detection accuracy and simplifies the network. Triki (Triki et al., 2021) proposed a Mask RCNN-based leaf detection and pixel segmentation technique that can segment leaves of different families, measure the length and width of leaves, and reduce the recognition error. Mu (Mu et al., 2020) performed detection of highly shaded unripe tomatoes based on deep learning techniques, combined with regional convolutional networks (R-CNN) and Resnet-101, for ripeness detection and yield prediction of tomatoes. Jia (Jia et al., 2022b) proposed a Mask R-CNN based segmentation model RS-Net, which achieves robust segmentation of green apples to meet the accuracy and robustness of vision systems in agronomic management. Kang (Kang and Chen, 2020) obtained DASNet-v2 by improving DASNet, which uses visual sensors to segment apple instances, so it can achieve segmentation of fruits more robustly and efficiently.

Compared with traditional machine learning, the detection accuracy of the above research has been greatly improved, but due to the complex environment of real orchards, the existence of difficult detection conditions such as leaves obscuring fruits, overlapping fruits, and the similar color of fruits and branches, the accuracy of the above methods for fruit detection still does not meet the needs of modern automated agriculture, and the precision needs to be further improved.

Therefore, in order to simulate the actual environment of the orchard as much as possible, this paper collects images of green apples and persimmons in various complex situations to make two datasets and proposes an improved YOLOX-m network model to improve the detection accuracy of the fruits. The model uses the CSPDarknet backbone network to better extract image features. In the multi-scale feature fusion stage, referring to the PAnet structure, it will not only upsample the features to achieve feature fusion, but the features are also downsampled to achieve feature fusion, and ASPP (Atrous spatial pyramid pooling) is used to increase the receptive field during fusion, so that each convolution output contains a larger range of information, thereby improving network performance and reducing the rate of missed and wrong detections. In addition, Varifocal loss is used instead of BCE (binary cross-entropy) loss to mitigate the negative effects of sample imbalance and better optimize the model parameters to improve the detection accuracy of green fruits in complex orchard environments.

2 Datasets production and experimental setup

2.1 Datasets collection

The datasets used in this paper are the immature green persimmon and green apple datasets. The persimmon images constituting the dataset were collected from the back mountains of Shandong Normal University (Changqing Lake Campus) and the southern mountainous region of Jinan, using a Canon EOS 80D

SLR camera with a CMOS image sensor, and the apple images were collected from the apple production base in Longwang Mountain, Fushan District, Yantai City, Shandong Province, using a Sony Alpha 7 II camera. The image resolution was 6000 pixels × 4000 pixels, saved in.jpg format, and 24-bit color image. Figure 1 list several collected images of apples and persimmons in different complex situations.

The actual environment of the orchard is more complex, and in order to simulate the real situation as much as possible, the dataset collects images of different complex situations. It is not easy to discriminate overlapping fruit boundaries by shading, water drops on fruits after rain can be a factor affecting detection, and different lighting can also affect the final detection effect. Considering, a total of 553 images of green persimmons and 1361 images of green

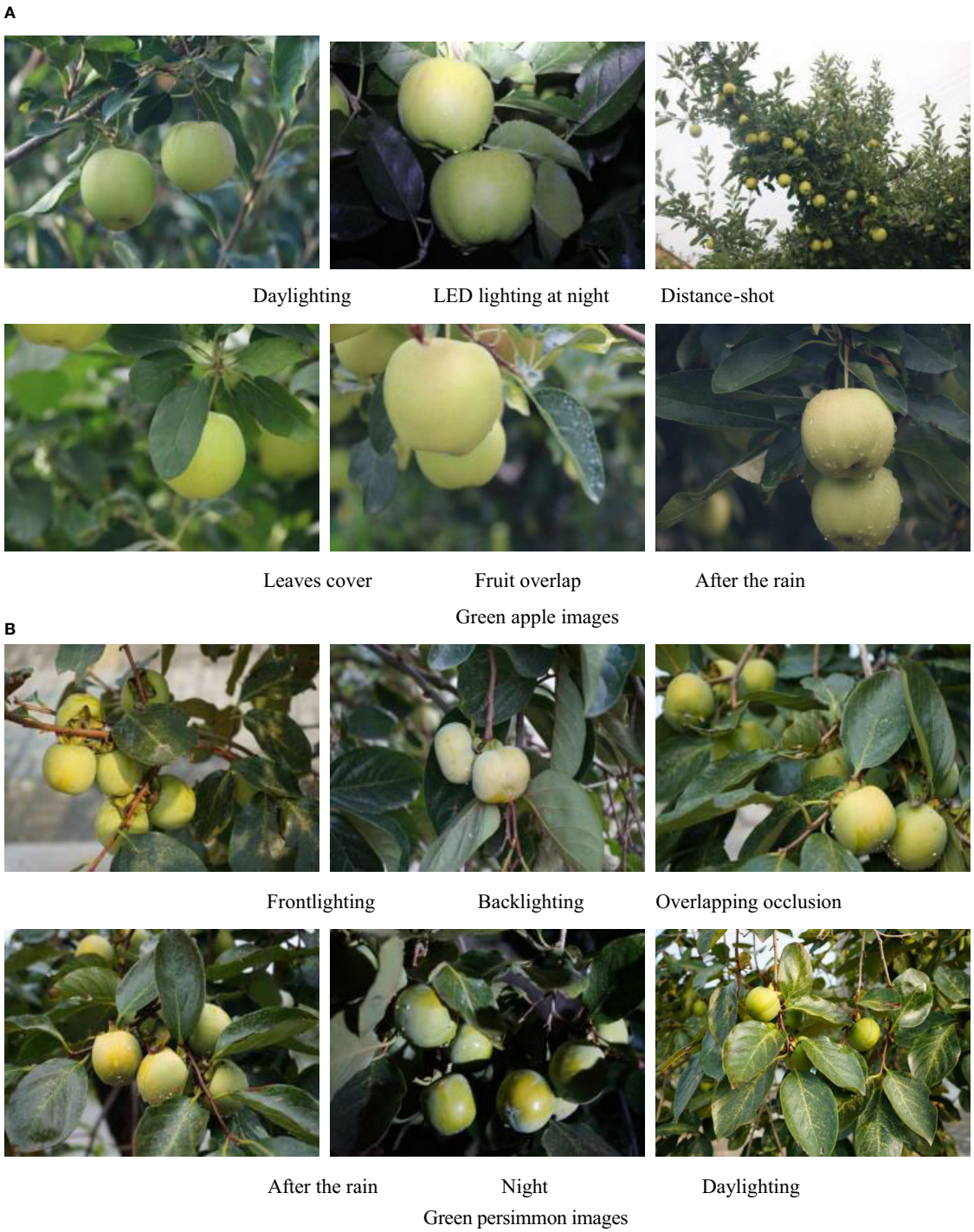


FIGURE 1
Images of green fruit in different situations. **(A)** Green apple images **(B)** Green persimmon images.

apples were finally collected under different situations, including down-lighting, back-lighting, daytime and nighttime LED lighting, overlapping fruits, and leaf shading. Among them, the persimmon and apple datasets contained 2524 and 7137 fruits, respectively, and Table 1 shows the number and proportion of fruits of different scale sizes, where the ground truth box area less than 32^2 belongs to the small-scale target fruits, the ground truth box area between 32^2 and 96^2 belongs to the medium-scale target fruits, and the ground truth box area greater than 96^2 belongs to the large-scale target fruits.

2.2 Datasets production

The collected apple and persimmon images were divided into training set and test set in the ratio of 7:3. After the division, the apple training set included 953 images and the test set included 408 images; the persimmon training set included 388 images and the test set included 165 images. And in order to reduce the computational effort and the subsequent experiment time, the image resolution was uniformly scaled from 6000×4000 pixels to 600×400 pixels. The labeling software used is LabelMe, and the edge contours of the fruit are labeled with labeling points, so that the fruit can be separated from the background. The labeling information of the image and the coordinates of the labeling points are saved in the corresponding.json file, and the completed json file is finally converted into a coco format dataset (Lin et al., 2014).

3 Optimized YOLOX-m network

The actual orchard environment is complex and variable, and the color of green fruits is similar to the leaves, which further makes the boundary between the background and the fruits blurred and unclear, not easy to decide, causing the detection of fruits to be more difficult and affecting the final accuracy of detection. In order to improve the object detection accuracy of green fruits and improve the vision system of agricultural automation equipment, this paper proposes an improved YOLOX_m (Ge et al., 2021) model for efficient detection of green fruits, and the specific detection framework is shown in Figure 2.

The model in this paper uses CSPDarknet (Bochkovskiy et al., 2020) as the backbone network for feature extraction of apple and persimmon images, and the input images will get three effective feature layers C1, C2, and C3 through the backbone network. The bottom feature layer has less semantic information, but accurate target location information, and the higher-level features have rich semantic information, but locate the target location more roughly, so the feature map needs to go through a feature fusion pyramid (Lin et al., 2017a) for feature fusion before classification and regression prediction, combining feature information of different scales. As shown in Figure 2, in the feature fusion stage, the model in this paper introduces the Atrous Spatial Pooling Pyramid (ASPP) module before the upsampling operation, which sets different dilation rates to construct convolution kernels with different receptive fields, and increases the receptive fields by parallelizing multiple Atrous convolution layers with different dilation rates to obtain multi-scale information of the target, so as to more effectively enhance feature extraction and improve the detection accuracy of the target green fruits. The three fused feature layers are input to the prediction head, and the prediction head of the model is decoupled to perform classification and regression to determine whether the target is a green fruit or a background, and to accurately locate the target fruit.

In addition, although the original model reduces the number of negative samples, the target fruit still only accounts for a small portion of the entire input image, and the number of positive samples is still far less than the number of negative samples. To further alleviate the negative impact of sample imbalance, the loss function was replaced from BCE (binary cross-entropy) loss to Varifocal loss (Zhang et al., 2021) to make the model focus more on difficult to classify samples and to focus training on positive samples, which can better optimize the model parameters, improve detection accuracy and reduce the false detection rate, thus improving the fruit picking and yield prediction and other aspects of accuracy.

3.1 Backbone network CSPDarkNet

Taking into account the difficult detection problems such as the similarity of green fruits to the background and the overlapping of fruit occlusion, in order to extract the features of the images more effectively, the model in this paper uses CSPDarkNet as the backbone network, and the input immature green persimmon and green apple images use the backbone network CSPDarkNet

TABLE 1 The divided results of datasets by area size of fruit.

Area	Small	Medium	Large	Fruit Total	Image Total
Apple Dateset					
Train	1701/34%	2007/41%	1235/25%	4943	953
Val	851/39%	816/37%	527/24%	2194	408
Total	2552/36%	2823/39%	1762/25%	7137	1361
Persimmon Dataset					
Train	272/15%	1111/59%	482/26%	1865	388
Val	47/7%	415/63%	197/30%	659	165
Total	319/13%	1256/60%	679/27%	2524	553

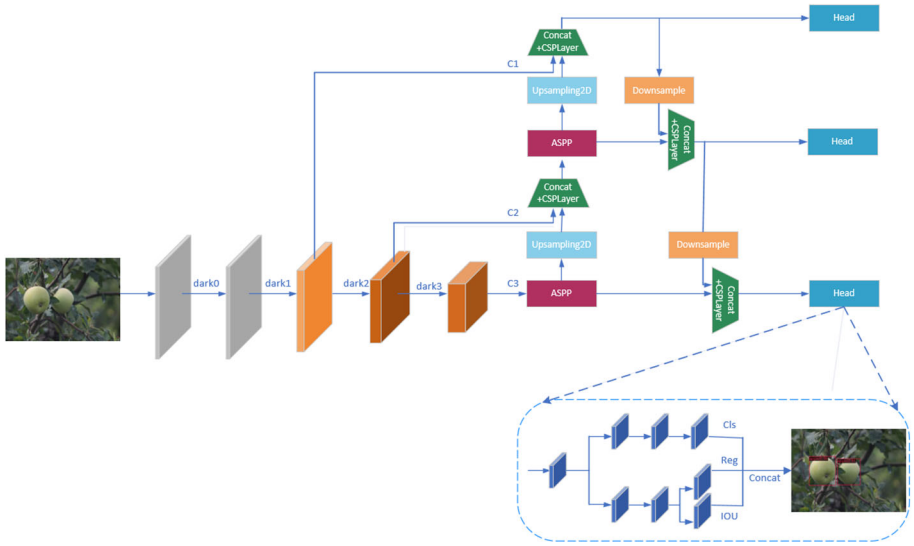


FIGURE 2
Improve YOLOX-m network detection framework.

for feature extraction to obtain three effective feature layers of different scales, using them for subsequent training and prediction. The residual module in CSPDarkNet is based on the network structure of residual network and CSPNet (Wang et al., 2020). The jump link in the residual network can effectively mitigate the gradient disappearance problem as the network depth increases, while the use of CSP structure can enhance the learning ability of the convolutional neural network and speed up the inference. First, the input image is passed through the Fcous network to reduce the number of parameters and improve the running speed of the model, then, after a series of operations of convolutional regularization and activation function for a channel expansion, and finally, three effective feature layers of different scales are output in turn through four residual modules, and the structure of CSPLayer in the residual module is shown in Figure 3.

The green persimmon and apple images are continuously feature extracted by four residual modules in the backbone network CSPDarkNet. During this process, the width and height of the feature maps are continuously halved and the number of channels is expanded to twice. When passing through the last three residual modules, three effective feature layers at different scales are output respectively. Although the semantic information is gradually enriched during the feature extraction process, the image resolution decreases and the boundary information is lost, so the information contained in the three feature layers will be different. Therefore, before inputting the feature map into the head for prediction, it is necessary to fuse the features of different scales through the feature pyramid, so as to better predict the fruit for classification regression.

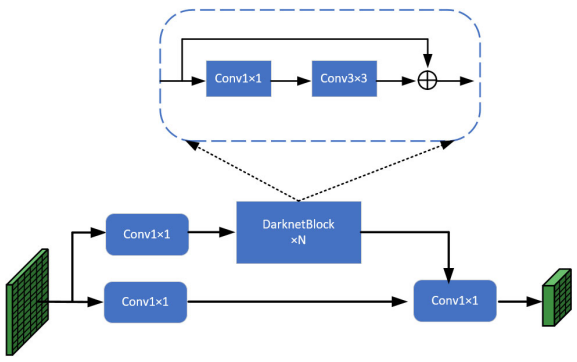


FIGURE 3
CSPLayer structure schematic diagram.

3.2 Feature pyramid network

Originally, Atrous convolution and ASPP (Atrous Spatial Pyramid Pooling) (Sullivan and Lu, 2007) were proposed in the semantic segmentation model DeepLabv2 (Chen et al., 2017). Compared with ordinary convolution, atrous convolution has an additional parameter dilation rate, which increases the receptive field of the convolution kernel without causing information loss. Atrous convolution is an important part of the ASPP module, which sets different dilation rates to construct convolution kernels with different receptive fields, and obtains multi-scale information of the target by parallelizing multiple atrous convolution layers with different dilation rates. In this way, the receptive field can be increased while ensuring that there is not much loss of resolution. If the loss of resolution is too large, the information of the fruit image boundary will be lost, which is not beneficial to the detection of green fruits. The module specifically consists of a 1×1 convolution, three atrous convolution layers with different dilation rates, and an atrous pooling layer in parallel, and the obtained results are concatenated in the channel dimension, and then, the output is obtained after another 1×1 convolution layer for channel number reduction. The specific structure as shown in Figure 4.

When feature extraction is performed on apple and persimmon images, the semantic information and location detail information of

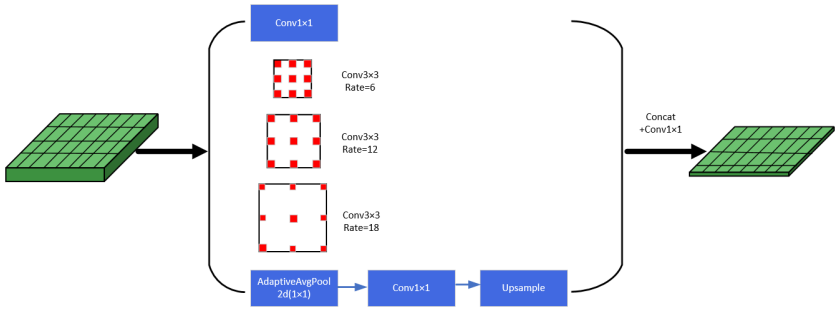


FIGURE 4
Atrous spatial pyramid pooling structure.

the feature layers change continuously because of the need for constant convolution and down sampling. The initial low-level feature layer C1 is rich in spatial information and locates the location more accurately, but contains less semantic information, so the green fruits with similar color to the branches and leaves are more difficult to be determined and easy to detect incorrectly. The feature layer from C1 to C3, the semantic information becomes richer, but the resolution gradually decreases and the detail information such as boundary is lost, so the localization of the target fruit is rougher, and the higher feature layer C3 can determine the target species more accurately, but it is not conducive to the localization of the target fruit. Therefore, to improve the accuracy of final classification and localization, a feature pyramid network is used to enhance feature extraction, and the feature layers of different scales of green fruit images are complemented with advantages to make the information of feature layers more comprehensive. The feature fusion pyramid network used in this model refers to the structure of PANet (Liu et al., 2018). In the process of feature fusion, it will not only start from the high level features, perform up-sampling operation and fuse with the low level features, but also perform down-sampling operation on the three feature layers after up-sampling and fusion from the low level, and perform feature fusion again to get the final input head for prediction of feature layers.

Before the upsampling operation, the feature layer needs to be down-dimensioned by a 1×1 convolution to reduce the number of channels. In order to increase the receptive field, capture multi-scale information, and better extract features at different scales, the model in this paper adds a 1×1 convolution layer, atrous convolution layers with different dilation rates, an atrous pooling layer, etc. in parallel before the dimensionality reduction operation, and concatenates the results together. Therefore, Atrous Spatial Pyramid Pooling (ASPP) is introduced to replace the 1×1 convolutional layer before upsampling to obtain more accurate localization and classification information of the target green fruits.

3.3 Loss function

The construction of the loss function has an important significance to the training of the model, and the main role is that during training, the model will use the loss values obtained during forward propagation to update the training parameter weights through backward propagation. After continuous iterations, the loss difference between the prediction box and the ground truth box is gradually reduced, and the loss function will gradually reach the minimum value, so that the prediction box gradually overlaps close to the ground truth box, thus achieving accurate localization of the target green fruits. In this paper, the loss of the model during training mainly contains classification loss, regression loss and confidence loss. The IOU (intersection of union) loss is used for the regression loss, and the Varifocal loss is used for the classification and confidence loss, and the formula for the overall loss function of the model is shown in equation (1).

$$\text{Loss} = \frac{1}{\text{N}_{\text{pos}}}(\text{L}_{\text{cls}} + \lambda L_{reg} + L_{obj}) \tag{1}$$

Where N_{pos} refers to the number of feature points that are assigned as positive sample points, L_{cls} refers to the classification loss, L_{reg} refers to the regression loss, and L_{obj} refers to the confidence loss, λ is the balance coefficient of the regression loss, set to 5.0.

The regression loss refers to the IOU loss between the ground truth box and the predicted box, and is calculated as shown in equation (2).

$$\text{IOU loss} = -\ln\frac{\text{Intersection}(\text{B}_{\text{gt}}, \text{B}_{\text{pred}})}{\text{Union}(\text{B}_{\text{gt}}, \text{B}_{\text{pred}})} \tag{2}$$

where Intersection(B_{gt}, B_{pred}) refers to the area where the real frame intersects the prediction frame, and Union(B_{gt}, B_{pred}) refers to the area where the real frame and the prediction frame are combined and summed.

In the actual training phase of the model, the target green fruit only accounts for a small portion of the whole input image, so the number of negative samples is much larger than the number of positive samples, and there will be an unbalanced distribution of positive and negative samples, which will lead to a decrease in training accuracy and the optimization direction of the model is not as desired. In addition, the commonly used loss function BCE loss does not distinguish between samples that are difficult to classify and those that are easy to classify. When the negative samples that are easy to classify are much more than the positive samples, the

model will focus more on these negative samples and drown out the impact of the positive samples that help training, causing a loss in the final detection precision. In order to alleviate the above negative effects and improve the detection accuracy of fruits, the classification and confidence loss function of the model in this paper adopts Varifocal loss, which is based on BCE loss, and the specific formula of the loss function is shown in equation (3).

$$\mathrm{VFL}(p,q) = \begin{cases} -q(q\log(p) + (1-q)\log(1-p) & q > 0 \\ -\alpha p^{\gamma}\log(1-p) & q = 0 \end{cases} \quad (3)$$

In the formula, α, γ are hyperparameters, α is the balance parameter to adjust the weight of positive and negative samples, and the tempering factor p^{γ} can reduce the influence of easy to classify samples on the loss and make the model focus more on difficult to classify samples, such as targets in the image that are obscured by leaves or overlapped with other fruits. Varifocal loss is treated differently for positive and negative samples compared to focal loss. For negative samples, q=0, in this case, p^{γ} can be used to reduce the loss contribution of negative samples, and for positive samples, which is the case of q>0, the value of q is the IOU between the prediction box and the ground truth box, and q is used to weight the positive samples, so that when the positive sample has a higher IOU, its contribution to the loss is also large, and it allows the model to focus its training on high-quality positive samples, which can result in better detection accuracy and better detection of the target green fruits.

4 Experiments

4.1 Experimental design and operation platform

The server environment used for model training in this paper is Ubuntu 18.04 OS, NVIDIA A30 graphics card and 11.1 CUDA environment. The programming language used in the model is python, and the Pytorch 1.8 (Paszke et al., 2019) deep learning library is also used in this process, and the implementation is built with the help of MMdetection (Chen et al., 2019) related modules.

Before formal training, the pre-training weights obtained using the ImageNet dataset are imported as initialization parameters to accelerate the detection speed and improve the robustness of the model. In the formal training phase, the model parameters are optimized and updated using the SGD optimizer. The learning rate, momentum factor, and weight decay factor are set to 0.00125, 0.9, and 0.0005, respectively, and 300 epochs are trained iteratively, and the parameter results are saved once every 10 iterations. The variation of the loss during training is shown in Figures 5A, B, where the x-axis represents the number of iterations and the y-axis indicates the value of the loss function, and different colors are used to distinguish the various types of losses.

4.2 Assessment metrics

In order to comprehensively evaluate the performance of the model, this paper uses a variety of assessment metrics to evaluate the effect, among which the main consideration is the average precision (AP) of detection. The precision (P) is the probability of the samples being correctly predicted among all samples, calculated as shown in equation (4), and the recall (R) is the probability of the positive samples being correctly predicted among the prediction results, calculated as shown in equation (5).

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + FP} \quad (4)$$

$$R = \frac{\mathrm{TP}}{TP + FN} \quad (5)$$

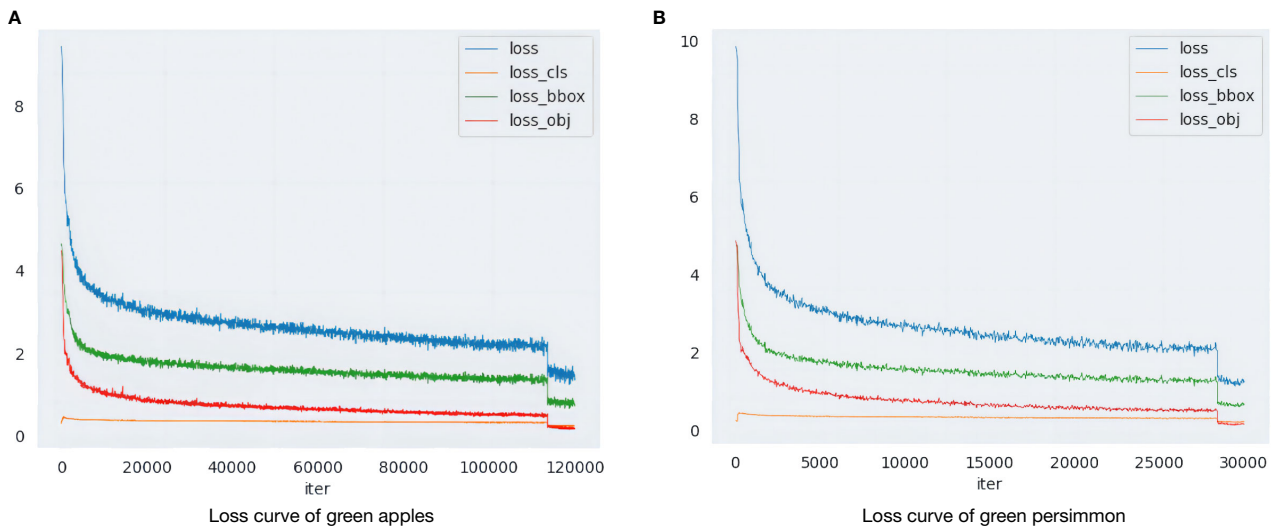


FIGURE 5
Loss function change curve. **(A)** Loss curve of green apples. **(B)** Loss curve of green persimmon.

Where TP, FP, and FN are the number of true positive samples, the number of false positive samples, and the number of false negative samples, respectively. Further it is possible to calculate the AP (Average precision) under a specific IOU threshold, and the calculation formula is shown in equation (6).

$$AP^{IOU=i} = 1/101 \sum_{r \in R} \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \quad (6)$$

where i is the value of the settable IOU threshold, whose value can be set in a range greater than or equal to 0.5 less than 1, i $\in$I [0.5,0.55,0.6, ……, 0.95], with a total of 10 values, p(r) denotes the accuracy rate associated with the recall, R $\in$ [0, 0.01, 0.02, ……, 1] with 101 values, and r denotes the value taken as the recall rate. Continuing to average the 10, the final AP metric used can be obtained, and the formula is shown in equation (7).

$$AP = \frac{1}{10} \sum_{i \in I} AP^{IOU=i} \quad (7)$$

In order to evaluate the performance of the model approach in more detail, a number of other evaluation metrics are used. AR refers to the average recall; $AP^{IOU=0.5}$ and $AP^{IOU=0.75}$ refer to the AP value when the IOU threshold is over 0.5 and 0.75, respectively; AP_S, AP_M and AP_L refer to the average detection accuracy for small, medium and large scale target fruits, respectively, where the ground truth box area less than 32^2 belongs to the small-scale target fruits, the ground truth box area between 32^2 and 96^2 belongs to the medium-scale target fruits, and the ground truth box area greater than 96^2 belongs to the large-scale target fruits; In addition, Time refers to the speed of validation set detection to evaluate an image in *ms*; Params refers to the total parameters to measure the size of the model; and FLOPs refers to floating point operations to measure the computational complexity of the model.

4.3 Results and analysis

4.3.1 Green fruit detection effect

In this paper, we use the improved yolox_m network model to analyze the target fruit detection effect on the collected immature green persimmon and green apple datasets. The pictures contained in the datasets restore the complex environmental conditions of real orchards as much as possible, considering different shooting distances, different situations such as overlapping fruit shading, after rain, at night and smooth backlighting, etc. The detection effect under several situations is selected for analysis, and the specific detection effect is shown in Figures 6A, B.

As can be seen from Figure 6, we can see that the sparse and independent fruits will have a clearer and more complete outline, so the detection accuracy of such target fruits is better, and the detection effect of the images collected at night can also reach a better level. In terms of distance, the detection effect of close-range fruit is better than that of distant target fruit. For those densely-distanced fruit or occluded and overlapping target fruit, the detection is relatively difficult, and the accuracy is slightly reduced, but there are almost no omissions and errors.

In Figures 7A, B, it can be seen that True Positive is 89% and 96% for apples and persimmons, respectively, which is an

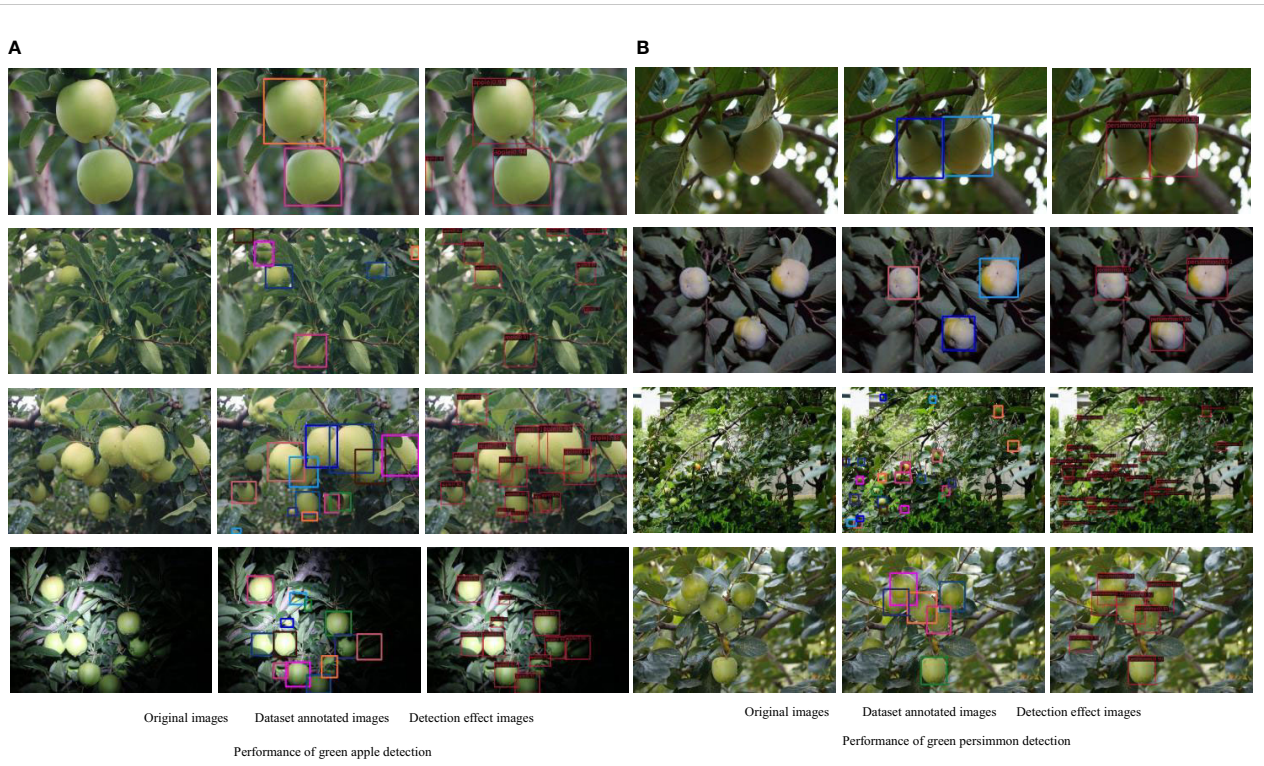


FIGURE 6
Green fruits detection effect images. **(A)** Performance of green apple detection **(B)** Performance of green persimmon detection.

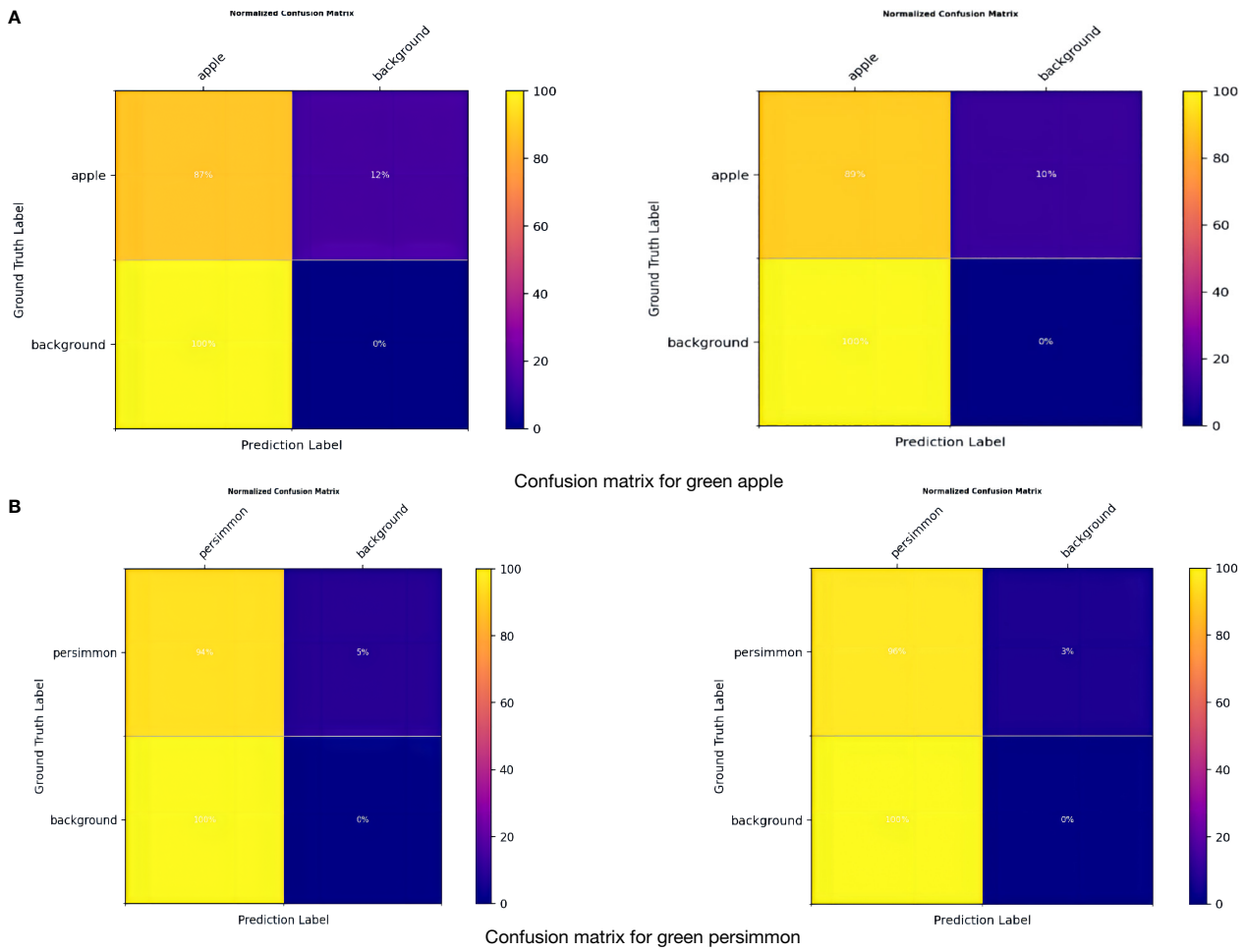


FIGURE 7
Confusion matrix. **(A)** Confusion matrix for green apple. **(B)** Confusion matrix for green persimmon.

improvement of 2%, and False Negative is a decrease of 2%. Overall, although the complex reality of orchards brings some negative effects on detection, the model in this paper achieves a good level of detection accuracy for target fruits, and some target fruits that were not labeled at the time of dataset labeling can be detected, with a decrease in the rate of omission and error.

In order to fully verify the performance of the model in this paper, the model was tested on two datasets, apple and persimmon, and the model detection effect basically reached the highest accuracy after the last epoch. In order to validate the improvement effect, the original network without improvement is recorded as yolox_origin, the network with only the improved feature pyramid is recorded as yolox_A, the network with only the improved loss function is recorded as yolox_V, and the network with all the improvements is recorded as yolox_after, and the results of various evaluation indicators on the validation set are shown in Table 2, and the change curve of mAP is shown in Figure 8.

From Table 2 and Figure 8, it can be found that the final detection average precision of the method in this paper for green apple and green persimmon images is 64.8% and 74.7%, and the average recall rate is 72.6% and 81.5%, respectively. It can be seen from the table that using the atrous spatial convolution pooling pyramid (ASPP) and the loss function using the Varifocal loss can improve the detection accuracy of the model and improve the model performance on both datasets. In addition, $AP^{IOU=0.5}$ and $AP^{IOU=0.75}$ have also been greatly improved, and in both data sets, the average accuracy of large, medium and small targets has been improved to a certain extent. The detection accuracy on large targets can also reach about 90%.

4.3.2 Comparison of model detection effects

In order to objectively analyze and compare the performance of the model in this paper, we compare the model with several common and representative object detection model algorithms. The selected models are FCOS (Tian et al., 2019), Faster-RCNN (Ren et al., 2015), YOLOv3 (Redmon and Farhadi, 2018), SSD (Liu et al., 2016), FSAF (Zhu et al., 2019) and ATSS (Zhang et al., 2020), where Faster-RCNN is a two-stage detection model based on anchor frames, YOLOv3, SSD and ATSS are single-stage detection models based on anchor frames, and FCOS as well as

TABLE 2 Image detection and evaluation results.

Network	Metric						
	AP	$AP^{IOU=0.5}$	$AP^{IOU=0.75}$	AP_S	AP_M	AP_L	AR
Apple Dataset							%
yolox_origin	62.9	87.3	68.4	44.3	69.4	91.9	68.6
yolox_A	63.7	88	70	46.6	69.8	90.9	69.7
yolox_V	63.8	87.4	69.8	46.4	70.2	91.4	69.5
yolox_after	64.8	88.4	71.2	47.7	70.7	92.1	72.6
Persimmon Dataset							%
yolox_origin	72.7	91.3	82.1	36.6	73.9	86.7	78.5
yolox_A	74	91.6	84.6	39.2	74.8	88.2	79.6
yolox_V	73.6	91.5	83.3	36.6	74.5	88.3	80.5
yolox_after	74.7	91.9	84	39	75.6	89.4	81.5

FSAF belong to the detection model with anchor-free. The above models will be trained and validated for evaluation on two datasets of apples and persimmons, respectively, and the specific evaluation index results obtained are shown in Table 3. In addition, a picture with high detection difficulty is randomly selected in each of the two datasets and detected with the above models, respectively, and the detection effect images are shown in Figures 9A, B.

From the images of different model detection effects on the two datasets, it can be seen that some target fruits in the images that are not labeled because they are not easily labeled or forgotten at the time of labeling can basically be detected at the time of model detection, among which the detection effect of the model method in this paper is better. For the fruits that are severely obscured by leaves in the figure, several other models did not detect them, but this model can still detect them, and it can be seen that the detection accuracy of this model is higher compared with several other detection models in the case of overlapping obscured fruits with LED lighting at night.

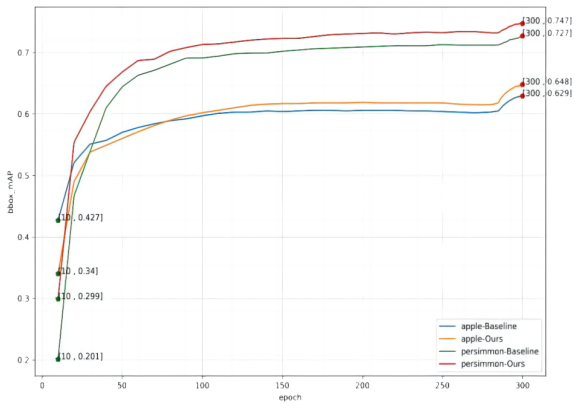


FIGURE 8
mAP curve for each epoch.

From the comparison results of various evaluation indexes of different models shown in Table 3, it can be seen that the average detection accuracy of this model is better than several other detection models on two datasets, the average accuracy is 2.6-7.2 percentage points higher than other models on the apple dataset, and the average accuracy is 1.9-5 percentage points higher than other models on the persimmon dataset. For $AP^{IOU=0.5}$ and AR, the results of this model are also basically better than other models. In addition, the model results are most similar to the model in this paper for ATSS, and the average precision of the model in this paper is also 2.6% and 1.9% higher than ATSS on both datasets, and the average recall is 3.3% and 1% higher, respectively. When evaluating on the validation set, it is also necessary to consider the detection time for recognizing an image. Through Table 3, the average precision and average recall of FSAF and ATSS are closest to the results of the models in this paper, but the detection time used by the models in this paper to recognize an image is only about 45% of theirs. Overall, the model in this paper has a better real-time performance with higher average accuracy and average recall than the other models.

As can be seen from Tables 3, 4, the model in this paper introduces some parameters, but the number of parameters is still lower than the anchor-based models Faster-RCNN and YOLOv3. The FLOPs and detection times of these two models are also higher than those of the model in this paper, and the average precision and average recall of the detection of the model in this paper on the green apple and green persimmon datasets are also significantly higher than those of these two models. In addition, compared with other models, the FLOPs of this model are only about 50% of those of the other models with some improvement in the average precision and recall rate.

5 Conclusion

In order to improve the accuracy of fruit detection in modern orchards, this paper proposes an efficient target detection and

TABLE 3 Comparison results of detection of different models.

Network	Metric			
	AP/%	$AP^{IOU=0.5/\%}$	AR/%	Time/ms
Apple Dataset				
FCOS	57.6	86.6	65.1	50.3
Faster-RCNN	59.2	85.9	65.1	54.5
YOLOv3	59.1	84.3	65.2	19.4
SSD	59.6	86.6	66.2	22.3
FSAF	61.7	87.6	68.5	54.2
ATSS	62.2	88.3	69.3	54.6
Ours	64.8	88.4	72.6	25.6
Persimmon Dataset				
FCOS	69.7	92.3	76.1	50.1
Faster-RCNN	70.7	91.3	76.1	54.3
YOLOv3	70.5	87.9	76.2	18.8
SSD	71.2	91.6	76.4	22.2
FSAF	72.1	92.1	78.1	54
ATSS	72.8	91.6	79.2	54.7
Ours	74.7	92	80.2	26.7

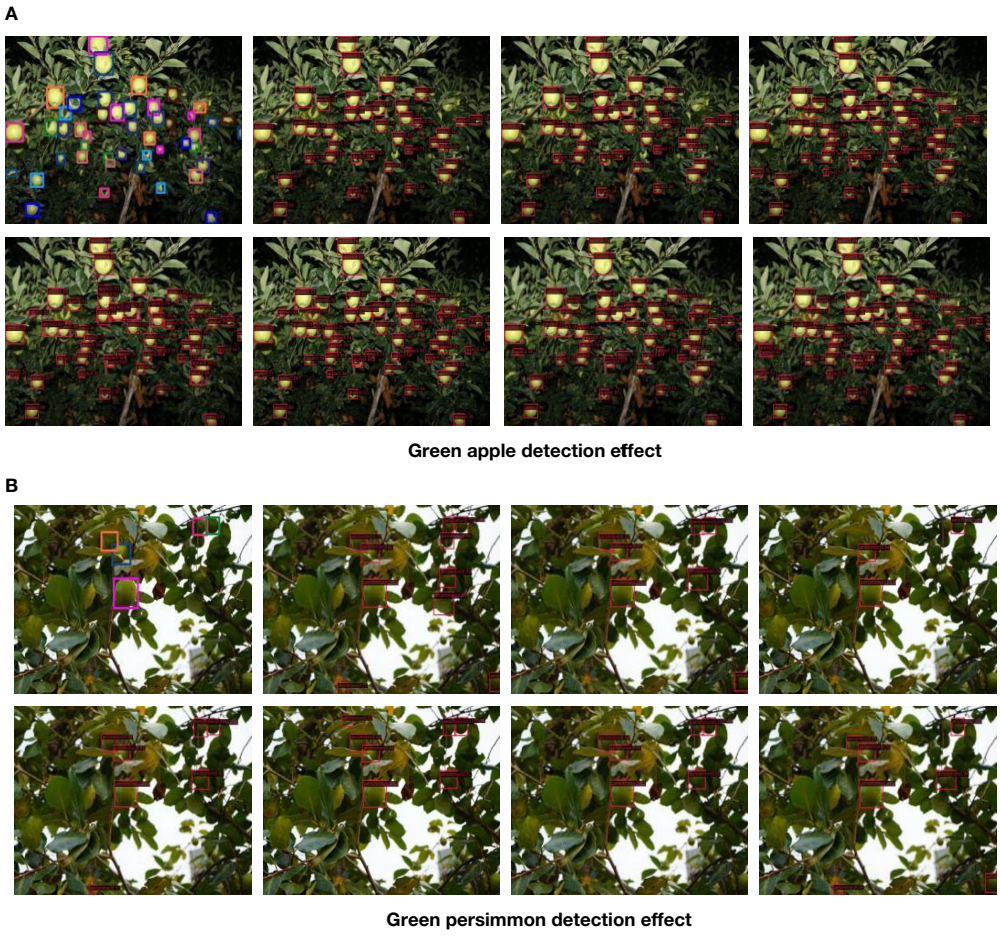


FIGURE 9
Different model detection effect. **(A)** Green apple detection effect. **(B)** Green persimmon detection effect.

TABLE 4 Comparison of assessment metrics of different detection models.

Network	Metric	
	FLOPs/G	Params/M
Faster-RCNN	206.66	41.12
YOLOv3	193.85	61.52
SSD	342.67	24.39
FSAF	202.39	36.01
ATSS	201.41	31.89
Ours	109.29	36.53

recognition method with improved yolox-m. The model uses two datasets, unripe green persimmon and green apple, for training detection. Considering the complex situation of real orchards, the images collected in the dataset include leaf occlusion, fruit overlap and after rain. In this paper, we use Atrous Spatial Pyramid Pooling (ASPP) in the feature pyramid network to increase the receptive field and combine the feature information at different scales to improve the detection accuracy of the model, in addition, in order to mitigate the negative impact of sample imbalance and make the model focus more on positive samples to optimize the updated model parameters. For the loss function, the original binary cross-entropy (BCE) loss is replaced by varifocal loss to better optimize the model, improve the model performance and increase the precision.

The experimental results prove that the average precision, average recall and real-time performance of the model in this paper are better than those of several other models, and the computational complexity is also lower, which can achieve the detection and recognition of fruits accurately and in real time. It meets the needs of agricultural automation equipment. The model achieves a good level of detection on both datasets, however, it also has certain limitations, as follows:

(1) The number of images contained in the dataset used is relatively small due to realistic experimental conditions, and therefore we will consider continuing to expand the dataset.

(2) In order to improve the accuracy of the model, some parameters are introduced in this paper, and we will try to reduce the parameters of the model and reduce the size of the model in the future, while continuing to improve the accuracy.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding authors.

Author contributions

WJ, YX, and XG conceived the idea for the paper; YX, YL, and NP with contributions for data curation; YL and XY wrote the code, designed and conducted the experiments; YX and RJ with contributions for visualization and validation; WJ, YX, and XG with contributions for writing- original draft preparation. All authors contributed to the article and approved the submitted version.

Funding

This work is supported by the Natural Science Foundation of Shandong Province in China (No.: ZR2020MF076); Young Innovation Team Program" of Shandong Provincial University (No.: 2022KJ250); Natural Science Foundation of Jiangsu Province (No.: BK20170256); National Nature Science Foundation of China (No.: 62072289); New Twentieth Items of Universities in Jinan (2021GXRC049); Taishan Scholar Program of Shandong Province of China.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Apolo-Apolo, O. E., Martínez-Guanter, J., Egea, G., Raja, P., and Pérez-Ruiz, M. (2020). Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. *Eur. J. Agron.* 115, 126030. doi: 10.1016/j.eja.2020.126030

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chen, L. C., Papandreou, G., Kokkinos, I., et al. (2017)Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS (Accessed IEEE transactions on pattern analysis and machine intelligence).

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al. (2019). MMDetection: open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17 (9), 2022. doi: 10.3390/s17092022

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430 2021*.

He, Q., Ma, B., Qu, D., Zhang, Q., Hou, X., and Zhao, J. (2013). Cotton pests and diseases detection based on image processing. *Indonesian J. Electrical Eng.* 11 (6), 3445–3450.

Jia, W., Zhang, Y., Lian, J., Zheng, Y., Zhao, D., and Li, C. (2020). Apple harvesting robot under information technology: a review. *Int. J. Adv. Robotic Syst.* 17 (3), 25310.

Jia, W., Zhang, Z., Shao, W., Ji, Z., and Hou, S. (2022b). RS-net: robust segmentation of green overlapped apples. *Precis. Agric.* 23 (2), 492–513.

Johnson, J., Sharma, G., Srinivasan, S., Masakapalli, S. K., Sharma, S., Sharma, J., et al. (2021). Enhanced field-based detection of potato blight in complex backgrounds using deep learning. *Plant Phenomics*, 9835724.

Kang, H., and Chen, C. (2020). Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* 171, 105302. doi: 10.1016/j.compag.2020.105302

Li, C., Adhikari, R., Yao, Y., Miller, A. G., Kalbaugh, K., Li, D., et al. (2020). Measuring plant growth characteristics using smartphone based image analysis technique in controlled environment agriculture. *Comput. Electron. Agric.* 168, 105123. doi: 10.1016/j.compag.2019.105123

Lin, T. Y., Dollár, P., Girshick, R., et al. (2017a). Feature pyramid networks for object detection (Accessed Proceedings of the IEEE conference on computer vision and pattern recognition).

Lin, T. Y., Goyal, P., Girshick, R., et al. (2017b). Focal loss for dense object detection (Accessed Proceedings of the IEEE international conference on computer vision).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft Coco: common objects in context. *Comput. Vision–ECCV*, 740–755. doi: 10.1007/978-3-319-10602-1_48

Linker, R., Cohen, O., and Naor, A. (2012). Determination of the number of green apples in RGB images recorded in orchards. *Comput. Electron. Agric.* 81, 45–57. doi: 10.1016/j.compag.2011.11.007

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. *Proc. IEEE Conf. Comput. Vision Pattern recognit.*, 8759–8768. doi: 10.1109/CVPR.2018.00913

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference*, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21–37). Springer International Publishing.

Moallem, P., Serajoddin, A., and Pourghassem, H. (2017). Computer vision-based apple grading for golden delicious apples based on surface features. *Inf. Process. Agric.* 4 (1), 33–40. doi: 10.1016/j.inpa.2016.10.003

Mu, Y., Chen, T. S., Ninomiya, S., and Guo, W. (2020). Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors* 20 (10), 2984. doi: 10.3390/s20102984

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32.

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement[J]. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: towards real-time object detection with region proposal network. *Adv. Neural Inf. Process. Syst.* 28.

Sullivan, A., and Lu, X. (2007). ASPP: a new family of oncogenes and tumour suppressor genes. *Br. J. Cancer* 96 (2), 196–200. doi: 10.1038/sj.bjc.6603525

Sun, M., Xu, L., Chen, X., Ji, Z., Zheng, Y., and Jia, W. (2022). Bfp net: balanced feature pyramid network for small apple detection in complex orchard environment. *Plant Phenomics* 2022. doi: 10.34133/2022/9892464

Tang, Y., Zhou, H., Wang, H., and Zhang, Y. (2023). Fruit detection and positioning technology for a camellia oleifera c. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Syst. Appl.* 211, 118573.

Tian, Y., Duan, H., Luo, R., Zhang, Y., Jia, W., Lian, J., et al. (2019). Fast recognition and location of target fruit based on depth information. *IEEE Access* 7, 170553–170563. doi: 10.1109/ACCESS.2019.2955566

Tian, Z., Shen, C., Chen, H., and He, T. (2019). Fcos: fully convolutional one-stage object detection (Accessed Proceedings of the IEEE/CVF international conference on computer vision).

Triki, A., Bouaziz, B., Gaikwad, J., and Mahdi, W. (2021). Deep leaf: mask r-CNN based leaf detection and segmentation from digitized herbarium specimen images. *Pattern Recognit. Lett.* 150, 76–83. doi: 10.1016/j.patrec.2021.07.003

Wang, D., and He, D. (2019). Recognition of apple targets before fruits thinning by robot based on r-FCN deep convolution neural network. *Trans. CSAE* 35 (3), 156–163.

Wang, Z. F, Jia, W. K., Mou, S. H., Hou, S. J., Yin, X., and Ji, Z. (2021). KDC: a green apple segmentation method. *Spectrosc. Spectral Anal.* 41 (9), 2980–2988.

Wang, C. Y., Liao, H. Y. M., Wu, Y. H., et al. (2020). CSPNet: a new backbone that can enhance learning capability of CNN (Accessed Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops).

Wang, Z., Zhang, Z., Lu, Y., Luo, R., Niu, Y., Yang, X., et al. (2022). SE-COTR: a novel fruit segmentation model for green apples application in complex orchard. *Plant Phenomics* 2022, 0005. doi: 10.34133/plantphenomics.0005

Wu, G., Li, B., Zhu, Q., Huang, M., and Guo, Y. (2020). Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy. *Comput. Electron. Agric.* 174, 105475. doi: 10.1016/j.compag.2020.105475

Yu, J., Jiang, Y., Wang, Z., et al. (2016). Unitbox: an advanced object detection network (Accessed Proceedings of the 24th ACM international conference on Multimedia).

Zhang, H., Wang, Y., Dayoub, F., et al. (2021). Varifocalnet: an iou-aware dense object detector (Accessed Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition).

Zhang, S., Chi, C., Yao, Y., Lei, Z., and Li, S. Z. (2020). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9759–9768.

Zhu, C., He, Y., and Savvides, M. (2019). Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 840–849.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 14 July 2023
DOI 10.3389/fpls.2023.1227011

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Fanan Wei,
Fuzhou University, China
Wenfeng Liang,
Shenyang Jianzhu University, China

*CORRESPONDENCE
Guangyuan Zhang
xdzhanggy@163.com

RECEIVED 22 May 2023
ACCEPTED 29 June 2023
PUBLISHED 14 July 2023

CITATION
Guan H, Fu C, Zhang G, Li K, Wang P and Zhu Z (2023) A lightweight model for efficient identification of plant diseases and pests based on deep learning.
Front. Plant Sci. 14:1227011.
doi: 10.3389/fpls.2023.1227011

A lightweight model for efficient identification of plant diseases and pests based on deep learning

Hongliang Guan, Chen Fu, Guangyuan Zhang*, Kefeng Li, Peng Wang and Zhenfang Zhu

School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan, China

Plant diseases and pests have always been major contributors to losses that occur in agriculture. Currently, the use of deep learning-based convolutional neural network models allows for the accurate identification of different types of plant diseases and pests. To enable more efficient identification of plant diseases and pests, we design a novel network architecture called Dise-Efficient based on the EfficientNetV2 model. Our experiments demonstrate that training this model using a dynamic learning rate decay strategy can improve the accuracy of plant disease and pest identification. Furthermore, to improve the model's generalization ability, transfer learning is incorporated into the training process. Experimental results indicate that the Dise-Efficient model boasts a compact size of 13.3 MB. After being trained using the dynamic learning rate decay strategy, the model achieves an accuracy of 99.80% on the Plant Village plant disease and pest dataset. Moreover, through transfer learning on the IP102 dataset, which represents real-world environmental conditions, the Dise-Efficient model achieves a recognition accuracy of 64.40% for plant disease and pest identification. In light of these results, the proposed Dise-Efficient model holds great potential as a valuable reference for the deployment of automatic plant disease and pest identification applications on mobile and embedded devices in the future.

KEYWORDS

plant diseases and pests, deep learning, lightweight model, dynamic decay strategy, transfer learning

1 Introduction

Plant diseases and pests can severely disrupt the normal growth and development of crops, leading to reduced crop yields and negatively impacting farmers' income. Moreover, they can have severe implications for the supply of grains and agricultural products in the market, potentially resulting in a significant food crisis. Prioritizing the prevention and control of plant diseases and pests is essential in agricultural production, as effective

management of these issues holds significant importance for ensuring food security, improving farmers' income, and promoting sustainable agricultural development (Elnahal et al., 2022; Sehrawat et al., 2022).

Plant diseases and pests arise from a combination of environmental factors and pathogen invasion. Pathogens, which include fungi, bacteria, and viruses, are the fundamental cause of plant diseases. They can enter plant organisms through different transmission pathways, leading to the development of plant diseases and pests (Barragán-Fonseca et al., 2022). Environmental changes are also a critical factor in the onset and spread of plant diseases and pests (Canassa et al., 2020). Most plant diseases and pests exhibit distinct characteristics depending on the disease type, and accurately identifying the disease type based on these characteristics is crucial in effectively preventing and controlling plant diseases and pests.

In the past, people relied on visual observation of plant leaves and fruits to determine the presence of plant diseases and pests. They identified the type of plant disease based on the distinctive features exhibited by affected plants. However, this manual identification method heavily relied on individual experience, resulting in high labor costs and low efficiency. Subsequently, with the advancement of computer technology, machine learning techniques were introduced to aid in the identification of plant diseases and pests. At first, machine learning utilized computer vision to analyze the morphological changes in diseased leaves or fruits and extract the pathological features of plant diseases. The computer then made predictions about the disease type based on the obtained features. However, machine learning-based methods for automated plant disease and pest identification faced limitations in terms of accuracy and generalizability. The use of rule-based image processing techniques to extract disease features led to sensitivity to image quality, as image noise could greatly affect the final results (Behmann et al., 2015; Wani et al., 2022).

In recent years, deep learning has made significant breakthroughs and has taken the forefront as become a research direction in computer vision, particularly in the field of agriculture. In this context, the use of deep learning for plant disease and pest type identification has emerged as an important application and research area (Liu and Wang, 2021). Currently, deep learning-based models for plant disease and pest identification are exhibiting a trend toward increased accuracy, smaller model sizes, faster training speeds, and stronger transferability. In response to this trend, this paper proposes a lightweight model for the efficient identification of plant diseases and pests based on deep learning, called the Dise-Efficient model.

The main contributions of this study are as follows:

1. Proposing the Dise-Efficient model, a novel deep learning-based model for efficient and accurate identification of plant diseases and pests.
2. Demonstrating how the number of convolutional layers and the size of the convolution kernel affect the accuracy of the Dise-Efficient model in identifying plant diseases and pests.
3. Training the Dise-Efficient model using the dynamic learning rate decay strategy and experimentally demonstrating that this strategy can significantly improve the accuracy of the model.
4. Excrementally validating the Dise-Efficient model has a good transfer learning ability.

2 Related work

The advancement of deep learning technology has led to rapid progress in the field of plant pest detection. The research on the automatic identification of plant pests and diseases has witnessed an evolution of convolutional neural network (CNN) models from small to large, resulting in continual improvement in accuracy rates. More recently, however, there has been a shift toward developing more lightweight models that maintain high accuracy rates while having smaller model sizes.

2.1 Convolutional neural network models

Following the proposal of the AlexNet model by Krizhevsky et al. (Krizhevsky et al., 2017), there has been rapid development of CNNs in the field of computer image recognition. Subsequently, CNN models began to be applied to the agricultural field. According to the experimental results presented by Mohanty et al. (Mohanty et al., 2016), the AlexNet model can achieve an accuracy rate of 99.28% in identifying plant diseases and pests on the Plant Village public dataset. This indicates the effectiveness of CNN models in identifying plant diseases and pests. He et al. (He et al., 2016) proposed a ResNet model, which involved adding an increased number of convolutional layers to a CNN model, as an improvement to the accuracy of image recognition. Following this, researchers have used the concept of the ResNet model to design CNN models with deep convolutional layers across various image recognition applications. The aim is to improve the accuracy of CNN models in identifying different image types. Fuentes et al. (Fuentes et al., 2019) used ResNet50 as the feature extractor in the SSD target detection framework to identify potato diseases, resulting in an accuracy rate of 85.98%. Similarly, Kumar et al. (Kumar et al., 2020) implemented the ResNet34 model to identify 14 different crop diseases on the Plant Village dataset, with a high accuracy rate of 99.40%.

As CNN models achieved high accuracy rates, researchers started exploring the issue of making the model lightweight. The emergence of lightweight CNN models such as MobileNet and EfficientNet has led the research on plant disease image recognition towards the development of lightweight CNN models (Howard et al., 2019; Tan and Le, 2019). Lightweight CNN models usually use depthwise (DW) separable convolution (DW) to replace ordinary convolution, reducing model and parameter size. However, this approach may result in a decline in recognition accuracy. To deal with this problem, a common approach is to add a squeeze and

excitation (SE) block (Hu et al., 2018) to lightweight models to improve their accuracy in identifying image types. Many lightweight CNN model structures, such as the EfficientNetV2 model (Tan and Le, 2021), have been proposed based on this concept. SE blocks are often added to ensure the accuracy of the model. Kamal et al. (Kamal et al., 2019) used the original MobileNet model to train their proposed model on the Plant Village dataset, achieving an accuracy rate of 98.65%. However, when compared to traditional CNN models such as AlaxNet and VGG, there was a decrease in the accuracy rate by approximately 1%. Chen et al. (Chen et al., 2021) embedded the SE block into MobileNet and trained it on the Plant Village dataset, achieving an accuracy rate of 99.78%, which surpassed those obtained by many traditional CNN models trained on this dataset for plant disease type identification.

2.2 Learning strategies

Initially, researchers used a fixed learning rate to train the CNN model, which caused the accuracy of the model to be heavily dependent on the learning rate parameter. Later, many researchers improved the training speed and identification accuracy of the model by proposing strategies for adjusting the learning rate parameter. These strategies can be categorized into two main groups: adaptive learning rate and learning rate decay. Among them, the Adam optimizer, which utilizes an adaptive learning rate strategy, is widely used in deep learning and is known for its effectiveness. Loshchilov et al. (Loshchilov and Hutter, 2017) proposed a cosine processing strategy to dynamically adjust the learning rate. He et al. (He et al., 2019) applied the cosine learning rate decay strategy to train the ResNet50 model, resulting in an improvement of approximately 2% in model accuracy.

Inspired by the successful application of dynamic learning rate, this paper applies the cosine-type progressive learning rate decay strategy to the Dise-Efficient model to improve the model's accuracy in identifying plant diseases and pests. Formula (1) outlines the dynamic learning rate decay strategy proposed in this paper:

$$lr = (1 + \cos\frac{\pi x}{n}) \cdot (1 - lrf) + lrf \quad (1)$$

where lr represents the learning rate of the next round; lrf represents the learning rate of the last round; x represents the learning rate of the current round; n represents the maximum number of iterations.

2.3 Transfer learning

Recent CNN models have shown high accuracy rates of over 95% on the Plant Village plant disease dataset (Ahmad et al., 2022). However, the performance of these CNN models on the IP102 large-scale plant pest dataset is lower than expected, with traditional CNN models achieving an accuracy rate of around 50% (Ren et al., 2019; Wu et al., 2019; Nurfauzi et al., 2023). Despite the improvements made to the CNN models, their accuracy on this dataset is only slightly over 60% (Nanni et al., 2020). This can be explained by the fact that the IP102 dataset is a plant pest dataset that reflects the actual environment, with images possessing more complex backgrounds and fewer samples for each pest category. Therefore, conducting deep learning model training utilizing transfer learning is an effective solution to address the issue of limited data samples for certain pest categories in the IP102 dataset.

Transfer learning involves transferring the knowledge or patterns learned from existing labeled training data to improve learning in a new target field (Weiss et al., 2016). Incorporating transfer learning in the deep learning model training process not only accelerates the model training process but also facilitates the acquisition of a more accurate deep learning model through the fine-tuning of the pre-trained model (Zhu et al., 2023). In current research on plant disease and pest identification, many researchers have applied transfer learning to CNN models to improve both the training speed of the model and the accuracy of identification (Thenmozhi and Reddy, 2019; Liu et al., 2022).

3 Experiments

3.1 Dataset and environment

Plant Village is a public plant disease dataset (Hughes and Salathé, 2015), containing 54,303 images of healthy or diseased leaves categorized into 38 different groups from 9 crop species. Researchers often utilize this dataset in studies related to the identification of plant diseases and pests, as well as for developing models aimed at identifying various types of plant pests.

IP102 is a large-scale dataset developed for identifying pests (Wu et al., 2019), comprising more than 75,000 images categorized into 102 types, exhibiting a natural long-tail distribution. IP102 has a hierarchical taxonomy that groups pests that primarily affect one particular agricultural product into the same upper category. This dataset is often used in research aimed at identifying plant pests and is implemented in this study as a training dataset for the plant pest identification model.

The Mini-ImageNet dataset (Satorras and Estrach, 2018) comprises 100 common categories selected from the ImageNet dataset, with each category containing 600 images and a total of 60,000 images. Given that this dataset is often used in the pre-training of small sample learning models, it is employed as the dataset for the pre-trained model in the present study.

Before it is applied for model training, the dataset must be split into different sets. In this study, we divided the Plant Village and IP102 datasets into a training set, a validation set, and a test set at a ratio of 3:1:1. Meanwhile, the Mini-ImageNet dataset was used for the pre-training model, so it was divided into a training set and a validation set at a ratio of 4:1. Table 1 shows the number of images present in the different sets of each divided dataset.

During the training phase of this experimental model, we employed the Tencent Cloud GN7-8-core 32G cloud server that supports GPU computing tasks. The GPU model used was Nvidia Tesla T4, featuring 16 GB video memory and 32 GB internal

TABLE 1 Number of images in different sets of each divided dataset.

Dataset	Training set/sheet	Validation set/sheet	Test set/sheet	Total/sheet
Plant Village	36,892	12,297	12,297	61,486
IP102	45,132	15,043	15,043	75,218
Mini-ImageNet	48,000	12,000	/	60,000

memory, and the operating system was Ubuntu Server 20.04 LTS 64-bit with a Cuda version of 11.2.

3.2 Model

Drawing upon our previous research experience, we thoroughly studied the structures and principles of the classic ResNet model and the lightweight EfficientNetV2 model. After careful consideration, we decided to use the residual block of ResNet to replace a portion of the MBConv block and Fused-MBConv block in the EfficientNetV2 model. Finally, we managed to design a lightweight CNN network model that can efficiently identify various types of plant diseases and pests: the Dise-Efficient model. The framework of this model is shown in Figure 1.

The Residual block is regarded as the basic residual block of ResNet18. It features three convolution kernels with a size of 3×3, along with a shortcut connection. The residual block can add the original feature map to the feature map resulting from the convolution process to obtain a new feature map. Because the image feature distribution of diseased crop leaves is relatively simple, issues of gradient explosion and gradient disappearance may arise due to the continuous deepening of the convolutional layer. These problems can be addressed by incorporating a Residual layer, which allows for the extraction of deep features from diseased crop leaf images. A detailed illustration of the Residual block's structure is provided in Figure 2.

The MBConv block represents an improvement based on the residual block. First, the ordinary convolution operation was replaced with a DW separable convolution operation. This involved adding two convolution kernels with a size of 1×1 into the residual structure, thereby realizing a DW separable convolution operation. Subsequently, a compression and excitation layer was added to enhance the self-attention mechanism of the model and mitigate the reduction in accuracy caused by a decrease in the number of parameters. As a result of these adjustments, the prediction accuracy of the model was improved. A detailed illustration of the MBConv block's structure is provided in Figure 3.

The Fused-MBConv block is a modified version of the MBConv block, which involves removing the first convolutional layer for dimensionality increment and the data squeezing and excitation layer in the MBConv module. The block was used to determine whether DW separable volumes are to be performed based on the expansion coefficient point-by-point operations of the product. A detailed illustration of the Fused-MBConv block's structure is provided in Figure 4.

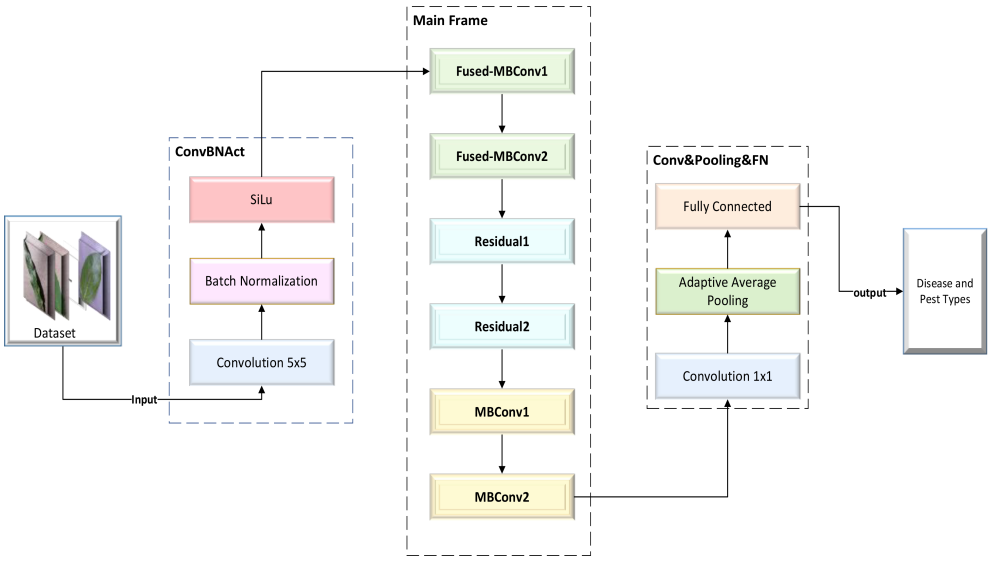


FIGURE 1
Framework of the Dise-Efficient model.

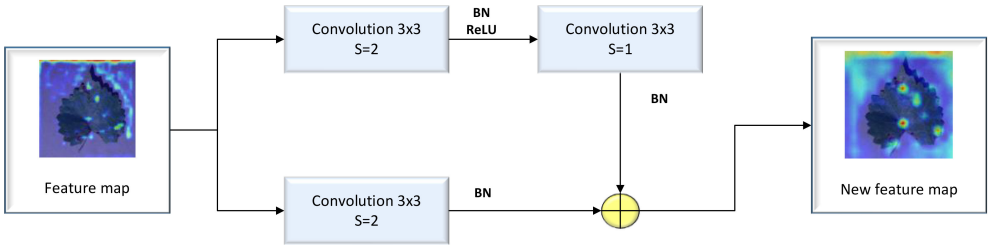


FIGURE 2
Structure of the Residual block.

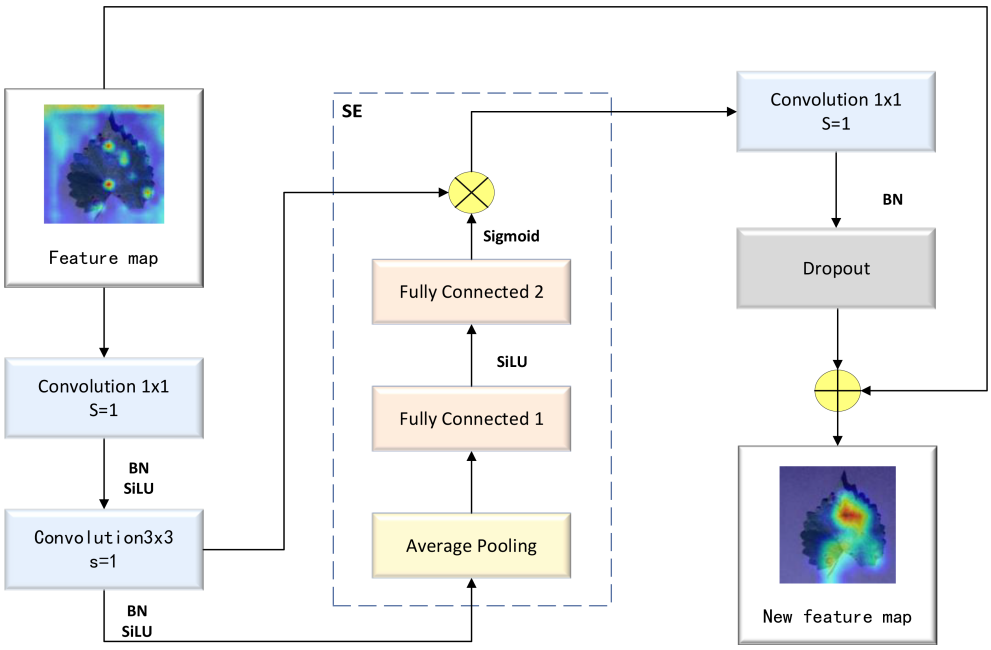


FIGURE 3
Structure of the MBConv block.

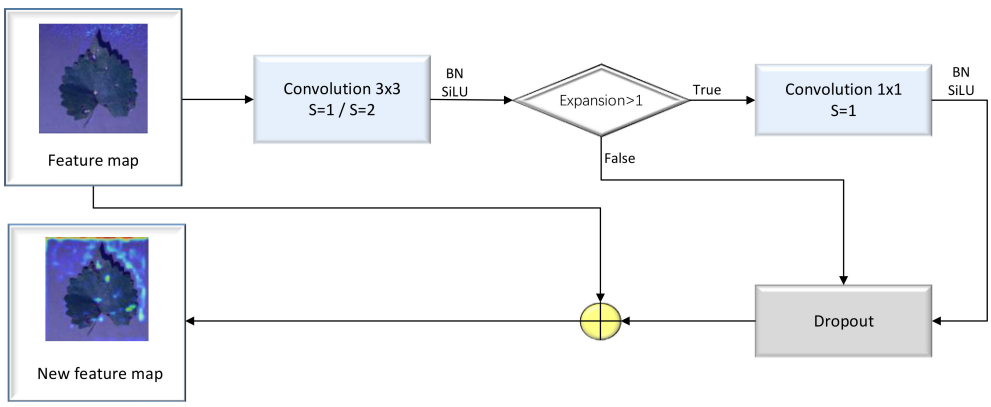


FIGURE 4
Structure of the Fused-MBCov block.

3.3 Experimental design

3.3.1 Experimental comparison of convolutional layers of different models

To verify the effect of different convolutional layers on the accuracy of the Dise-Efficient model in identifying plant disease types, we designed a baseline model called Dise-Efficient-B0-N, also referred to as B0-N. In this baseline model, each convolutional layer consists of two layers. In addition, we developed the B0-S model, which is smaller than the B0-N model, and the B0-L model, which is larger than the B0-N model.

In the experiment, we trained the B0-N, B0-S, and B0-L models on the Plant Village dataset. After the training, we compared the accuracy of the three models in identifying plant disease types. The main parameters of the three models are presented in Table 2.

3.3.2 Experimental comparison of different learning strategies

The learning strategy designed in this study is comprised of a stochastic gradient descent (SGD) optimizer, which utilizes momentum to improve the model training process. Additionally, we implemented a cosine dynamic decay strategy for the learning rate, which started at 0.01 and decayed in a cosine manner as the number of training rounds increased. Formula (1) illustrates the dynamic decay strategy for the learning rate, with the final learning rate being 0.001. The learning rate decay result is depicted in the form of a curve in Figure 5.

Generally, the Adam optimizer provides better optimization performance for model training than the SGD optimizer combined with the momentum learning strategy. However, our experiments revealed that the model generally achieved higher accuracy in identifying disease types when the SGD optimizer was implemented in combination with the cosine dynamic decay strategy, as designed in this paper, compared to when the Adam optimizer was used.

To verify whether the cosine dynamic decay learning strategy can improve the accuracy of the automatic plant disease and pest identification model, we conducted experiments on the Plant Village dataset, using the B0-N, B0-S, and B0-L models for comparative analysis. In experimental group 1, we implemented the Adam optimizer commonly used in CNN model training, while setting the learning rate parameter to a fixed value of 0.001. In experimental group 2, we utilized the SGD optimizer with a fixed learning rate. In the control group, we employed the SGD optimizer with a cosine dynamic decay strategy that gradually reduced the learning rate from 0.01 to 0.001 based on formula (1). The specific experimental parameters are listed in Table 3.

3.3.3 Experimental comparison of convolution kernel sizes of different models

Generally, smaller convolution kernels tend to capture finer-grained features, while larger ones are better suited for capturing more macroscopic features (Szegedy et al., 2015). Therefore, by changing the size of the convolution kernel and observing how the accuracy of the model accordingly, we can understand the effect of different feature scales on the performance of the model. With this in mind, we changed the size of the module convolution kernel to investigate the effect of replacing a small convolution kernel with a large one on each module's performance.

In this experiment, we constructed models from Dise-Efficient-B1 to Dise-Efficient-B7, all based on the Dise-Efficient-B0-N (abbreviated as B0) model. Specifically, the B1 to B7 models were designed with 5x5 large convolution kernels to replace the 3x3 small convolution kernels of different modules. Table 4 shows the details of the convolution kernel replacements, and other parameters remain unchanged from the B0 model.

3.3.4 Experimental comparison of transfer learning abilities of different models

The migration learning process consists of two phases: pre-training and migration learning. In the pre-training phase of this experiment, we used the cosine dynamic learning rate decay strategy designed in this study to train the B0 and B2 models, as shown in Table 4, on the Mini-ImageNet dataset, generating pre-training models for B0 and B2. Finally, the pre-trained model weights were uploaded in the IP02 dataset for use in the transfer learning process, as illustrated in Figure 6.

In this study, two transfer learning methods were used to compare the experimental results. The first one involved freezing the feature layer of the pre-trained model before performing transfer learning. The second one involved using the full set of

TABLE 2 Parameters of the B0-N, B0-S, and B0-L models.

Block	B0-N Layers	B0-S Layers	B0-L Layers	Stride	Number of convolution kernels	Dropout	Expansion
ConvBNAct	1	1	1	2	32	0	–
Fused-MBConv1	2	1	3	1	32	0	1
Fused-MBConv2	2	1	3	2	64	0	4
Residual1	2	1	3	2	64	0	–
Residual2	2	1	3	2	128	0	–
MBConv1	2	1	3	1	160	0.25	6
MBConv2	2	1	3	2	256	0.25	6

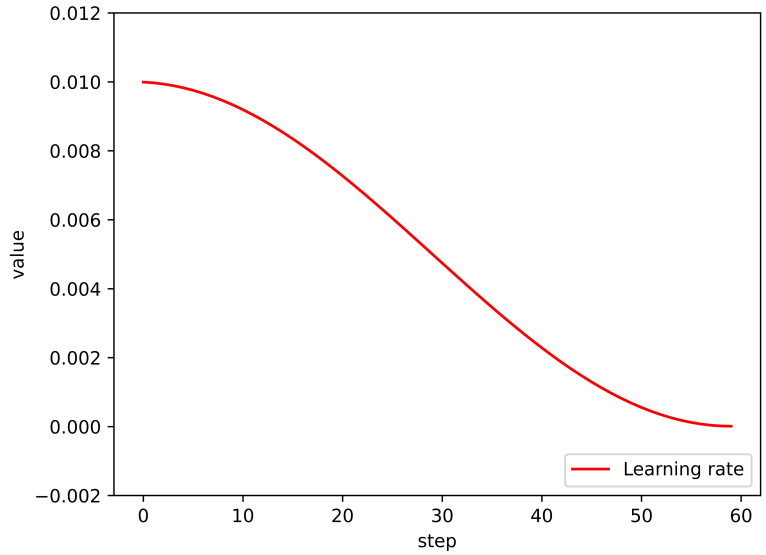


FIGURE 5
Learning rate decay result.

TABLE 3 Experimental conditions and parameters.

	Experimental group 1	Experimental group 2	Control group
Learning strategy	Fixed learning rate	Fixed learning rate	Cosine dynamic attenuation
Optimizer	Adam	SGD	SGD
Momentum	/	0.9	0.9
Initial learning rate (lr)	0.001	0.001	0.01
Final learning rate (lrf)	0.001	0.001	0.001
Epochs	60	60	60
Batch size	64	64	64

parameters for direct transfer learning. Details of the specific experimental design are shown in Table 5.

4 Results and analysis

4.1 Validity of the model

To evaluate the performance of the proposed Dise-Efficient model in identifying plant pest types, we trained the baseline model Dise-Efficient-B0 on the Plant Village dataset. This experiment was conducted under the experimental conditions and parameters for the experimental groups in Table 3. We compared the accuracy rate obtained by the final model on the test set with the accuracy rates of other CNN models used for agricultural pest detection. The comparison results are presented in Table 6.

From the results in Table 6, it can be seen that the Dise-Efficient-B0 model achieved the highest accuracy rate in identifying plant disease types on the Plant Village dataset, reaching 99.71%. The model delivered a 61.84% accuracy rate in identifying plant pest types on the IP102 dataset, which was only lower than the accuracy rate of a previously proposed model (Nanni et al., 2020). These findings demonstrated that the Dise-Efficient model has a strong ability in identifying various types of plant diseases and pests. Therefore, this model holds substantial research and practical value for the identification of plant diseases and pests.

4.2 Effect of the number of convolutional layers on model performance

To investigate the effect of the number of convolutional layers on the accuracy of plant disease and pest identification models, we experimentally implemented the B0-N, B0-S, and B0-L models presented in Table 1 under the experimental conditions and parameters for the experimental groups in Table 3. Finally, we obtained the indexes of the models in the experimental groups, as shown in Table 7.

10.3389/fpls.2023.1227011

TABLE 4 Number of model layers and convolution kernel size.

Block	B0	B1	B2	B3	B4	B5	B6	B7
BNConvAct	3x3							
Fused-Conv1	3x3	5x5	3x3	3x3	5x5	5x5	3x3	5x5
Fused-Conv2	3x3	5x5	3x3	3x3	5x5	5x5	3x3	5x5
Residual1	3x3	3x3	5x5	3x3	5x5	3x3	5x5	5x5
Residual2	3x3	3x3	5x5	3x3	5x5	3x3	5x5	5x5
MBConv1	3x3	3x3	3x3	5x5	3x3	5x5	5x5	5x5
MBConv2	3x3	3x3	3x3	5x5	3x3	5x5	5x5	5x5

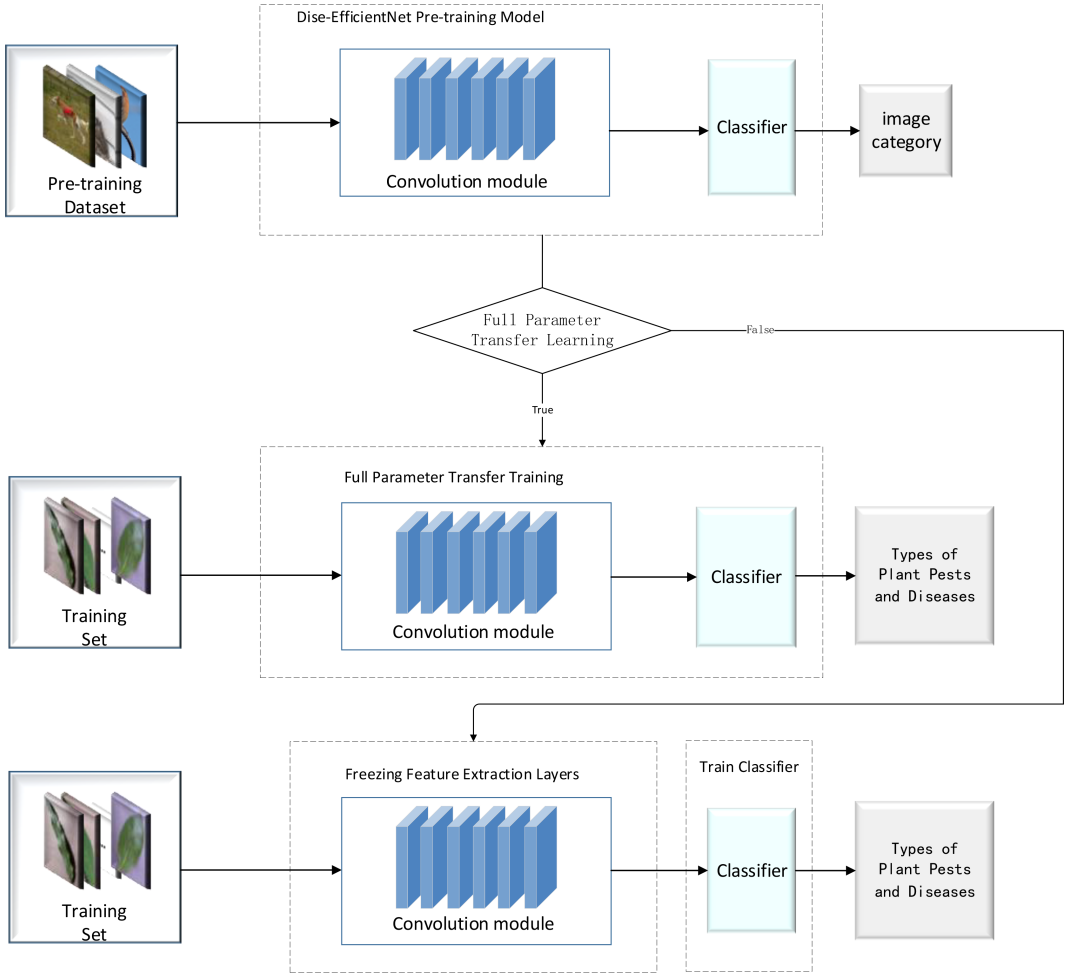


FIGURE 6
Flowchart of transfer learning.

The above results indicate that the B0-N model is the most accurate in identifying plant disease types, achieving an accuracy rate of 99.71%. Furthermore, the B0-S model is the smallest in size, at only 5.86 MB, but delivers a 0.16% lower accuracy rate than the B0-N model. In contrast, the B0-L model has the largest size, measuring 20.80 MB.

Through an analysis of the above results, we found that the B0-S model has one less convolutional layer in each module when compared to B0-N, so the model size of B0-S is smaller than that of B0-N; the B0-L model has one more layer convolutional layer in each module when compared to B0-N, so the model size of B0-L is larger than that of B0-N. Hence, the number of convolutional layers

TABLE 5 Experimental design of transfer learning.

Model name		Original model	Feature layer freezing	Full parameter transfer
B0	B2	√		
B0-Freeze-TF	B2-Freeze-TF		√	
B0-TF	B2-TF			√

TABLE 6 Comparison between Dise-Efficient and other plant disease and pest identification models.

Dataset	Research paper	Model name	Accuracy (%)	Dataset	Research paper	Model name	Accuracy (%)
Plant Village	Sladojevic et al., 2016	CaffeNet	98.21	IP102	Nurfauzi et al., 2023	EfficientNetV2-B0	51.00
	Gokulnath, 2021	LF-CNN	98.93		Ren et al., 2019	FR-ResNets	55.24
	Ganatra and Patel, 2020	Inception V4	98.30		Lin et al., 2023	GPA-Net	56.90
	Bedi and Gole, 2021	Models in the research	99.38		Nanni et al., 2020	**Models in the research**	**61.93**
	Ours	**Dise-Efficient-B0-N**	**99.71**		Ours	Dise-Efficient-B0	61.48

Bold values mean the line with the best model evaluation index.

TABLE 7 Indexes of different models in the experimental groups.

Index	Dise-Efficient-B0-N	Dise-Efficient-B0-S	Dise-Efficient-B0-L
Accuracy/%	**99.71**	99.55	99.60
Model size/MB	13.30	**5.86**	20.80

Bold values mean the line with the best model evaluation index.

will directly affect the model size – the more convolutional layers, the larger the model size.

4.3 Effect of dynamic learning strategy on model performance

Based on the experimental design in Table 3, we obtained the accuracy and model size of the Dise-Efficient model used for identifying plant disease types in experimental group 1, experimental group 2, and the control group on the Plant Village test set. The results are shown in Table 8.

We implemented the Adam optimizer with a fixed learning rate for experimental group 1 and the SGD optimizer with a fixed learning rate for experimental group 2. From Table 8, it can be seen that for the same model trained under a fixed learning rate strategy, using the Adam optimizer for training leads to higher accuracy rates compared to using the SGD optimizer. In the control group, we used the SGD optimizer in combination with the cosine dynamic learning decay strategy to train the model, resulting in a higher accuracy in identifying plant disease types than the model trained under the conditions and parameters for experimental group 1. It can be concluded that incorporating a cosine dynamic learning rate decay strategy into the model training process can improve the model's accuracy in identifying plant diseases and pests.

4.4 Effect of convolution kernel size on model performance

The experiment was conducted based on the B0-N model (B0 for short), which had its convolution kernel replaced according to the design in Table 3, resulting in the creation of models B1 to B7. Models B0 to B7 were trained on the IP102 plant pest dataset utilizing the experimental conditions and parameters for the control group outlined in Table 3. The trained model's accuracy and other indexes of these models are presented in Table 9.

Based on the abovementioned experimental findings, it is evident that replacing a small-sized ordinary convolution kernel with a larger one usually improves the accuracy of the Dise-Efficient model in identifying plant pest types. However, replacing a small-sized DW separable convolution kernel with a larger one negatively affects the model's accuracy in identifying plant pest types.

It can be seen from Figure 1 that the Residual block of the Dise-Efficient model is the only one utilizing a common convolution kernel, while the MBConv and Fused-MBConv blocks use a DW

TABLE 8 Model indexes for comparison of experimental results.

Index	B0-N	B0-S	B0-L
Accuracy for experimental group 1 (%)	99.71	99.55	99.60
Accuracy for experimental group 2 (%)	99.27	99.19	99.51
Accuracy for control group (%)	**99.81**	**99.77**	**99.82**

Bold values mean the line with the best model evaluation index.

TABLE 9 Indexes of Dise-Efficient-B0 to B7 models.

Model	Accuracy (%)	Increase in accuracy (%)	Model size (MB)	Increase in size (MB)
Dise-Efficient-B0	61.48	0	**13.3**	0
Dise-Efficient-B1	61.24	-0.24	15.0	+1.7
Dise-Efficient-B2	**61.84**	**+0.36**	16.1	+2.8
Dise-Efficient-B3	61.39	-0.09	13.9	+0.6
Dise-Efficient-B4	61.32	-0.16	17.5	+4.2
Dise-Efficient-B5	60.75	-0.73	15.3	+2.0
Dise-Efficient-B6	61.32	-0.16	16.4	+3.1
Dise-Efficient-B7	61.45	-0.03	17.8	**+4.5**

Bold values mean the line with the best model evaluation index.

separable convolution kernel. From the convolution kernel sizes set for the different blocks of each model in Table 4, it can be seen that Dise-Efficient-B2 only replaces the small convolution kernel with a larger one in its Residual block. Consequently, this model experiences a substantial improvement in identifying plant pest types, which is evidenced by a peak accuracy rate of 61.84%. As for models B1 and B3, they only replace the DW convolution kernel in their Fused-MBConv and MBConv blocks. As a result, both of these models experience varying degrees of reductions in accuracy.

Table 8 shows that the Dise-Efficient-B5 model delivers the lowest accuracy rate, likely due to its use of a larger DW convolution kernel in the place of a smaller one in its Fused-MBConv and MBConv blocks. This replacement caused the model's accuracy in identifying plant pest types to experience the largest drop. Additionally, models B4, B6, and B7 all replace smaller DW convolution kernels with larger ones, leading to varying degrees of reductions in the accuracy in identifying plant pest types.

In terms of the number of parameters, a DW convolution kernel of the same specification has fewer parameters than the ordinary convolution kernel. Therefore, replacing the ordinary convolution kernel with a larger one will increase the size of the model compared to replacing a DW convolution kernel. As a result, as illustrated in Table 8, B2 experiences a more significant increase in model size when compared to B1 and B3. This can be explained by the fact that B2 replaces the ordinary convolution kernel with a larger one, while B1 and B3 replace simply DW convolution kernels. Similarly, model B4 only replaces DW convolution kernels in the Fused-MBConv and MBConv blocks without replacing the ordinary convolution kernel, leading to a smaller increase in model size compared to B4, B6, and B7.

Through the above analysis, it can be concluded that in the application of the Dise-Efficient model for identifying plant pests, replacing the convolution kernel in the Residual module with a larger one can improve the model's accuracy in plant pest type identification, although this improvement comes at the cost of increased model size. In contrast, while replacing a smaller DW convolution kernel with a larger one only causes a small increase in model size, it results in a reduction in accuracy. Therefore, sacrificing a lightweight cost for greater accuracy improvement could be a meaningful research direction to explore.

4.5 Application of transfer learning

Based on the experimental design in Table 5, the accuracy of the model during the transfer learning process on the IP102 dataset is depicted in Figure 7, and the experimental results are summarized in Table 10.

It can be seen from Figure 7 that at the beginning of training, the transfer learning model for plant pest identification delivered a higher accuracy rate than the original model. After the model training was completed, the transfer learning model with a frozen special feature layer significantly outperformed the prototype in terms of accuracy when it comes to identifying plant pests. In other words, the transfer learning training process gave the model a much stronger ability to identify plant pests accurately.

According to Table 10, the transfer learning effect of B0 is superior to that of B2. When under the same transfer learning conditions, the transfer learning model obtained through B0 exhibited higher accuracy rates and faster training speeds than

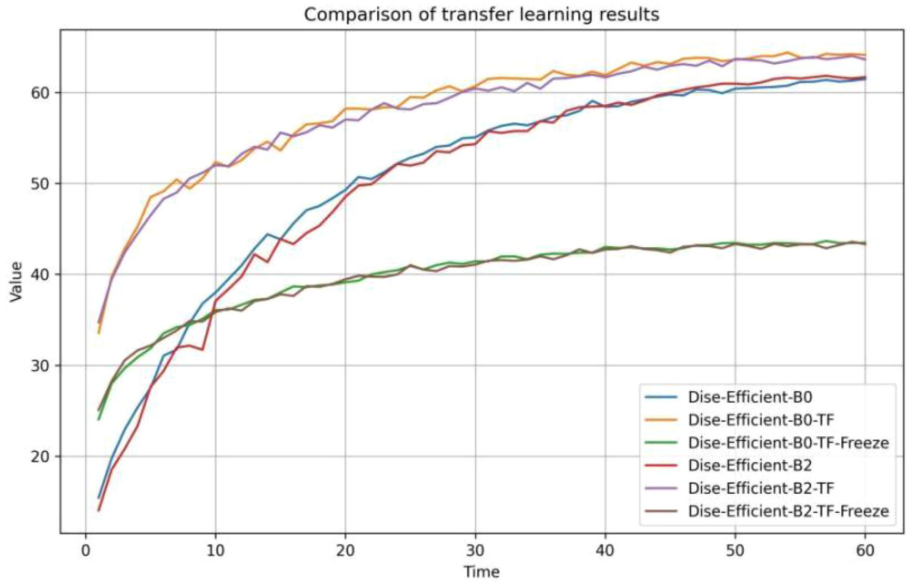


FIGURE 7
Comparison of the accuracy of the transfer learning process.

TABLE 10 Comparison of model accuracy.

Model	Accuracy (%)	Total time spent (h)	Model	Accuracy (%)	Total time spent (h)
B0	61.48	2.49	B2	61.84	2.73
B0-Freeze	43.67(-17.81)	**1.63**	B2-Freeze	43.58(-18.26)	1.67
B0-TF	**64.40(+2.92)**	2.50	B2-TF	64.02(+2.18)	2.73

Bold values mean the line with the best model evaluation index.

the one obtained through B2. Moreover, freezing the feature layer and then performing transfer learning resulted in a significant improvement of over 30% in the model's training speed. Direct transfer learning was performed on both groups of models, leading to accuracy improvements of over 2% compared to the original model.

Therefore, in practical applications, it is desirable for the Dise-Efficient model to make more precise judgments about the types of plant pests, thereby achieving accurate pest and disease prevention. Therefore, full-parameter migration holds great importance in enhancing the accuracy of the Dise-Efficient model in identifying plant pest types.

5 Conclusions

This present study introduces a novel Dise-Efficient model based on previous related research, capable of identifying various types of plant diseases and pests. A series of experiments were conducted to evaluate how the number of convolutional layers, learning strategy, and convolution kernel size affect the model's performance and how transfer learning can be applied to train the model. The following conclusions have been drawn from the experiments.

The Dise-Efficient-B0-N model achieved 99.71% accuracy in identifying plant disease types on the Plant Village plant disease dataset, with a model size of 13.3 MB. In addition, the model size decreases with fewer convolutional layers, leading to a slight reduction in accuracy. In contrast, more convolutional layers result in larger model size, but there is no obvious effect on accuracy improvements.

Also on the Plant Village plant disease dataset, implementing a cosine dynamic learning rate decay strategy during the training of the Dise-Efficient-B0-N model resulted in an accuracy rate of 99.80% in identifying plant disease types, higher than that of the B0-N model. The accuracy rate of the B0-L reached 99.81%, without any overfitting. Therefore, using a cosine dynamic learning rate decay strategy can effectively improve the accuracy of the model in identifying plant disease types.

The effect of convolution kernel size on the performance of the Dise-Efficient model on the IP102 plant pest dataset was investigated through experiments. Results indicate that the accuracy rates of the Dise-Efficient-B0 and Dise-Efficient-B2 models in identifying plant pest types on this dataset were 61.48%

and 61.84%, respectively, exceeding those of other advanced models in this field. Furthermore, the experimental results suggest that replacing small convolution kernels with larger ones in the Residual layer of the Dise-Efficient model is effective in improving the model's accuracy in identifying plant pest types.

The results obtained through the transfer learning experiment conducted on the IP102 plant pest dataset demonstrate that freezing the feature layer of the pre-trained model during transfer learning training increases the model training speed by more than 30%, which, however, comes at the cost of greatly reduced accuracy. Conversely, performing full-parameter transfer learning training on the pre-trained model keeps the model training speed unchanged while increasing the accuracy of the obtained model by more than 2%. These findings demonstrate the strong transfer learning ability of the Dise-Efficient model and suggest full-parameter transfer learning as an effective approach to improve the model's accuracy in identifying plant pest types.

In summary, our proposed Dise-Efficient model can effectively identify various types of plant diseases and pests, thereby contributing to preventing them in agricultural production. The baseline model Dise-Efficient-B0 exhibits the most comprehensive performance and boasts a compact size of only 13.3MB, making it ready for deployment in almost all kinds of lightweight mobile device applications. Specifically, the Dise-Efficient-B0 model achieves an accuracy rate of 99.80% for plant disease identification on the Plant Village dataset and an accuracy rate of 64.40% for plant pest type identification on the IP102 pest dataset after full-parameter transfer learning training. Consequently, it is anticipated that the Dise-Efficient-B0 model will be one of the top-performing models for plant disease and pest identification.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

HG: Paper writing and deep learning algorithm research. CF: Design of Dise-Efficient Plant Pest Type Recognition Model. GZ: Design of paper experiments. KL: Collation and analysis of the experimental data of the paper. PW and ZZ: Research and analysis of related work for the thesis. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ahmad, A., Saraswat, D., and El Gamal, A. (2022). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agric. Technol.* 3, 100083. doi: 10.1016/j.atech.2022.100083

Barragán-Fonseca, K. Y., Nurfikari, A., Van De Zande, E. M., Wantulla, M., Van Loon, J. J., De Boer, W., et al. (2022). Insect frass and exuviae to promote plant growth and health. *Trends Plant Sci.* 27 (7), 646–654.

Bedi, P., and Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artif. Intell. Agric.* 5, 90–101. doi: 10.1016/j.aiia.2021.05.002

Behmann, J., Mahlein, A. K., Rumpf, T., Römer, C., and Plümer, L. (2015). A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. *Precis. Agric.* 16, 239–260. doi: 10.1007/s11119-014-9372-7

Canassa, F., Esteca, F. C., Moral, R. A., Meyling, N. V., Klingen, I., and Delalibera, I. (2020). Root inoculation of strawberry with the entomopathogenic fungi metarhizium robertsii and beauveria bassiana reduces incidence of the twospotted spider mite and selected insect pests and plant diseases in the field. *J. Pest Sci.* 93 (1), 261–274. doi: 10.1007/s10340-019-01147-z

Chen, J., Zhang, D., Suzauddola, M., Nanehkaran, Y. A., and Sun, Y. (2021). Identification of plant disease images *via* a squeeze-and-excitation MobileNet model and twice transfer learning. *IET Image Process.* 15 (5), 1115–1127. doi: 10.1049/ipr2.12090

Elnahal, A. S., El-Saadony, M. T., Saad, A. M., Desoky, E. S. M., El-Tahan, A. M., Rady, M. M., et al. (2022). The use of microbial inoculants for biological control, plant growth promotion, and sustainable agriculture: a review. *Eur. J. Plant Pathol.* 162 (4), 759–792. doi: 10.1007/s10658-021-02393-7

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2019). A robust deep-Learning-Based detector for real-time tomato plant diseases and pests recognition. *Sensors Agric.* 1, 17, 153. doi: https://doi.org/10.3390/s17092022

Ganatra, N., and Patel, A. (2020). Performance analysis of fine-tuned convolutional neural network models for plant disease classification. *Int. J. Control Automation* 13 (3), 293–305.

Gokulnath, B. V. (2021). Identifying and classifying plant disease using resilient LF-CNN. *Ecol. Inf.* 63, 101283. doi: 10.1016/j.ecoinf.2021.101283

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition* 770–778.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. (2019). "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 558–567.

Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., et al. (2019). "Searching for mobilenetv3," in Proceedings of the IEEE/CVF international conference on computer vision. 1314-1324.

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132-7141.

Hughes, D., and Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*. doi: https://doi.org/10.48550/arXiv.1511.08060

Kamal, K. C., Yin, Z., Wu, M., and Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* 165, 104948. doi: 10.1016/j.compag.2019.104948

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60 (6), 84–90. doi: 10.1145/3065386

Kumar, V., Arora, H., and Sisodia, J. (2020). Resnet-based approach for detection and classification of plant leaf diseases. *In 2020 international conference on electronics and sustainable communication systems (ICESC)*. (IEEE), 495–502.

Lin, S., Xiu, Y., Kong, J., Yang, C., and Zhao, C. (2023). An effective pyramid neural network based on graph-related attentions structure for fine-grained disease and pest identification in intelligent agriculture. *Agriculture* 13 (3), 567. doi: 10.3390/agriculture13030567

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Liu, Y., Zhang, X., Gao, Y., Qu, T., and Shi, Y. (2022). Improved CNN method for crop pest identification based on transfer learning. *Comp. Intelligence Neurosci.* 2016.

Loshchilov, I., and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv* 05101. doi: https://doi.org/10.48550/arXiv.1711.05101

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1419. doi: 10.3389/fpls.2016.01419

Nanni, L., Maguolo, G., and Pancino, F. (2020). Insect pest image detection and recognition based on bio-inspired methods. *Ecol. Inf.* 57, 101089. doi: 10.1016/j.ecoinf.2020.101089

Nurfauzi, A. H., Azhar, Y., and Chandranegara, D. R. (2023). Penerapan model EfficientNetV2-B0 pada baseline IP102 dataset untuk menyelesaikan masalah klasifikasi hama serangga. *Jurnal Repositor* 5 (3), 805–814. doi: https://doi.org/10.22219/repositor.v5i3.1583

Ren, F., Liu, W., and Wu, G. (2019). Feature reuse residual networks for insect pest recognition. *IEEE Access* 7, 122758–122768. doi: 10.1109/ACCESS.2019.2938194

Satorras, V. G., and Estrach, J. B. (2018). "Few-shot learning with graph neural networks," in *International conference on learning representations*.

Sehrawat, A., Sindhu, S. S., and Glick, B. R. (2022). Hydrogen cyanide production by soil bacteria: biological control of pests and promotion of plant growth in sustainable agriculture. *Pedosphere* 32 (1), 15–38. doi: 10.1016/S1002-0160(21)60058-9

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comp. Intelligence Neurosci*. 2016.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

Tan, M., and Le, Q. (2019). "Efficientnet: rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. (PMLR), 6105–6114.

Tan, M., and Le, Q. (2021). "Efficientnetv2: smaller models and faster training,". in *International conference on machine learning*. (PMLR), 10096–10106.

Thenmozhi, K., and Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* 164, 104906. doi: 10.1016/j.compag.2019.104906

Wani, J. A., Sharma, S., Muzamil, M., Ahmed, S., Sharma, S., and Singh, S. (2022). Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: methodologies, applications, and challenges. *Arch. Comput. Methods Eng.* 29 (1), 641–677. doi: 10.1007/s11831-021-09588-5

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *J. Big Data* 3 (1), 1–40. doi: 10.1186/s40537-016-0043-6

Wu, X., Zhan, C., Lai, Y. K., Cheng, M. M., and Yang, J. (2019). "Ip102: a large-scale baseline dataset for insect pest recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8787–8796.

Zhu, Z., Zhang, D., Li, L., Li, K., Qi, J., Wang, W., et al. (2023). Knowledge-guided multi-granularity GCN for ABSA. *Inf. Process. Manage.* 60 (2), 103223. doi: 10.1016/j.ipm.2022.103223

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 21 July 2023
DOI 10.3389/fpls.2023.1184352

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Yunchao Tang,
Zhongkai University of Agriculture and Engineering, China
José Dias Pereira,
Instituto Politecnico de Setubal (IPS), Portugal

*CORRESPONDENCE
Jun Li
autojunli@scau.edu.cn

RECEIVED 11 March 2023
ACCEPTED 12 June 2023
PUBLISHED 21 July 2023

CITATION
Zhang W, Zeng Y, Wang S, Wang T, Li H, Fei K, Qiu X, Jiang R and Li J (2023) Research on the local path planning of an orchard mowing robot based on an elliptic repulsion scope boundary constraint potential field method. *Front. Plant Sci.* 14:1184352. doi: 10.3389/fpls.2023.1184352

Research on the local path planning of an orchard mowing robot based on an elliptic repulsion scope boundary constraint potential field method

Wenyu Zhang[1], Ye Zeng[1], Sifan Wang[1], Tao Wang[1], Haomin Li[1], Ke Fei[1], Xinrui Qiu[1], Runpeng Jiang[1] and Jun Li[1,2]*

[1]College of Engineering, South China Agricultural University, Guangzhou, China, [2]Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou, China

In orchard scenes, the complex terrain environment will affect the operational safety of mowing robots. For this reason, this paper proposes an improved local path planning algorithm for an artificial potential field, which introduces the scope of an elliptic repulsion potential field as the boundary potential field. The potential field function adopts an improved variable polynomial and adds a distance factor, which effectively solves the problems of unreachable targets and local minima. In addition, the scope of the repulsion potential field is changed to an ellipse, and a fruit tree boundary potential field is added, which effectively reduces the environmental potential field complexity, enables the robot to avoid obstacles in advance without crossing the fruit tree boundary, and improves the safety of the robot when working independently. The path length planned by the improved algorithm is 6.78% shorter than that of the traditional artificial potential method, The experimental results show that the path planned using the improved algorithm is shorter, smoother and has good obstacle avoidance ability.

KEYWORDS

mowing robot, artificial potential field, path planning, local minimum, boundary potential field

1 Introduction

With the development of science and technology, mobile robots are increasingly used in agriculture. In orchards, mowing robots with autonomous navigation ability are a hot research topic. As a key autonomous navigation technology, path planning has attracted increasing attention from researchers.

According to the degree of a mobile robot's mastery of the information in an area, path planning can be divided into two types: one is global path planning based on complete area

information (Li et al., 2022), and the other is local path planning based on local area information (Wu et al., 2022). Algorithms to solve global path planning include particle swarm optimization (PSO) (Delice et al., 2017; Wang et al., 2018), visibility methods (Zimmermann and König, 2016; Salman et al., 2023), and link graph methods (McCammon and Hollinger, 2021) and topology method (Jin and Choi, 2011). Algorithms to solve local path planning include artificial potential field methods (Khatib, 1986), the ant colony algorithm (Gao et al., 2023; Li et al., 2023), the A* algorithm (Zhang et al., 2022), artificial immune methods (Lin et al., 2023) and rolling window methods (Xin et al., 2023). Real-time mowing robot obstacle avoidance mainly utilizes local robot path planning algorithms. Because of the advantages of a simple structure, easy understanding, small calculation and real-time capability, artificial potential field methods are widely used in the robot field.

The basic idea of an artificial potential field (APF) method is constructing a virtual APF that senses the positions of the robot, obstacles and target points in an environment using sensors so that the mobile robot can be influenced by the target points and obstacles at the same time. In the potential field, the robot is attracted by the target points and moves toward them while being repelled by the obstacles and moves away from them. Therefore, under the action of this resultant force, the robot avoids obstacles and moves toward the target points, thus planning a collision-free path. Compared with other classical obstacle avoidance algorithms, an APF method has the advantages of fewer calculations, solving local obstacle avoidance problems and solving sudden challenges. Therefore, this algorithm is widely used in obstacle avoidance methods. However, an APF method has the following obvious disadvantages:

1. Target unreachable problem: When the robot is far away from a target point, the attraction will become extremely large. If the relatively small repulsion force can be ignored, the robot may encounter obstacles on its path. When there are obstacles near the target point, the repulsion force will be very large, and the attraction will be relatively small, making it difficult for the robot to reach the target point. When the distance between the robot and the target point is very close, if there is an obstacle near the target point, the attraction on the robot is approximately zero relative to the large repulsion, and the robot will always wander around the target point and cannot reach the target point.
2. Local minimum problem: The robot relies on the overlapping of the potential fields detected from all directions to obtain the overlapping field, and the direction and size of the overlapping field are used to determine the next trajectory. However, if the overlapping field is close to zero, the robot will not move and stop.
3. Poor adaptability in a complex environment: The more obstacles there are in the overlapping field, the higher the probability that the overlapping field is zero, and the easier it is to stagnate, leading to the local minimum problem.

In this regard, many scholars have invested much energy in research and improvement. Based on an artificial immune algorithm, Hou YB (Hou et al., 2012) adopted a potential field function method in an APF method to easily obtain the optimal path and improve the quality of path planning. Q. Song (Q. Song et al., 2012) To effectively solve the local minimum problem of APF methods, the force function of the potential field was improved using a velocity vector, and the repulsive potential field coefficient was adjusted in real time by combining it with a fuzzy control algorithm, which overcomes the robot easily falling into a local minimum and alleviates the oscillation problem. Li G (Li et al., 2013) proposed an improved APF method based on a regression search method, redefined the potential field function to solve the local minimum and oscillation problems, improved a wall-following method to solve the unreachable problem, and optimized the planned path using a regression search algorithm to obtain a better and shorter effective path. To solve the problems of local minimum and inefficiency of classical APF methods, Abdalla T Y (Abdalla et al., 2017) proposed a fuzzy control algorithm to improve the APF method, and the proposed problems were successfully solved. A fuzzy logic controller was used to control the movement of the robot, and a particle swarm optimization algorithm was used to optimize the membership function of the controller. Rostami S M H (Rostami et al., 2019) proposed an improved APF method to address the optimal path and solve the problems of local minima and unreachable targets in the APF algorithm, realizing effective robot obstacle avoidance without falling into local minima. Orozco-Rosas U (Orozco-Rosas et al., 2019) proposed a membrane evolution APF method for robot path planning, combining membrane calculation using a genetic algorithm and APF method to find suitable parameters, thus generating a feasible and safe path. This method consists of limited separated regions, in which there are several groups of parameters evolving according to biochemical inspiration to minimize the path length. Compared with classical APF methods, it shows better performance in path length. Jiachen Yang (Yang et al., 2022) proposes a Residual-like Soft Actor Critic (R-SAC) algorithm for agricultural scenarios, which improves the efficiency of reinforcement learning through offline experts experience pre-training methods, and optimizes the reward mechanism of the algorithm by using multi-step TD error, which solves the dilemma that may occur in the training process, and is a stable and efficient path planning method.

The author analyzes the above three problems in detail and proposes three improvement methods:

1. A target point distance factor is introduced into the attraction and repulsion potential field functions to reduce the resultant attraction and repulsion force received near a target point when the algorithm is far away;
2. An improved variable polynomial is used in the repulsion potential field function, which minimizes the distorted obstacle potential field when the robot is not near the target point and simultaneously ensures that the robot takes the global minimum at the target point;

3. The scope of the repulsion potential field is changed to an ellipse, and a fruit tree boundary potential field is added to reduce the environmentally potential field complexity so that the robot can avoid obstacles in advance without crossing the fruit tree boundary.

The effectiveness of the improved algorithm is verified through simulation and field tests.

2 Improved artificial potential field method with boundary constraints

2.1 Attractive potential field with distance factor introduced

The distance between the robot and a target point in a traditional APF method directly determines the attractive potential field function or the attractive force. When the distance between the robot and the target point is very large, the attractive potential field function or attraction will also become very large. In other words, the attraction plays a major role, while the repulsion plays a very small role in the robot motion control, which will easily lead to collisions between the robot and obstacles. To solve the collision risk of robots in an obstacle environment when considering the deviation of path planning, the attractive potential field function of the APF method is optimized, and a target point distance factor is added to reduce the attraction of the algorithm when the target point is far away. The improved attractive potential field function is defined as follows:

$$U_{att}(X) = \begin{cases} \frac{1}{2}k \cdot \rho^2(X, X_g), \rho(X, X_g) \le d = 2\rho_0 \\ \frac{1}{2}k \cdot d \cdot \rho(X, X_y), \rho(X, X_g) > d = 2\rho_0 \end{cases} \quad (1)$$

where k is the attractive gain coefficient, d is a constant determined by the environment, $X(x,y)$ is the current position of the robot, $\rho(X,X_g)$ is the distance between the robot and the target point, and ρ_0 is the influence radius of the obstacle.

The improved attractive function is shown in Formula (2):

$$F_{att}(X) = \begin{cases} -k \cdot \rho(X, X_g) \cdot \nabla(X, X_g), \rho(X, X_g) \le d = 2\rho_0 \\ -\frac{1}{2}k \cdot d \cdot \nabla(X, X_g), \rho(X, X_g) > d = 2\rho_0 \end{cases} \quad (2)$$

2.2 Improved elliptic repulsion potential field with variable polynomials

In the actual operation process, a mowing robot is limited by the orchard environment and its own performance, so the obstacle repulsion potential field influence range is different from that of a traditional APF method. Therefore, the repulsion potential field influence range is improved as follows: the longitudinal distance of the influence range is increased so that the mowing robot can correct its direction in advance and enter the obstacle avoidance mode; the lateral distance of the influence range is reduced to ensure that the mowing robot can avoid obstacles safely. After modification, the influence range becomes oval, as shown in Figure 1:

In this study, the major axis and minor axis of the influence range of the repulsive potential field are ρ_0 and $\rho_1 = \frac{\rho_0}{2}$.

By improving the repulsive potential field function, the local minimum and the oscillation around obstacles are solved. To address the problems in an APF, a method of adding a rotating force is adopted to improve the repulsion function (Gao et al., 2023) by applying a polynomial factor not less than zero to the repulsion potential field, which becomes zero when the robot reaches the target position. When the superimposed potential fields are all equal to zero at the target position, the robot position is the global minimum. This polynomial is the squares (Yang et al., 2016; Xin et al., 2022) of the distance from the robot to a target point. This form of the repulsive potential field greatly distorts the shape of the repulsive potential field when the robot is not near a target point while ensuring the global minimum of the target point. Therefore, in this study, an improved variable polynomial is used to minimize the distorted obstacle potential field when the robot is not near a

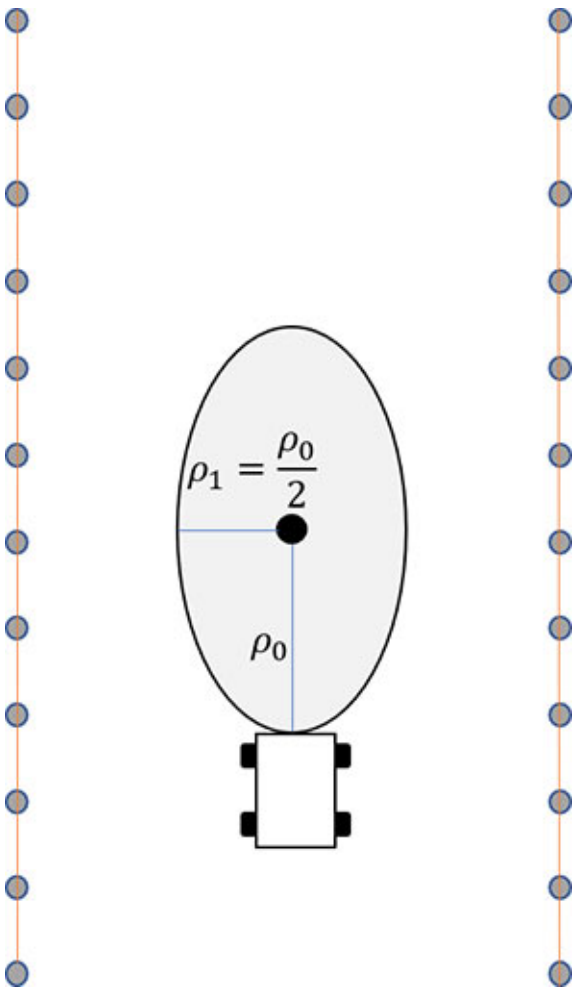


FIGURE 1
Influence range of the elliptic repulsion potential field.

10.3389/fpls.2023.1184352

target point and at the same time ensure that the robot has the global minimum at the target point. The improved repulsion potential field function is defined in Formula (3):

$$U_{rep}(x)=\begin{cases}\frac{1}{2}\eta\cdot\left[\frac{1}{\rho(X,X_0)}-\frac{1}{\rho_0}\right]\cdot\left[1-e^{\frac{\rho^2(X,X_g)}{R^2}}\right], X\epsilon\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1\\ 0, X\notin\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1\end{cases}\quad(3)$$

where η is the repulsion gain coefficient, ρ_0 is the major axis of the influence range of the obstacle, $\rho_1=\frac{\rho_0}{2}$ is the minor axis of the influence range of the obstacle, R is the radius of the robot, $X_0(x_0,y_0)$ is the position of an obstacle, $X_g(x_g,y_g)$ is the position of a target point, $\rho(X,X_0)$ is the Euclidean distance between the current position of the robot and the position of obstacle X_0 and $\rho(X,X_0)$ is the Euclidean distance between the robot and target point. When the robot moves to the target position, the total potential field $U_{total}(X)$ is equal to zero. Therefore, when the robot moves to the target position, the robot will stop moving at the target position when the speed drops to zero, so the total potential field of the robot at the target position is equal to zero.

The improved repulsion function is shown in Formula (4):

$$F_{rep}(X)=\begin{cases}F_{rep1}(X)+F_{rep2}(X), X\epsilon\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1\\ 0, X\notin\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1\end{cases}\quad(4)$$

where Formula (5) $F_{rep1}(X)$ means that the robot is far away from an obstacle along the line connecting it with the obstacle, and it decreases with the decrease in the distance between the robot and the target point; Formula (6) $F_{rep2}(X)$ means that the robot approaches the target position along the line connecting the robot and target position.

$$F_{rep1}(X)=\frac{1}{2}\eta\cdot\frac{1}{\rho^2(X,X_0)}\cdot\left[1-e^{\frac{\rho^2(X,X_g)}{R^2}}\right]\cdot\nabla(X,X_0), X\epsilon\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1\quad(5)$$

$$F_{rep2}(X)=\frac{1}{2}\eta\cdot\left[\frac{1}{\rho(X,X_0)}-\frac{1}{\rho_0}\right]\cdot\left[e^{\frac{\rho^2(X,X_g)}{R^2}}\cdot\frac{2\rho(X,X_g)}{R^2}\right]\cdot\nabla(X,X_g)\quad(6)$$
$$,X\epsilon\frac{(x-x_0)^2}{\rho_0^2}+\frac{(y-y_0)^2}{\rho_1^2}=1$$

2.3 Introduction of a fruit tree boundary potential field

When a mowing robot operates in an actual orchard, it needs to consider the influence of the surrounding environment while considering the obstacles. When the mowing robot moves to avoid obstacles, it cannot hit the fruit trees. In most orchards, facilities such as water and fertilizer irrigation and green prevention and control are installed among the fruit trees, as shown in Figure 2. If the fruit trees are regarded as individual obstacles, a large number of obstacles will easily make the mowing robot fall into the local minimum, and it is impossible to drive to the target point. At the same time, according to the operating characteristics of the mowing robot, it is easy to damage the facilities when driving into a fruit tree row. Therefore, adding a repulsive potential field to each fruit tree row as a boundary can effectively reduce the environmental potential field complexity and prevent orchard facilities from being damaged. According to mowing robot operating experience, the fruit tree boundary is the area with the highest risk factor, followed by the middle area of the path, as shown in Figure 3. According to the above distribution of the path danger degree, a path boundary potential field function is considered in sections. When the mowing robot is located in the area between paths, a function with a relatively gentle change trend is adopted; however, when it is close to the fruit tree boundary area, because of the high risk coefficient, a function with a large change trend is adopted. Based on

FIGURE 2
Facilities installed in fruit tree intervals.

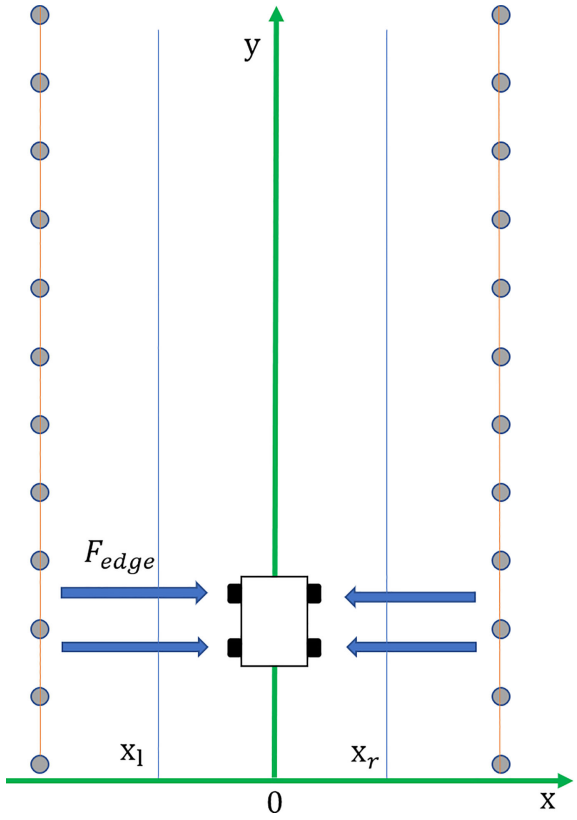


FIGURE 3
Boundary potential field.

the above factors, the orchard path is divided into four parts, and a fruit tree boundary potential field function is established as shown in Formula (7):

$$U_{edge}(X) = \begin{cases} \eta_{edge} \cdot v \cdot e^{x-x_l}, x \le x_l \\ \frac{1}{3}\eta_{edge} \cdot x^2, x_l < x < 0 \\ -\frac{1}{3}\eta_{edge} \cdot x^2, 0 \le x < x_r \\ \eta_{edge} \cdot v \cdot e^{x-x_r}, x_r \le x \end{cases} \quad (7)$$

where η_{edge} is the potential energy gain coefficient near the fruit tree boundary, $\frac{1}{3}\eta_{edge}$ is the potential energy gain coefficient of the middle part of the path, $X_l = -\frac{L}{4}$ is the dividing line near the left boundary, $X_r = \frac{L}{4}$ is the dividing line near the right boundary, and L is the path width.

In summary, the total potential field function of the improved APF method is:

$$U_{total}(X) = U_{att}(X) + \sum_{i=1}^{n} U_{rep}(X) + U_{edge}(X) \quad (8)$$

3 Algorithm test and result analysis

To verify the effectiveness of the improved APF method designed in this study. Written the improved algorithm, and validated the code through the MATLAB simulation platform, defining a path planning evaluation model. The simulation results of the improved algorithm are compared with those of the traditional artificial potential field method, and the practical test of the improved artificial potential field method is conducted on the self-developed mowing robot platform.

3.1 Path planning evaluation model

To directly evaluate the quality of different path planning methods, a path planning evaluation model is established in combination with a practical application, which mainly includes the following three key evaluation parameters:

3.11 Path planning evaluation

The primary goal and basic requirement of path planning is to generate a safe collision-free path. If the robot collides with an obstacle or cannot reach the target during the path planning, the path planning is invalid, which is a typical 0-1 problem.

$$f_{success} = \begin{cases} 1, success \\ 0, fail \end{cases} \quad (9)$$

3.12 The total length of the planned path

When the robot actually moves and runs, the total length of the planned path can be equated to the cost of energy and time

consumed by the robot. The shorter the total length of the planned path is, the better the path planning result. Assuming that the planned path is divided into N sections and the path length of time period i is S_i, the total length of the planned path is:

$$S_{total} = \sum_{i=1}^{N} S_i \tag{10}$$

3.13 Maximum turning angular velocity

In the real working environment space, the path planned by the robot is often a curve due to the existence of obstacles, so it is easy to know that the course of the robot changes in real time. The heading angular velocity of the robot is the first derivative of the heading angle with respect to time. The smaller the angular velocity of the robot is, the smoother the planned path, and the better the stability and maneuverability of the robot.

Let the heading of the i time period be θ_i and the heading of the i + 1 time period be θ_{i+1}; then, the heading angular velocity of the robot is:

$$\Delta\theta_i = \frac{\theta_{i+1} - \theta_i}{step_i}, (i = 1, 2, \ldots N-1) \tag{11}$$

where $step_i$ is the time consumed in planning period i. In the whole path planning cycle, the maximum absolute value of the heading difference may be taken as the maximum turning angular velocity, that is,

$$\omega_{\max} = \max\{\Delta\theta_1, \ldots, \Delta\theta_i, \ldots, \Delta\theta_{n-1}\}, \ (i = 1, 2, \ldots, N-1) \tag{12}$$

Based on the above three parameters, the path planning is evaluation model determined, and the evaluation function value is VF, as shown in Formula (13):

$$VF = f_{success} \cdot \left(\frac{r_1}{S_{total}} + \frac{r_2}{\omega_{\max}}\right) \tag{13}$$

where $r1$ and $r2$ are greater than zero and satisfy $r1 + r2 = 1$. By definition, the larger VF is, the higher the quality of the planned path.

3.2 Test steps and parameter settings

3.2.1 Simulation test steps and parameter settings

The lawn mower robot obstacle avoidance path planning of based on the improved APF method can be divided into the following steps:

S1, setting the positions of the starting point and the target point of the mowing robot, initializing the parameters, and establishing an environmental model around the robot using sensors mounted on the robot for environmental perception;

S2, calculating the attraction potential field function;

S3, calculating the repulsion potential field function;

S4, calculating the boundary potential field function;

S5, calculating the magnitude and direction of the attraction and repulsion exerted by the robot, calculating the components of the attraction and repulsion in the horizontal direction and the vertical direction, and determining the magnitude and direction of the total potential force exerted by the robot;

S6, setting the mowing robot moving step and updating the robot coordinates.

$$x(k+1) = x(k) + l\cos\theta \tag{14}$$

$$y(k+1) = y(k) + l\sin\theta \tag{15}$$

Guided by the total potential force of the APF method, the robot moves to a target point and the coordinates are updated. When the robot does not reach the target point, it continues to run under the combined force. When the mowing robot reaches the target point, it stops running. Thus, the planning path that meets the mowing robot operating requirements is obtained.

The attraction gain coefficient k =15, the repulsion gain coefficient η = 20 and the boundary repulsion gain coefficient η_{edge} = 35 in the improved APF method are set through continuous experimental tests, and the major axis $\rho_{0\ =}$ 2 m, minor axis $\rho_1 = \frac{1}{2}\rho_0$ = 1m, and step length l = 0.05 m. The traditional APF method has an attractive gain coefficient k = 15, a repulsive gain coefficient η = 20, an obstacle influence range $\rho_{0\ =}$ 2 m, and a step size l = 0.05 m. Considering the distance between the robot and the target, the repulsion potential field function is improved as a polynomial factor with an index m =1, and the other parameters are the same.

3.2.2 Real machine test steps and parameter settings

Using a self-developed mowing robot platform, a real machine verification test of the improved APF method is carried out, and a four-wheel electric differential structure is used. Equipped with 16-wire mechanical LIDAR, it can perceive the 360° environment around the lawn mower. The mowing robot use GPS to obtain global absolute position information and fuse IMU high-frequency body posture information to realize the navigation and positioning. The mowing robot measures the wheel speed through the rotary encoder to receive real-time feedback and control the vehicle speed, and obtain the actual trajectory value through the path tracking algorithm. The experimental environment is a modern standard orchard in the school, shown in Figure 4, with a spacing of 4 m and a length of 25 m. The real machine platform is shown in Figure 5.

According to the research objectives and content, the real-time obstacle avoidance experiment steps of the mowing robot are as follows:

1. A starting position (0, 0) and a target position (0,20) for the mowing robot are set according to an actual application scene;
2. To verify the applicability of the mowing robot, the scene is set according to the simulation test, obstacles are randomly placed between the starting position and the target position, and the obstacle position information is collected and recorded.

FIGURE 4
Orchard environment.

FIGURE 5
Real machine platform.

3. The mowing robot and all its instruments and equipment are started at the initial position, and the path planning algorithm based on the improved APF method is run to make the robot move autonomously in the obstacle scene set in step (2) and realize real-time obstacle avoidance;
4. A data acquisition program is run during the experiment and the GPS and IMU are used to collect the experimental data of the mowing robot during autonomous operation;
5. The real vehicle experimental data collected in step (4) are analyzed and compared with the planned path to verify the feasibility and effectiveness of the designed improved APF method.

In the Visual Studio Code software environment, the improved APF method is compiled into a Python program, uploaded to the vehicle controller, and the program is run in the set obstacle

environment for a real vehicle test. The algorithm parameters are set as follows: attractive gain coefficient k = 15, repulsive gain coefficient η =30, boundary repulsive gain coefficient η_{edge} = 40, obstacle influence range major axis ρ_0 = 3 m, minor axis $\rho_1 = \frac{1}{2}\rho_0$ = 1.5 m, and step length l = 0.1 m.

3.3 Simulation test results and analysis

Scenario 1:

In general, the obstacle environment is set as follows: there are n obstacles, with n=6, and the obstacle positions are X_0=[3 0.2; 7 -0.4; 10 0.3; 13 -0.2; 15 0.5; 17 -0.4], the starting position of the robot is X_s=[0 0.1], and the target position is X_g=[20 0.1].

According to the established path planning evaluation model, the path quality planned using the different model algorithms under different scenarios is evaluated. The evaluation data are shown in Table 1:

The scenario 1 simulation results are shown in Figures 6, 7, and the experimental results show that both the improved APF method and the APF method can realize collision-free effective path planning. Among them, there is slight oscillation in the planned path in Figure 6. There is no oscillation or jitter in the planned path shown in Figure 7. From Table 1, by comparing the parameters S_{total} and ω_{max}, it is found that the path length planned in Case 2 is the shortest, ω_{max} is greatly reduced, and the evaluation function VF value is the largest, so the path planned in Case 2 is shorter, smoother and better in quality than that planned using the traditional APF method.

Scenario 2:

In the setting of an obstacle environment with a local minimum, there are n obstacles, n=6, and the obstacle positions are X_0=[3 0.2; 7 -0.4; 13 -0.2; 15 0.5; 17 -0.4; 17 0.5], the starting position of the robot is X_s=[0 0.1], and the target position is X_g=[20 0.1].

According to the established path planning evaluation model, the path quality planned using the different model algorithms under scenario 2 is evaluated, and the evaluation data are shown in Table 2.

The scenario 2 simulation results are shown in Figures 8, 9, and the experimental results show that the traditional APF method is ineffective in path planning. Among them, there is a local minimum problem in the planned path in Figure 8, and the robot cannot continue to move to the target position when it falls into a local minimum. It can be seen from Figure 9 that the Case 2 method can

TABLE 1 Scenario 1 path planning data quality evaluation under the same environment.

Case	*fcollision*	S_{total}(m)	ω_{max} (°/s)	*VF*
Case 1	1	21.72	72.8	0.0428
Case 2	1	20.34	15.9	0.0505

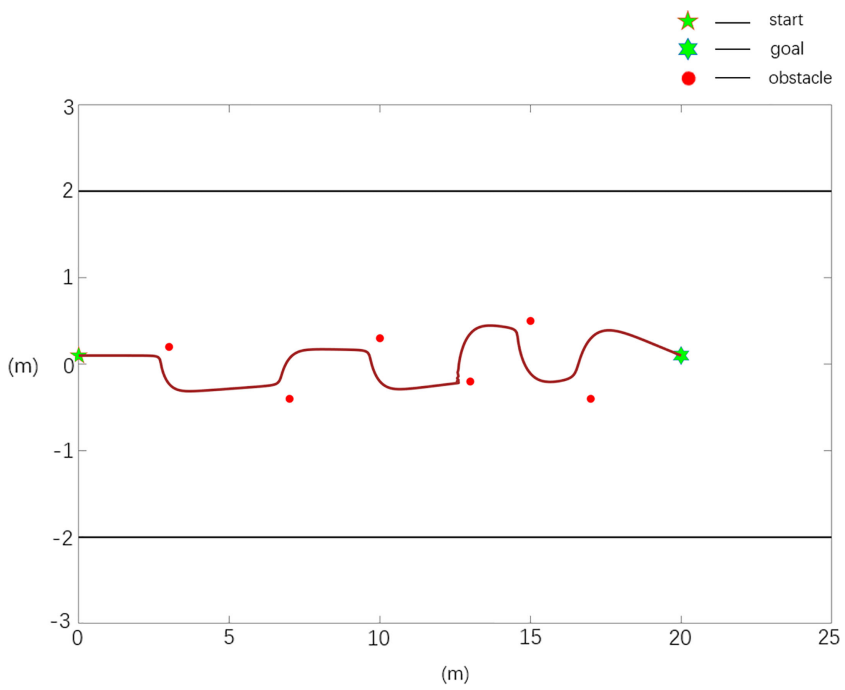


FIGURE 6
Scenario 1, case 1 test result using the traditional APF method.

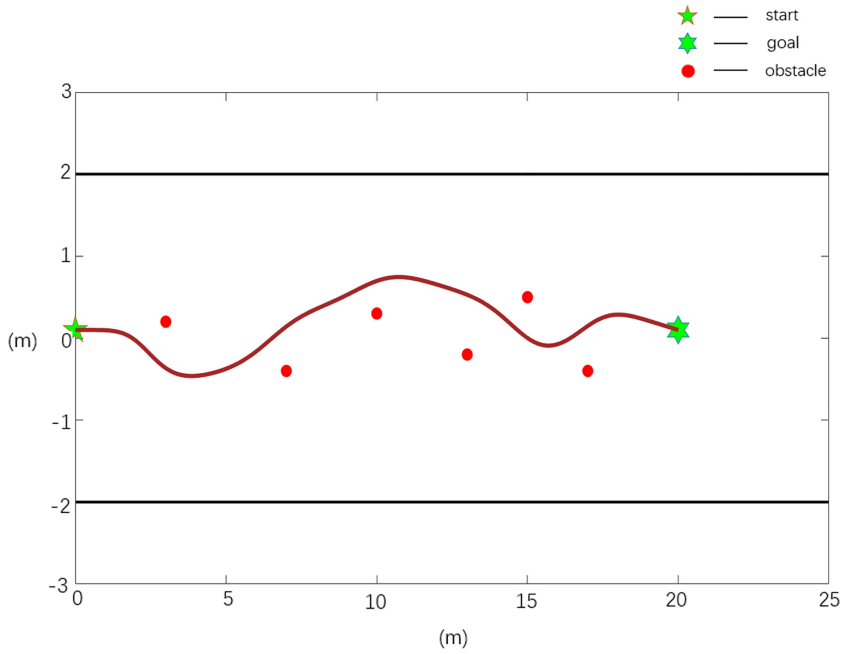


FIGURE 7
Scenario 1, case 2 test result 1 using the improved APF method.

TABLE 2 Scenario 2 path planning data quality evaluation under the same environment.

Case	*fcollision*	$S_{total}(m)$	ω_{max} (°/s)	*VF*
Case 1	0	16.92	68.4	0
Case 2	1	20.24	10.1	0.0543

realize effective path planning without collision, and in Table 2, compared with the parameter ω_{max}, the path planned in Case 2 is smoother and has better quality.

Scenario 3:

In the obstacle environment in the case of boundary collision, there are n obstacles, n=6, and the obstacle positions are X_0=[3 0.2; 5 -0.2; 7 -0.6; 9,-0.9; 10.5 -1.2; 12 -1.5], the starting position of the robot is X_s=[0 0], and the target position is X_g=[20 0].

According to the established path planning evaluation model, the path quality planned using the different model algorithms under different scenarios is evaluated, and the evaluation data are shown in Table 3.

The scenario 3 simulation results are shown in Figures 10, 11, and the experimental results show that the traditional APF method is ineffective in path planning. Among them, the path planned in Figure 10 has a boundary collision problem, and the robot collides with the fruit tree boundary during obstacle avoidance, resulting in obstacle avoidance failure. As seen from Figure 11, the Case 2 method can realize effective path planning without collision and will not collide with the fruit tree boundary. Based on the experimental results and analysis of scenario 3, the designed Case 2 method can not only effectively realize collision-free path planning, overcome the oscillation or jitter phenomenon in the path planning process, and effectively solve the problem that the robot easily falls into a local minimum but also avoid the boundary collision problem in the obstacle avoidance process and has the best comprehensive performance.

3.4 Real machine test results and analysis

Scenario 1:

In general, in the obstacle environment, the actual layout of obstacle position information is X_0=[4.0 0.6; 8.0 -0.8; 12.0 0.5; 16 -1.0]. The experimental results are shown in Figure 12. Among them, the dark blue point in the figure is the starting point of the mowing robot, the green point is the target point, the red points are obstacles, the black straight line is the orchard boundary, the blue curve represents the reference path planned based on the improved APF method, and the purple dotted line represents the experimental results of obstacle avoidance for the real vehicle.

The experimental results of scenario 1 obstacle avoidance verification are analyzed, and the analysis data are shown in Table 4.

Scenario 2:

When there is a local minimum, the actual obstacle position information is X_0=[4.0 0.2 8.0 -0.8; 16.0 0.7; 16 -0.6]. The experimental results are shown in Figure 13.

The experimental results of obstacle avoidance verification in scenario 2 are analyzed, and the analysis data are shown in Table 5.

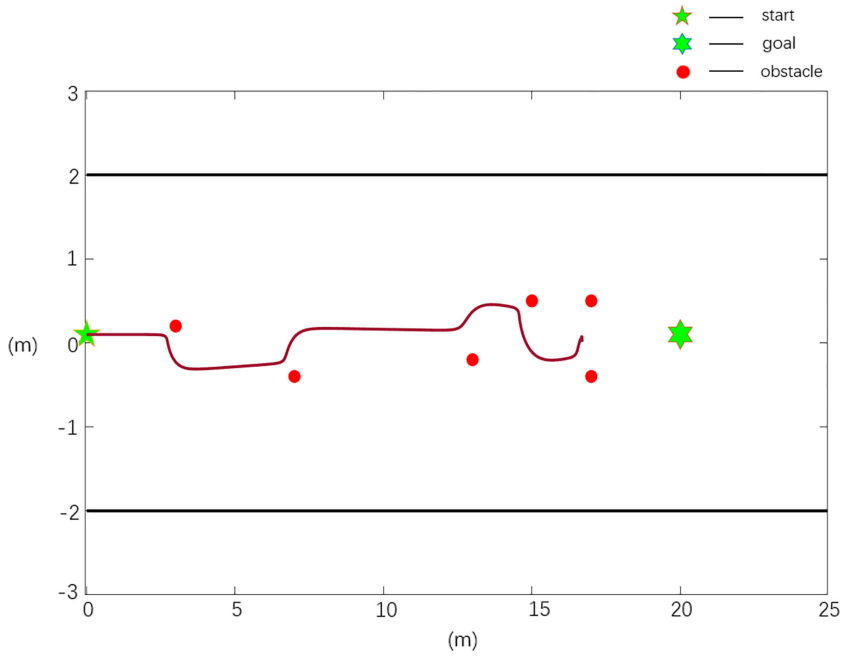


FIGURE 8
Scenario 2, case 1 test result using the traditional APF method.

Scenario 3:

In general, in the obstacle environment, the actual layout of obstacle position information is X_0=[4.0 -0.2; 6.5 0.5; 9.0 1.0; 11.5 -0.6]. The experimental results are shown in Figure 14.

The experimental results of obstacle avoidance verification in scenario 3 are analyzed, and the analysis data are shown in Table 6.

The test results of scenario 1 are shown in Figure 12 and Table 4. Compared with the planned path, the actual path has a length of 2.59% and a maximum rotation angle of 22.2%, with a maximum deviation of 0.137 m in the X direction and 0.051 m in the Y direction. As shown in Figure 13 and Table 5, the test results of scenario 2 show that the actual path is 2.3% longer than the

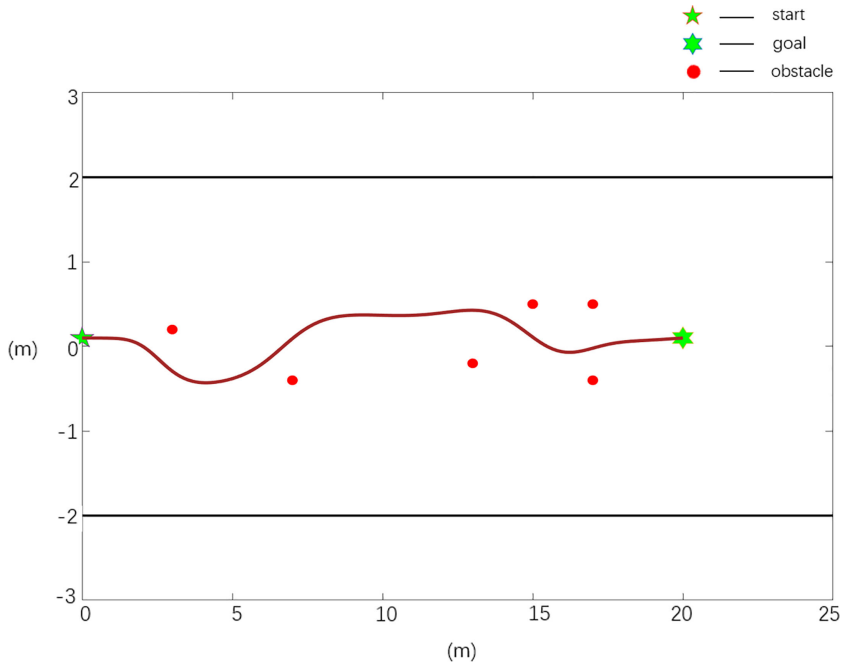


FIGURE 9
Scenario 2, case 2 test result using the improved APF method.

TABLE 3 Scenario 3 path planning data quality evaluation under the same environment.

Case	*fcollision*	$S_{total}(m)$	ω_{max} (°/s)	*VF*
Case 1	0	13.69	51.95	0
Case 2	1	20.17	6.97	0.0589

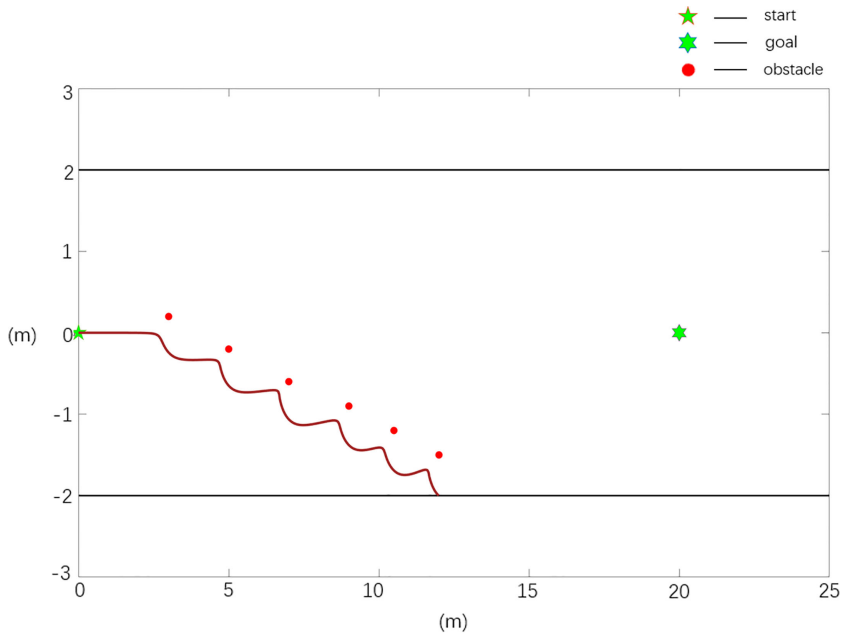


FIGURE 10
Scenario 3, case 1 test result using the traditional APF method.

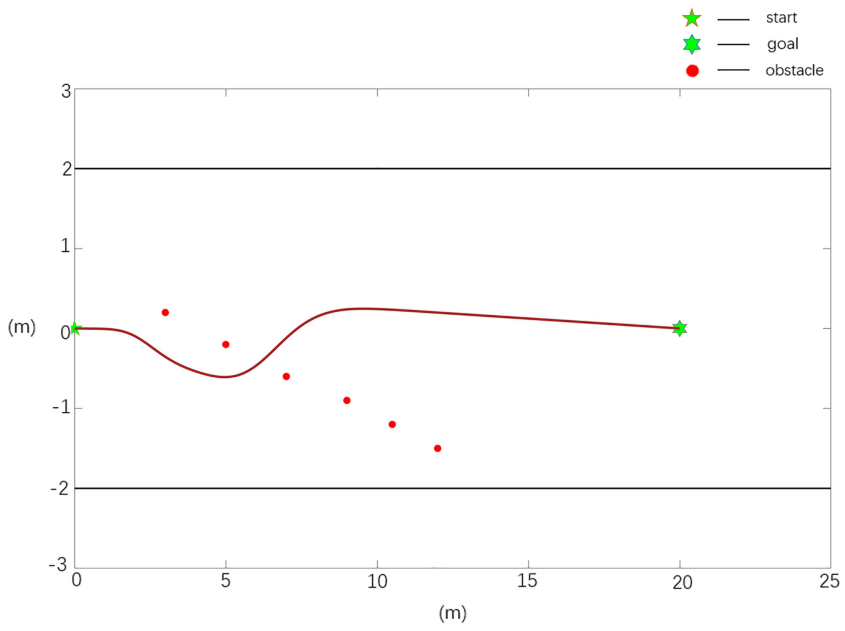


FIGURE 11
Scenario 3, case 2 test result using the improved APF method.

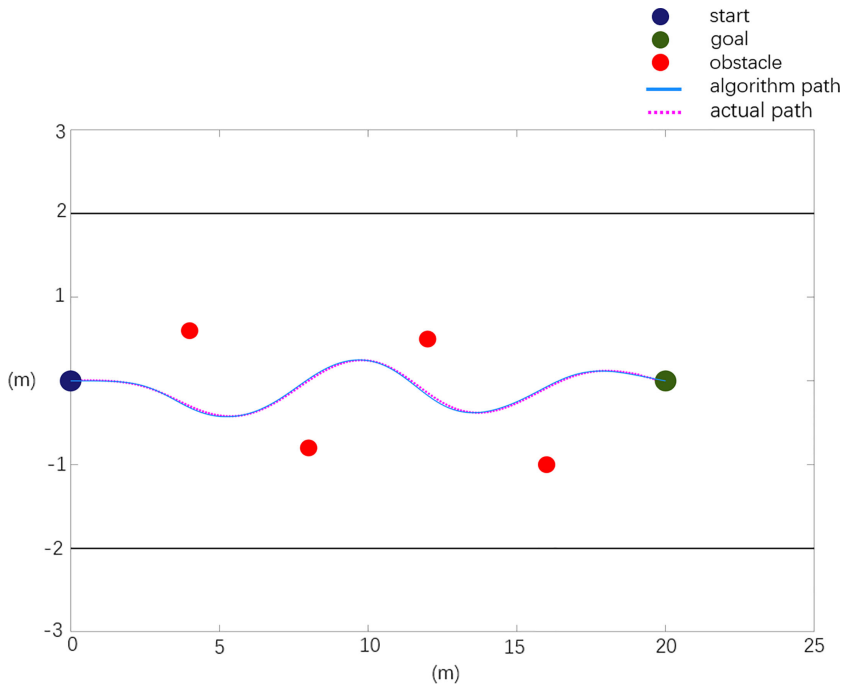


FIGURE 12
Real machine obstacle avoidance verification scenario 1.

planned path, and the maximum rotation angle is 12% smaller, of which the maximum deviation in the X direction is 0.105 m and the maximum deviation in the Y direction is 0.048 m. The test results of scenario 3 are shown in Figure 14 and Table 6. Compared with the planned path, the actual path length is 2.7% longer and the maximum rotation angle is 7.1% smaller, of which the maximum deviation in the X direction is 0.126 m and the maximum deviation in the Y direction is 0.053 m. The above situation shows that the gap between the actual path and the planned path is small, and the maximum displacement error is kept within 0.15 m, which meets the design needs.

The experimental results show that in an actual orchard environment, the mowing robot can effectively solve the local minimum problem and effectively avoid obstacles in the obstacle environment. The robot successfully completes the path planning from the initial position and avoids all obstacles to reach the target position safely. In the actual driving process, due to the influence of the orchard ground environment, the actual driving path deviates from the planned path, but it meets the control requirements of the mowing robot within the allowable control error.

4 Conclusion

The artificial potential field method has been widely used in local path planning because of its simple and real-time characteristics. To further improve the performance of the algorithm, many scholars have studied improving the method algorithms. In this study, the following methods are adopted: by improving the attractive field model, the problem of colliding with obstacles when the distance is too far and the attraction is too large is avoided; on the basis of the original repulsive force field, considering the influence of the relative position and speed between the target and the robot, a new repulsive function is introduced, and the repulsive potential field strength of obstacles near the target is reduced by adding a rotating force, thus solving the local minimum problem.

TABLE 4 Real machine obstacle avoidance verification experimental data analysis.

Path	Planning path	Actual path
Path length (m)	20.20	20.725
Maximum rotation angle (degree)	5.705	6.972
Maximum relative deviation in x direction (m)	0	0.137
Maximum relative deviation in y direction (m)	0	0.051

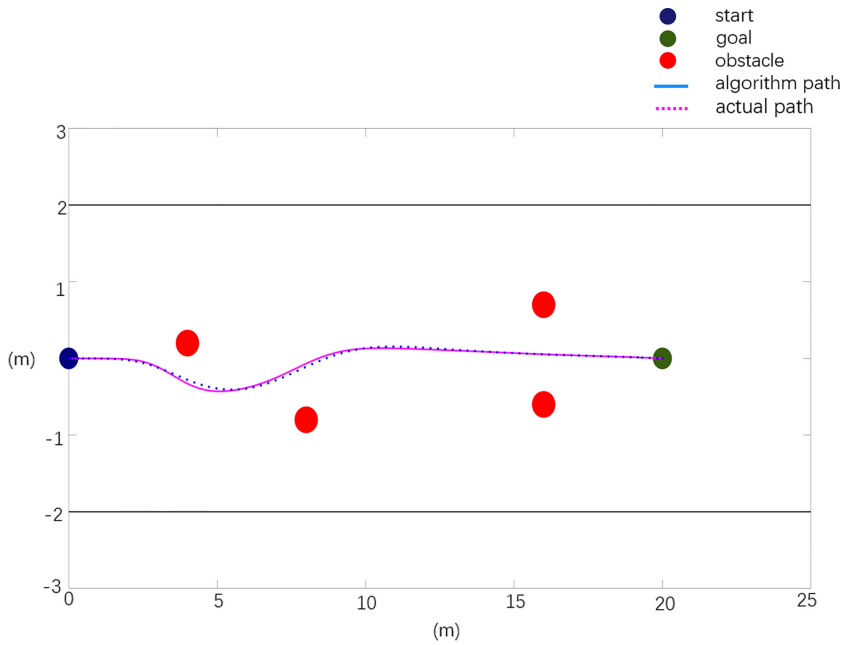


FIGURE 13
Real machine obstacle avoidance scenario 2 verification.

TABLE 5 Real machine obstacle avoidance verification experimental data analysis.

Path	Planning path	Actual path
Path length (m)	20.12	20.592
Maximum rotation angle (degree)	5.63	4.975
Maximum relative deviation in x direction (m)	0	0.105
Maximum relative deviation in y direction (m)	0	0.048

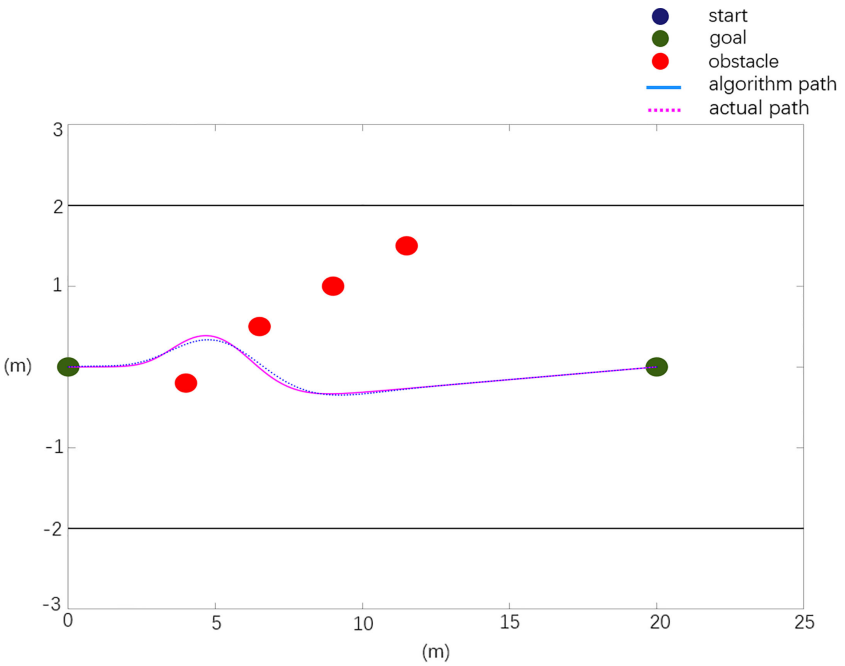


FIGURE 14
Real machine obstacle avoidance verification scenario 3.

TABLE 6 Real machine obstacle avoidance verification experimental data analysis.

Path	Planning path	Actual path
Path length (m)	20.17	20.730
Maximum rotation angle (degree)	9.23	8.572
Maximum relative deviation in x direction (m)	0	0.126
Maximum relative deviation in y direction (m)	0	0.053

The actual operation requirements of a mowing robot require path planning in complex environments. This study combines the advantages of these two methods, considers the environmental constraints in an actual orchard, modifies the scope of the repulsive potential field, and introduces boundary potential field constraints to ensure that the algorithm can realize planning path that meets the actual operation requirements of mowing robots.

To address the shortcomings of traditional APF path planning algorithms, an improved APF path planning algorithm suitable for orchard mowing robots is proposed. The simulation experiment in this study can be divided into three parts: first, the robustness of the improved APF method compared with a traditional APF method is verified, and the planning path is smoother and shorter. Second, it is verified that the improved algorithm has a stronger ability to solve local minimum problems. Finally, an actual orchard working environment is simulated, and it is verified that the improved APF method has better adaptability to the orchard environment and can successfully avoid boundary collisions and complete obstacle avoidance to reach the target point. At the same time, according to the scenario set up in the simulation experiment, the corresponding practical verification experiment of the improved APF method is carried out. The experimental results verify the effectiveness and reliability of the improved algorithm. This provides a new method for the path planning of this kind of mowing robot working in orchard environments.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported by the Guangdong Laboratory for Lingnan Modern Agriculture under Grant NZ2021040 NZ2021009, the China Agriculture Research System under Grant CARS-32, the Special Project of Rural Vitalization Strategy of Guangdong Academy of Agricultural Sciences under Grant TS-1-4, and the Guangdong Provincial Modern Agricultural Industry Technology System under Grant 2021KJ123.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdalla, T. Y., Abed, A. A., and Ahmed, A. A. (2017). Mobile robot navigation using PSO-optimized fuzzy artificial potential field with fuzzy control. *J. Intell. Fuzzy Syst.* 32 (6), 3893–3908. doi: 10.3233/IFS-162205

Delice, Y., Kızılkaya Aydoğan, E., Özcan, U., and İlkay, M. S. (2017). A modified particle swarm optimization algorithm to mixed-model two-sided asembly line balancing. *Mechanical Syst. Signal Process.* 28 (01), 23–26. doi: 10.1007/s10845-014-0959-7

Gao, Y., Bai, C., Fu, R., and Quan, Q. (2023). A non-potential orthogonal vector field method for more efficient robot navigation and control. *Rob. Auton. Syst.* 159. doi: 10.1016/j.robot.2022.104291

Gao, P., Zhou, L., Zhao, X., and Shao, B. (2023). Research on ship collision avoidance path planning based on modified potential field ant colony algorithm. *Ocean Coast. Manage.*, 235. doi: 10.1016/j.ocecoaman.2023.106482

Hou, Y. B., Wang, W., and Lu, X. Y. (2012). Mobile robot path planning and research in the improved artificial immune algorithm. *Adv. Mater. Res.*, 466–467. doi: 10.4028/www.scientific.net/AMR.524-527.466

Jin, S., and Choi, B. (2011). *Fuzzy logic system based obstacle avoidance for a mobile robot* (Berlin, Heidelberg: Springer Berlin Heidelberg), 1–6.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Res.* 5 (1), 90–98. doi: 10.1177/027836498600500106

Li, G., Tamura, Y., Yamashita, A., and Asama, H. (2013). Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning. *Int. J. Mechatron. Autom.* 03 (03), 141–170. doi: 10.1504/IJMA.2013.055612

Li, W., Wang, C., Huang, Y., and Cheung, Y. (2023). Heuristic smoothing ant colony optimization with differential information for the traveling salesman problem. *Appl. Soft Computing J.* 133. doi: 10.1016/j.asoc.2022.109943

Li, H., Zhao, T., and Dian, S. (2022). Forward search optimization and subgoal-based hybrid path planning to shorten and smooth global path for mobile robots. *Knowledge-Based Syst.* 258. doi: 10.1016/j.knosys.2022.110034

Lin, S., Liu, A., Wang, J., and Kong, X. (2023). An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. *J. Comput. Sci.*, 67. doi: 10.1016/j.jocs.2022.101938

McCammon, S., and Hollinger, G. A. (2021). Topological path planning for autonomous information gathering. *Auton. Robots* 45 (6). doi: 10.1007/s10514-021-10012-x

Orozco-Rosas, U., Montiel, O., and Sepúlveda, R. (2019). Mobile robot path planning using membrane evolutionary artificial potential field. *Appl. Soft Computing* 77, 236–251. doi: 10.1016/j.asoc.2019.01.036

Rostami, S. M. H., Sangaiah, A. K., Wang, J., and Liu, X. (2019). Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *Eurasip J. Wirel. Commun. Netw.* 70, 1–19. doi: 10.1186/s13638-019-1396-2

Salman, M., Khan, H., and Lee, M. C. (2023). Perturbation observer-based obstacle detection and its avoidance using artificial potential field in the unstructured environment. *Appl. Sci.* 13 (2). doi: 10.3390/app13020943

Song, Q., XL, L., and Wen-Xing, Z.. (2012). Mobile robot path planning based on dynamic fuzzy artificial potential field method. *Int. J. Hybrid Inf. Technol.* 5 (4), 85–93.

Wang, B., Li, S., Guo, J., and Chen, Q. (2018). Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. *Neurocoputing*, 42–51. doi: 10.1016/j.neucom.2017.12.015

Wu, B., Chi, X., Zhao, C., Zhang, W., Lu, Y., and Jiang, D. (2022). Dynamic path planning for forklift AGV based on smoothing a* and improved DWA hybrid algorithm. *Sensors* 22 (18). doi: 10.3390/s22187079

Xin, L., Dan, W., Di, W., Jia, H., Li,, and Hang, Y. (2023). Enhanced DWA algorithm for local path planning of mobile robot. *Ind. Robot* 50 (1). doi: 10.1108/IR-05-2022-0130

Xin, Z., Rongwu, X., and Guo, C. (2022). AUV path planning in dynamic environment based on improved artificial potential field method based on visibility graph. *J. Phys. Conf. Ser.* 2383, 1:012090. doi: 10.1088/1742-6596/2383/1/012090

Yang, J., Ni, J., Li, Y., Wen, J., and Chen, D. (2022). The intelligent path planning system of agricultural robot via reinforcement learning. *Sensors* 22 (12), 4316. doi: 10.3390/s22124316

Yang, X., Yang, W., Zhang, H., and Chang, H. (2016). A new method for robot path planning based artificial potential field[C]. *IEEE*, 1294–1299. doi: 10.1109/ICIEA.2016.7603784

Zhang, H.-E., Xu, G.-Q., Chen, H., and Li, M. (2022). Stability of a variable coefficient star-shaped network with distributed delay. *J. Syst. Sci. Complexity* 35 (06), 2077–2106. doi: 10.1007/s11424-022-1157-x

Zimmermann, M., and König, C. (2016). Integration of a visibility graph based path planning method in the ACT/FHS rotorcraft. *CEAS Aeronautical J.* 7 (3). doi: 10.1007/s13272-016-0197-0

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 28 July 2023
DOI 10.3389/fpls.2023.1211830

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Muhammad Azam,
University of Agriculture, Pakistan
Kusumiyati Kusumiyati,
Padjadjaran University, Indonesia

*CORRESPONDENCE
Xin Wang
h09036@cau.edu.cn

RECEIVED 25 April 2023
ACCEPTED 29 June 2023
PUBLISHED 28 July 2023

CITATION
Tang C, Chen D, Wang X, Ni X, Liu Y, Liu Y, Mao X and Wang S (2023) A fine recognition method of strawberry ripeness combining Mask R-CNN and region segmentation. *Front. Plant Sci.* 14:1211830. doi: 10.3389/fpls.2023.1211830

A fine recognition method of strawberry ripeness combining Mask R-CNN and region segmentation

Can Tang[1], Du Chen[1,2], Xin Wang[1]*, Xindong Ni[1], Yehong Liu[1], Yihao Liu[1], Xu Mao[1,3] and Shumao Wang[1,3]

[1]College of Engineering, China Agricultural University, Beijing, China, [2]State Key Laboratory of Intelligent Agricultural Power Equipment, Henan, China, [3]Beijing Key Laboratory of Optimized Design for Modern Agricultural Equipment, Beijing, China

As a fruit with high economic value, strawberry has a short ripeness period, and harvesting at an incorrect time will seriously affect the quality of strawberries, thereby reducing economic benefits. Therefore, the timing of its harvesting is very demanding. A fine ripeness recognition can provide more accurate crop information, and guide strawberry harvest management more timely and effectively. This study proposes a fine recognition method for field strawberry ripeness that combines deep learning and image processing. The method is divided into three stages: In the first stage, self-calibrated convolutions are added to the Mask R-CNN backbone network to improve the model performance, and then the model is used to extract the strawberry target in the image. In the second stage, the strawberry target is divided into four sub-regions by region segmentation method, and the color feature values of B, G, L, a and S channels are extracted for each sub-region. In the third stage, the strawberry ripeness is classified according to the color feature values and the results are visualized. Experimental results show that with the incorporation of self-calibrated convolutions into the Mask R-CNN, the model's performance has been substantially enhanced, leading to increased robustness against diverse occlusion interferences. As a result, the final average precision (AP) has improved to 0.937, representing a significant increase of 0.039 compared to the previous version. The strawberry ripeness classification effect is the best on the SVM classifier, and the accuracy under the combined channel BGLaS reaches 0.866. The classification results are better than common manual feature extraction methods and AlexNet, ResNet18 models. In order to clarify the role of the region segmentation method, the contribution of different sub-regions to each ripeness is also explored. The comprehensive results demonstrate that the proposed method enables the evaluation of six distinct ripeness levels of strawberries in the complex field environment. This method can provide accurate decision support for strawberry refined planting management.

KEYWORDS

strawberry, ripeness recognition, deep learning, image processing, Mask R-CNN

1 Introduction

Strawberries, being a typical non-climacteric fruit, can continue to ripen after being picked, but their edible quality does not improve with further ripening (Chen et al., 2014; Van de Poel et al., 2014). Once strawberries begin to bear fruit, they typically take 20-30 days to reach full ripeness. Furthermore, the transition from the white ripe stage to the fully ripe stage takes only about 7 days for strawberries. Therefore, an efficient and accurate method for assessing strawberry ripeness would align with practical requirements. The traditional manual observation method is characterized by low work efficiency, poor accuracy and significant variability, rendering it inadequate to meet the demands of efficient detection. Despite the high accuracy of the sensor detection method, its requirement for professional operation and low efficiency make it unsuitable for large-scale detection (Moghimi et al., 2010; Abbaszadeh et al., 2014; Aghilinategh et al., 2020). Therefore, it is of great significance to study an efficient and accurate strawberry ripeness judgment method in an unstructured environment for strawberry harvest management. However, the field environment where strawberries grow is characterized by leaf occlusion and fruit overlapping, presenting challenges in accurately recognizing the ripeness of strawberries.

With the advancement of new information technology and the promotion of technical methods, machine learning (ML) and deep learning (DL) have made significant strides in scene recognition and object classification. Considering their characteristics of faster detection, better generalization, and stronger robustness, these methods have also emerged as a research hotspot in strawberry detection and recognition (Yu et al., 2019; Pérez-Borrero et al., 2020; Le Louëdec and Cielniak, 2021). The current strawberry ripeness detection method predominantly revolve around the integration of ML, DL, and hyperspectral imaging techniques. Zhang et al. (2016) used PCA to obtain optimal wavelengths from hyperspectral images, and then extracted texture features from the optimal wavelength images. They finally obtained the best strawberry ripeness classification in SVM with the combined information of the best wavelength and texture features. Shao et al. (2020) extracted effective wavelengths for field and outdoor hyperspectral strawberry images, respectively. Finally, their PLS-DA and LS-SVM classifiers achieved between 91.7% and 96.7% accuracy in field strawberry ripeness classification. Su et al. (2021) established a 1D residual network and a 3D residual network to process 1D and 3D strawberry hyperspectral data. The accuracy of ripeness classification exceeded 84% in both networks. Raj et al. (2022) obtained over 98% ripeness classification accuracy when using the full spectrum data of strawberries as the input data of SVM. Furthermore, they developed a strawberry water content index based on a portion of the spectral data from the band, achieving the highest accuracy of 71.2% when using the water content index as input data. Additionally, there have been studies exploring the utilization of image processing techniques in conjunction with deep learning for strawberry ripeness detection. Fan et al. (2022) used a dark channel enhancement algorithm to preprocess strawberry images taken at night, and finally achieved a ripeness recognition accuracy of over 90% on YOLOv5. Despite achieving some results in strawberry ripeness estimation, hyperspectral imaging is known for its high cost and inconvenience in practical usage. Moreover, its application is primarily limited to indoor environments, making it challenging to fulfill the requirements of real-time detection in the field.

According to the characteristics of strawberry at different ripeness stages, most of the above studies have categorized strawberry ripeness into 2-3 levels. However, the classification of 2-3 levels is rough and cannot provide an accurate decision-making basis for strawberry harvesting management. On the one hand, foliar fertilizer spraying before strawberry ripening can increase the firmness of strawberries at harvest and prolong the storage time (He et al., 2018). This necessitates the identification of early ripeness in strawberries to determine optimal timing for fertilization. On the other hand, for the two different modes of on-site sales and off-site sales, it is necessary to identify the harvest ripeness of strawberries in the later stage to determine the harvest time. Therefore, considering the current large-scale strawberry cultivation, there is a need for finer ripeness grading to offer precise decision support for strawberry harvesting management.

Based on the above analysis, combined with deep learning technology and image processing technology, this paper proposes a strawberry ripeness recognition method combined with Mask R-CNN and region segmentation. This method not only enhances the segmentation accuracy of strawberries in complex field environments but also accurately estimates six distinct levels of ripeness, providing richer and more detailed information about strawberry maturation.

2 Materials and methods

2.1 Dataset

2.1.1 Image acquisition

In order to improve the robust performance of the model in various environments, the strawberry images for this study were acquired in two batches to increase data diversity. The first shot was taken on January 7, 2022 in a strawberry plantation in Changping District, Beijing, China, from 10:00 to 14:00, and the local weather was sunny and cloudless. The device used is an MI 8 smart mobile phone with a SONY IMX363 lens. The second shot was taken on February 9, 2023 in a strawberry plantation in Pinggu District, Beijing, China, from 13:00 to 17:00, and the local weather was cloudy. The device used is a MI 12X smart mobile phone, and the lens is SONY IMX766. The distance from the lens to the strawberry ridge was 0.2-0.3 m for each shooting, and finally 500 pictures with a size of 4032×2268 pixels and 700 pictures with a size of 4096×2304 pixels were obtained respectively. The pictures include images under different lighting conditions such as normal, frontlighting, and backlighting, as shown in Figure 1A. We compressed all images to a size of 1280×720 pixels to reduce computational cost.

2.1.2 Dataset partitioning and annotation

The strawberry datasets were divided into two parts: instance segmentation dataset and image classification dataset. For the instance segmentation dataset, the initial images were randomly

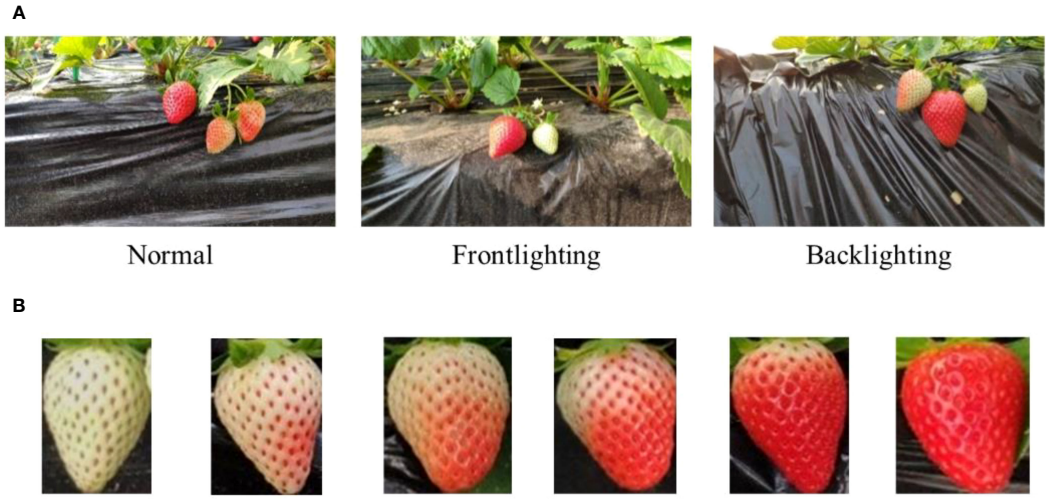


FIGURE 1
(A) Initial images. **(B)** Images of strawberries at different ripeness. From left to right: White, Breaking, Turning-1, Turning-2, Ripe and Full ripe.

divided into 860 images for training set, 100 images for validation set, and 240 images for test set. Each strawberry contour was annotated with labelme annotation tool. For the image classification dataset, the dataset consisted of a series of strawberry patches. The training set comprised a total of 2172 strawberry patches, which were manually cropped from the training set of the instance segmentation task. The test set consisted of a total of 651 strawberry patches, which were detected by the instance segmentation model from the test set of the instance segmentation task.

Efficient and accurate decision-making is crucial for the management of large-scale strawberry harvesting in order to enhance economic benefits. This necessitates a more precise classification of strawberry ripeness to meet the requirements of the industry. Based on the physiological changes (Azodanlou et al., 2004; Zhang et al., 2011) and color representation of strawberries during the ripening process, the strawberry ripeness has been categorized into six levels: White, Breaking, Turning-1, Turning-2, Ripe and Full ripe. At White the fruit is light green, and it is basically no longer growing. At Breaking the fruit is one-fifth red and begins to enter the color changing period. It is suitable to apply foliar fertilizer to improve the hardness of the strawberry when it is mature. Turning-1 is two-fifths red strawberry, and Turning-2 is three-fifths red strawberry. At Ripe the strawberry is approximately four-fifths red, indicating it is ready for harvest, particularly for off-site sales. At Full ripe the strawberry is dark red and is completely ripe. Completely ripe strawberries offer the best taste but are not ideal for storage and transportation. Therefore, the Full ripe stage is considered the harvest period for local sales. The patches of strawberries with different ripeness are shown in Figure 1B. The details of the dataset are shown in Table 1.

2.2 Annotation validation

The strawberry ripeness labels are manually annotated, and the quality of the annotation results directly impacts the effectiveness of subsequent classification. Therefore, it is necessary to verify the accuracy of manual labels. Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) is a nonlinear data dimensionality reduction algorithm (McInnes et al., 2018). It can map the structural features of high-dimensional space x_i to low-dimensional space y_i for representation, and preserve the global structure of the data well. Through low-dimensional data visualization, potential relationships among raw data can be observed. We input the strawberry patches into UMAP for dimensionality reduction, and then observe the distribution of strawberries.

Let $X = \{x_1, \ldots, x_N\}$ be the input data set. First, we use the nearest neighbor or approximate nearest neighbor algorithm to obtain the k nearest neighbor set $\{x_{i1}, \ldots, x_{ik}\}$, and then for each x_i, we use Eq. (1) and (2) to find the nearest neighbor distance ρ_i and the normalization factor σ_i.

$$\rho_i = min\{d(x_i, x_{ij}) | 1 \le j \le k, d(x_i, x_{ij}) > 0\} \quad (1)$$

$$\sum_{j=1}^{k} exp(\frac{-max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}) = log_2(k) \quad (2)$$

TABLE 1 Strawberry ripeness classification dataset.

Ripeness category	#Training set	#Test set
White	603	178
Breaking	313	83
Turning-1	230	61
Turning-2	230	64
Ripe	359	116
Full ripe	437	149
Total	2172	651

In high-dimensional space, the distance probability is expressed as Eq. (3) and (4).

$$p_{i|j} = exp(\frac{-max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}) \tag{3}$$

$$p_{ij} = p_{i|j} + p_{j|i} - p_{i|j}p_{j|i} \tag{4}$$

In the low-dimensional space, the distance probability is expressed as Eq. (5), where y_i, y_j are low-dimensional space data, a≈1.93, b≈0.79 are hyperparameters.

$$q_{ij} = (1 + a(y_i - y_j)^{2b})^{-1} \tag{5}$$

Finally, a low-dimensional representation of UMAP is obtained by minimizing the cross-entropy cost function, which can be expressed as Eq. (6).

$$CE(X, Y) = \sum_i \sum_j \left[p_{ij}(X) log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + \left(1 - p_{ij}(X)\right) log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right) \right] \tag{6}$$

After resizing the strawberry patches to a size of 30×40 pixels, the pixel values of each patch were inputted into UMAP as the original high-dimensional data for 1000 iterations. The algorithm was implemented by umap of the python third-party tool library. The size of local neighborhood and effective minimum distance were respectively set to 25 and 0.4 for iteration. By reducing the initial data to three-dimensional space through the UMAP algorithm, we can observe the distribution of strawberries with different ripeness levels (Figure 2). Strawberries at different ripeness levels exhibit distinct boundaries and tend to cluster together based on their ripeness. This observation confirms the correctness of strawberry image annotation to a certain extent. But some points have large deviations, and we checked the strawberry patch annotations corresponding to these points. Then based on this result, the annotations of some images in the dataset were modified to improve the quality of manual annotations, making them more suitable for subsequent training tasks.

2.3 The overall processing flow of strawberry image

The image processing flow is shown in Figure 3. First, the initial image is input into the Mask R-CNN network for strawberry instance segmentation, which generates a mask map. Next, each strawberry instance is segmented using the corresponding mask and divided into four sub-regions to extract features. Finally, the extracted feature values are input into a classifier to determine the ripeness level, resulting in the final visualization on the initial image. The ripeness detection of strawberries can be completed through the above three steps.

2.4 Strawberry detection model

Convolutional neural networks have strong feature extraction capabilities. However, in common convolution operations, the

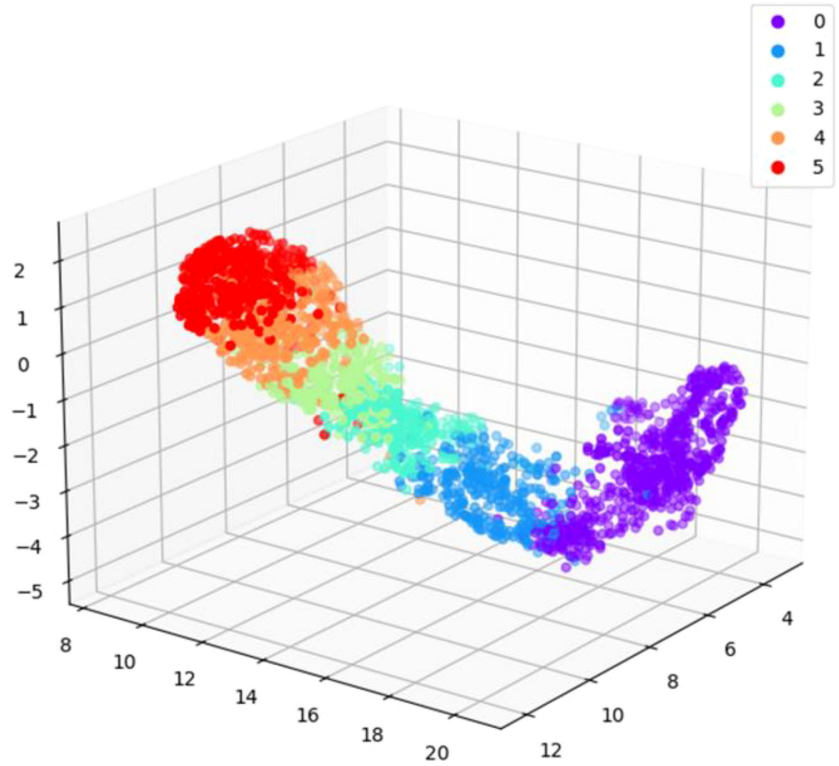


FIGURE 2
3D visualization of partial data sets on UMAP. 0 to 5 indicates increasing ripeness.

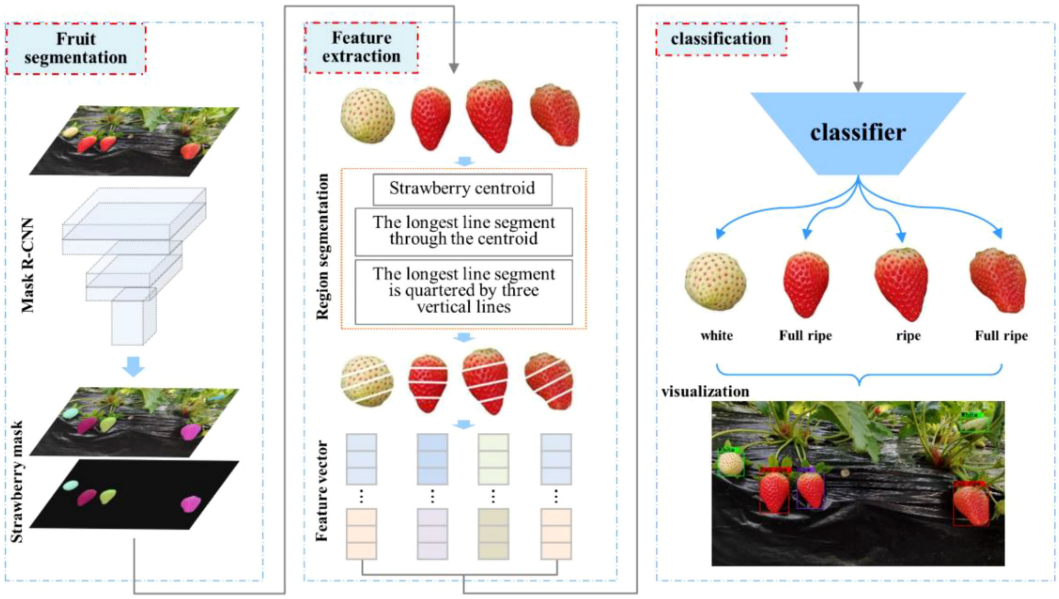


FIGURE 3
Flow chart of strawberry image processing.

convolution is typically performed using multiple sets of convolution kernels of the same size, and the individual channels are then summed to obtain feature maps. The common convolution operation mode is the same, resulting in a limited richness of the learned feature representation. Therefore, the final segmentation results may exhibit shortcomings such as unclear object edges and incomplete segmentation of large objects (Pérez-Borrero et al., 2020). However, the utilization of self-calibrated convolutions can to a certain extent mitigate the above target segmentation issues. The Mask R-CNN instance segmentation model with self-calibrated convolutions will be explained in detail below.

2.4.1 Self-calibrated convolutions

A larger receptive field means that CNN can extract richer semantic information. In the traditional convolution process, the convolution kernels in same size result in fixed receptive fields, which may lack the capability to capture higher-level semantic information from a larger receptive field. The idea of self-calibrated convolution is to use deep features with a larger receptive field (such as strawberry advanced global information) to calibrate shallow features with richer position information (such as strawberry shape contour information) (Liu et al., 2020). The conventional convolutional layer applies a convolution operation to the feature map using a set of convolution kernels (K) of identical size. The self-calibrated convolution technique involves dividing the set of convolution kernels (K) into four parts, K1, K2, K3, and K4, and each part performs distinct convolution operations. Assuming that the number of input and output channels is the same, and the shape of K is (C, C, w, h), then the shape of K1 to K4 is (C/2, C/2, w, h). The details are shown in Figure 4. The input feature maps are divided into two parts, Part A and Part B. The K2 branch feature maps are first down-sampled to make it have a larger receptive field, and then convolution operation and up-sampling are performed with K2. Subsequently, the upsampling results are added to the feature maps of part B, and these results are then mapped to a weight value ranging from 0 to 1. This weight value assists in the convolution operation of the K3 branch, thereby achieving the goal of calibration. Finally, Part A and the calibrated Part B are concatenated after K1 and K4 convolution operations to obtain the final output feature maps.

The self-calibrated convolutions can effectively expand the receptive field and make the target positioning more complete and accurate without introducing additional parameters and complexity. The growth of strawberries in the field is influenced by a multitude of environmental factors, which often leads to variations in their sizes. The receptive field of common convolution is fixed and cannot adapt to changes in strawberry size. To address this limitation, the self-calibrated convolutions module is introduced to enhance the feature extraction results.

2.4.2 Mask R-CNN combined with self- calibrated convolutions

Mask R-CNN (He et al., 2017) is a convolutional neural network designed for instance segmentation tasks, and it can segment fruits from complex natural environments (Ge et al., 2019; Yu et al., 2019; Huang et al., 2020). Mask R-CNN uses ResNet50/ResNet101 (He et al., 2016) as the backbone network and FPN (Lin et al., 2017) as the neck. Its head is the Faster R-CNN (Ren et al., 2017) head and adds a Mask head branch for pixel-level image segmentation. In order to reduce the computational cost, ResNet50 is selected as the backbone network. The Mask R-CNN network structure is shown in Figure 5A.

To enhance the performance of Mask R-CNN and achieve more accurate strawberry segmentation, the aforementioned self-calibrated convolutions are integrated into the original network. ResNet50 is constructed by stacking multiple building blocks, which consist of convolutional blocks and identity blocks. The architectural details of

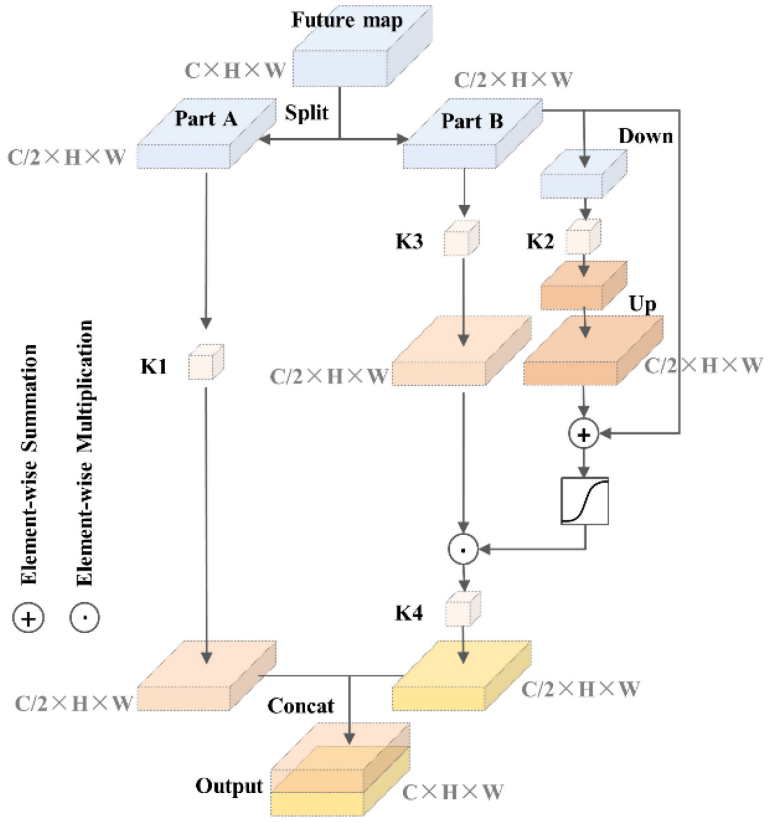


FIGURE 4
Self-calibrated convolutions structure.

ResNet50 can be found in Figure 5B. It is worth mentioning that in Figure 5B, the last average pooling layer and fully connected layer of the original ResNet50 architecture are omitted. Convolutional block has a structure similar to identity block, which consists of a series of 1 × 1 convolution and 3 × 3 convolution, but the former has one more 1 × 1 convolution calculation in upper branch, as shown in Figure 5C. The self-calibrated convolution module can improve the network feature extraction results, so the convolution calculation of 3 × 3 convolution layers in all building blocks are replaced by self-calibrated convolutions.

2.4.3 Model training

The training of DL model performed under the environment of Intel(R) Core(TM) i7-10700KF CPU @ 3.80GHz, 10 GB NVIDIA GeForce RTX 3080 GPU and 32 GB of RAM. The network was built through MMDetction open source tool library on the basis of PyTorch DL framework. In the training process, the horizontal flip data augmentation was performed randomly to prevent overfitting. The SGD optimizer was used for back-propagation to update the network parameters. The learning rate decay strategy was applied in the model training, and the learning rate was multiplied by 0.1 at the 15th, 20th, and 25th epoch to gradually reduce the learning rate. The model had been converging when the epoch was set as 30, so we saved the training results of each epoch and selected the best one on the validation set as test model. The specific hyperparameters are shown in Table 2.

2.5 Feature extraction method

First, the RGB images were converted to HSV and Lab color spaces, and the color features of strawberry patches were counted. Then the change relationship between the color mean of each channel and the ripeness level can be observed in Figure 6. The ordinate in the figure represents the mean value of strawberry foreground pixels, and the abscissa from 0 to 5 represents the gradually increasing ripeness. It can be seen from the figure that the average color values of channels B, G, and L show an obvious decreasing trend with the increase of strawberry ripeness. The mean color values of channels a and S increased significantly with the increase of ripeness. There is a certain correlation between the color feature value of strawberry and its ripeness, among which the channel a is the strongest, but the channels R, b, H, and v are not obvious enough. Channels B, G, L, a, and S are selected for strawberry color feature extraction based on region segmentation to reduce computational complexity and eliminate noise interference in other data.

To extract strawberry features effectively, the strawberry is divided into four sub-regions, and the color mean of each region is extracted as the color feature of the strawberry. Before feature extraction, it is necessary to divide and mark the strawberry, which can be accomplished through the following steps. The specific process is shown in Figure 7A.

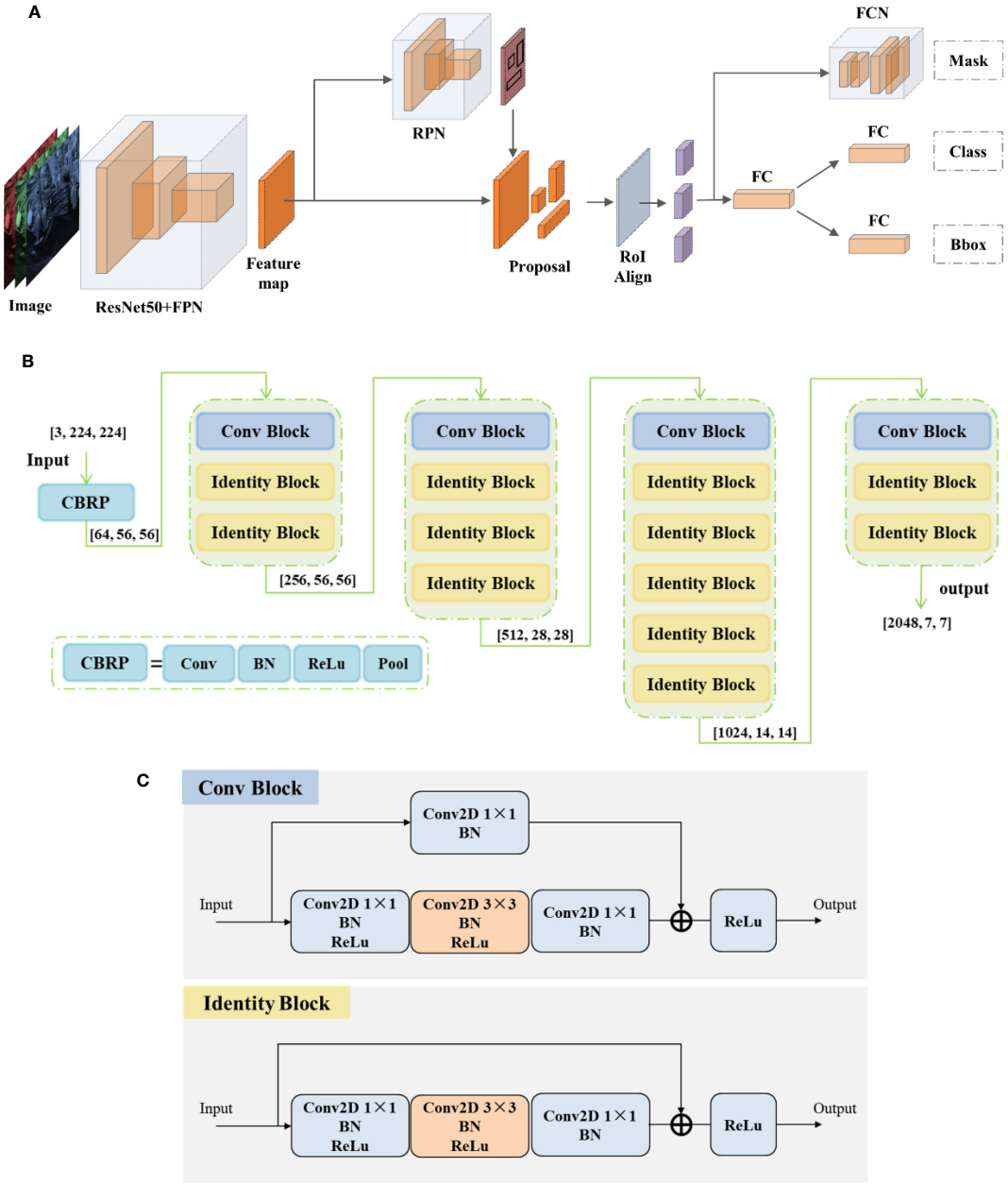


FIGURE 5
(A) Mask R-CNN network architecture. **(B)** ResNet50 network architecture. **(C)** Building block architecture.

TABLE 2 Hyperparameters of model training.

Hyperparameter	Value
Learning Rate	0.02
Momentum	0.9
Optimizer	SGD
Batch Size	3
Epoch	30
Wormup Iterations	500
Decay Steps(epoch)	[15,20,25]

Step 1: Determine the strawberry centroid. After processing the original image with Mask R-CNN, a masked binary image of strawberry will be generated. The mask coordinate (x_i, y_i) and Eq. (7) are used to determine the center of mass coordinate $C\,(x_0, y_0)$ of strawberry.

$$\begin{cases} x_0 = \frac{\sum_{i=1}^{N} p_i x_i}{\sum_{i=1}^{N} p_i} \\ y_0 = \frac{\sum_{i=1}^{N} p_i y_i}{\sum_{i=1}^{N} p_i} \end{cases} \tag{7}$$

where N is the total number of strawberry pixels, and p_i is the value of the i-th pixel.

Step 2: Find the longest line segment through the centroid. The outer contour point P_i of the strawberry binary image can be

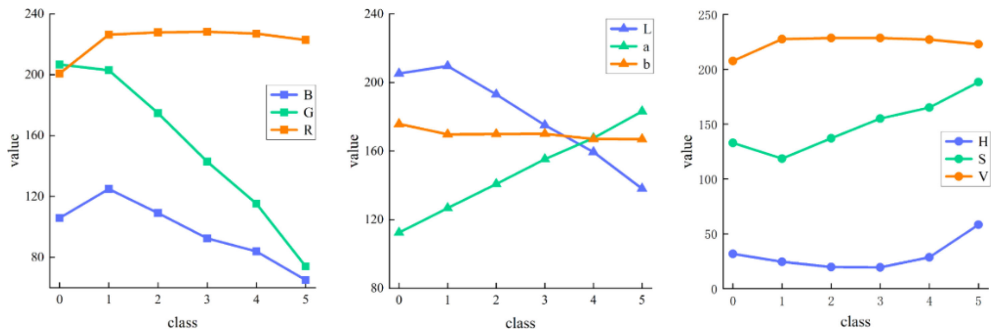


FIGURE 6
Mean values of different color spaces. 0 to 5 indicates increasing ripeness.

expressed as $\{(x_i, y_i) \mid 1 \le i \le M\}$, and by traversing each outer contour point, M straight lines passing through the centroid C (x_0, y_0) can be obtained, which can be expressed as $\{(x, y) \mid A_i y + B_i x + C_i = 0, 1 \le i \le M\}$. These lines are traversed, and the distance from each contour point to the line is obtained using Eq. (8). Find the contour point P_i' at the minimum distance and use it as another approximate intersection of this line with the contour. When the minimum distance is 0, it indicates that the point is on the line (excluding the contour points that construct the line). This results in a total of M approximate intersections. Finally, each line has two intersections with the strawberry outline. The farthest set of intersection points are connected and used as the longest line segment PP' through the strawberry's centroid.

$$d = \frac{\left|A_i x_j + B_i y_j + C_i\right|}{\sqrt{A_i^2 + B_i^2}}, (1 \le i \le M, 1 \le j \le M) \tag{8}$$

Step 3: Find three vertical lines to divide the longest line segment into four equal parts. We can easily find the three coordinate points a, b, c on the line segment PP' such that PP' is divided into four equal parts. Then through these three points, three vertical lines l_a, l_b, l_c perpendicular to the line segment PP' are obtained. Each vertical line approximately intersects with the strawberry contour at two points, which can be obtained by calculating the approximate intersection point in step 2.

Step 4: Area marking. The three sets of intersection points in step 3 are connected respectively, and the strawberry is divided into four sub-regions. The centroid coordinate C of each sub-region is calculated separately by Eq. (7). The sub-regions are sorted from bottom to top according to the value of y_0 and marked as R_1, R_2, R_3, R_4. The purpose of region marking is to enable subsequent feature extraction in this order.

Figure 7B shows some examples of results after the strawberry region is automatically divided. It can be seen that each sub-region of strawberry is well segmented by three line segments, and the four sub-regions are correctly marked in order.

2.6 Classification method

According to the extracted strawberry features, selecting a classifier that matches the data type can maximize the classification effect. Strawberry features are high-dimensional data and have nonlinear characteristics. To fully leverage the performance of the classifier and enhance the accuracy of ripeness classification, the SVM (Support Vector Machine) was considered first. SVM is a linear classifier suitable for processing high-dimensional data. Due to its advantages of fast training speed, high accuracy, and good robustness, SVM has gained extensive usage in the field of image classification (Tu et al., 2018; Dhakshina Kumar et al., 2020). For comparison, we tried other classic machine learning methods, including LR (Logistic Regression), KNN (K-Nearest Neighbors), RF (Random Forest), and finally obtained the best classifier by comparative analysis. We used 5-fold hierarchical cross-validation and grid search methods to optimize the parameters of these classifiers. The optimized parameters were used as the final parameters of the model (Table 3).

3 Results

3.1 Evaluation methods

For segmentation tasks, we will compare the segmentation effects of Mask R-CNN's backbone network before and after adding self-calibrated convolutions. For the task of strawberry ripeness classification, we will evaluate the classification performance of different classifiers using various combinations of color channels. Subsequently, we will identify the optimal classifier based on the results. Then we will use the optimal classifier to evaluate the classification effect of different feature extraction methods to illustrate the superiority of our proposed feature extraction method. Finally, the proposed method will be compared with the common CNN.

The following is an introduction to the model evaluation indicators. AP, AP.50, AP.75 are used to evaluate the segmentation effect of the model. F1 and accuracy are used to evaluate the classification performance of the classifier. AP represents the mean of the average precision under 10 IoU thresholds from 0.50 to 0.95 with 0.05 intervals, which is the most important evaluation metric for MS COCO competition. AP.50 represents the average precision when IoU=0.50, and AP.75 represents the average precision when IoU=0.75. IoU is the intersection and union ratio of the mask area. The average precision is the area under the P-R curve, which can be

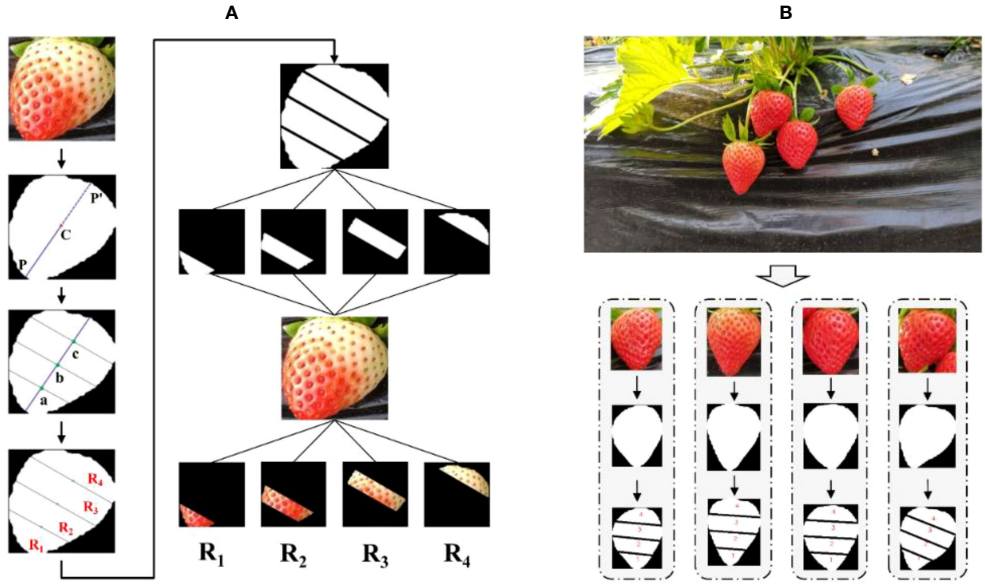


FIGURE 7
(A) Flow chart of strawberry region segmentation. **(B)** Example of strawberry region segmentation results.

obtained from Eq. (9). P(r) is the P-R curve obtained from precision and recall. TP represents the number of positive samples correctly predicted. TN represents the number of negative samples correctly predicted. FP represents the number of positive samples that were incorrectly predicted. FN represents the number of negative samples that are incorrectly predicted.

$$\begin{cases} \text{Precision} = \frac{TP}{TP+FP} \\ \text{Recall} = \frac{TP}{TP+FN} \\ \text{Average Precision} = \int P(r)dr \end{cases} \tag{9}$$

$$\begin{cases} F1 \;\; Score = \frac{2\times Precision\times Recall}{Precision+Recall} \\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \end{cases} \tag{10}$$

3.2 Detection performance of instance segmentation model

To assess the impact of the Mask R-CNN model improvement, we conduct a comprehensive comparison by considering the training phase, testing phase, and the final strawberry segmentation results. This allows us to observe the effectiveness of the model before and after the proposed enhancements. The loss curve and training error curve of the model are shown in Figure 8. It can be seen from the figure that the loss of the model begins to stabilize around 25 epochs, and the model has converged at 30 epochs. After incorporating self-calibrated convolutions to the original ResNet50 backbone network, the model exhibits lower loss during convergence, indicating an improved fit of the model. Additionally, it is evident that the training error of SCNet50, after incorporating self-calibrated convolutions, is lower than that of ResNet50. This demonstrates that the inclusion of self-calibrated convolutions leads to an improvement in model accuracy to a certain extent.

During the training process, the best performing model on the validation set was saved. Then the final performance of the model was verified on the test set. The test results of the model are shown in Table 4. Mask R-CNN utilizing SCNet50 as the backbone network exhibits a higher average precision compared to using ResNet50. The AP of SCNet50 reaches 0.937, which is 0.039 higher than that of ResNet50, and the AP.50, AP.75 are also improved by 0.021 and 0.032, respectively. But in inference speed, the FPS of SCNet50 is reduced, which is within our allowable range. The feature extraction ability of ResNet50 is improved after adding self-calibrated convolutions. Not only did the model perform better on training, it also performed well on testing. This indicates its strong generalization ability, but at the same time it also increases a certain time cost.

The final segmentation results of strawberry are shown in Figure 9. The strawberry marked by the yellow box in the first row of picture has missed detection. The reason may be that the surrounding background color is similar to the strawberry. The strawberry in the picture on the right is successfully detected because SCNet50 extracts richer semantic information. It is still capable of identifying the target even in cases

TABLE 3 The main parameters of the different classifiers.

Classifier	Param
LR	'c': 0.7, 'solver': 'newton-cg, 'penalty': l2
KNN	'n_neighbors': 12
RF	'max_depth': 20, 'n_estimators': 35
SVM	'C': 10, 'kernel': 'rbf', 'gamma': 0.0005

* 'c': reciprocal of penalty term coefficient, 'penalty':penalty item, 'solver': optimization method, 'n_neighbors': number of neighbors, 'max_depth': decision tree maximum depth, 'n_estimators': number of decision trees, 'C': penalty coefficient, 'kernel': kernel function, 'gamma': gamma coefficient.

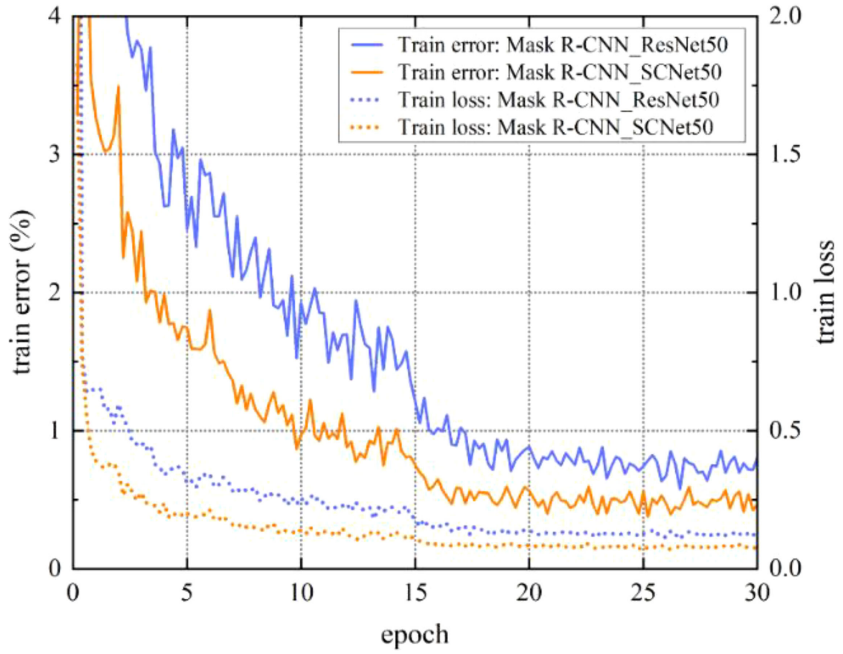


FIGURE 8
Model training loss and training error. SCNet50 is the backbone network with self-calibrated convolutions.

where the background and the target have similar colors. In the second row of the figure, the overlapping strawberries marked by the yellow box on the left are not completely segmented. In the third row of the picture, the strawberry marked by the yellow box on the left is incorrectly identified as part of the strawberry because the strawberry is occluded by the leaf. These erroneous segmentations will have an impact on subsequent strawberry ripeness classifications. From Figure 9D, it can be observed that the aforementioned erroneous segmentations have been effectively improved, and overall, the edges of the strawberries are more detailed. By adding self-calibrated convolutions, the model has a larger receptive field and can generate richer feature representations, making target positioning more accurate.

To further analyze the model's robustness against occlusion, we have compared the strawberry segmentation accuracy under different occlusion areas (Table 5). We manually counted the number of strawberries covered by stalks, leaves, and other strawberries in the test set, dividing them into two categories: 0-20% and 20-50% based on occlusion area. As shown in Table 5, SCNet50 demonstrates higher accuracy in segmenting strawberry when faced with occlusion interference, particularly under the 20-50% occlusion area where its mean IoU improves by 0.056 compared to ResNet50. Examples of the segmentation results can be found in Figure 10.

3.3 Strawberry color feature extraction

We employ the approach outlined in Section 2.5 to extract the color features of strawberries. By calculating the color mean of each sub-region in each channel, we can observe the trends and variations in these color features. The results are shown in Figure 11. The ordinate in the figure represents the average pixel value of the strawberry sub-region, and the abscissa from 0 to 5 represents the gradually increasing ripeness. With the change of sub-regions R_1 to R_4, the color feature values in channels B, G, L show an increasing trend at the same ripeness stage, and show a decreasing trend in channels a and S. In addition, the color feature values of the B, G, and L channels have similar trends with ripeness. Among them, R1, R2, and R3 decrease with increasing ripeness, while R4 gradually increases in the first three ripeness stages and then gradually decreases in the last three ripeness stages. Channel a and S have a gradual rise in overall. Among them, R_4 gradually decreases in the first three ripeness stages in the channel S, and the latter three ripeness stages gradually increases. As the strawberry ripeness increases, we observe a systematic change in the color feature values of the different sub-regions across each channel. This consistent pattern proves beneficial for the effective functioning of subsequent classifiers.

3.4 Classification of strawberry ripeness

The classification results of strawberry ripeness are shown in Table 6. From the perspective of each color channel, Channel a achieves the highest classification accuracy when considered individually. Among the classifiers, SVM shows the best performance with an accuracy of 0.850. It can be easily explained from Figure 11. The color feature values of Channel a increase with the ripeness, indicating a strongest correlation and providing favorable conditions for classifier judgment. In the combined

TABLE 4 The test results of instance segmentation model.

Model	Backbone	AP	AP.50	AP.75	FPS
Mask R-CNN	ResNet50	0.898	0.958	0.937	19.4
	SCNet50	0.937	0.979	0.969	18.2

SCNet50 is the backbone network with self-calibrated convolutions.

channels, as the number of channels increases, the accuracy of the LR and SVM classifiers gradually increase. However, in the KNN classifier, BGa, GaS, BGaS, and BGLaS under the combination channels have decreased accuracy compared to Ga. This shows that the features of the B, S, and L channel have a certain interference effect on the classification effect of KNN. In the RF classifier, the results of GaS have decreased compared to Ga, and the results of BGLaS have decreased compared to BGaS. This indicates that the feature information from the S and L channels is redundant for the classifier, and including this data dose not lead to an improvement in performance. When all channels are combined, SVM achieves the highest classification accuracy of 0.866, demonstrating its effectiveness in handling high-dimensional data. The classification performance of RF is second only to SVM, with an accuracy of 0.861 achieved using the BGaS channel. The inaccurate classification may be due to abnormal distribution of surface color in some strawberries or the strawberries not being in a downward fruit-hanging posture overall. These will cause outliers in feature extraction, which will lead to wrong classification.

Figure 12. is the confusion matrix when RF and SVM respectively obtain the best results. Except in Breaking (label 1) and Turning-1 (label 2), SVM is better than RF. According to the above analysis, SVM is selected as the suitable classifier.

The final detection results of strawberry ripeness is visualized (Figure 13). It is worth mentioning that the probabilities in the results represent SVM classification probabilities. It is important to mention that in the left image of the second row, there was an undetected green strawberry. This is because it is not considered in the model training and does not belong to any of the six ripeness categories. Strawberries can be detected in both frontlighting and backlighting environments, as shown in the first row of images. Even under slight occlusions, as depicted in the second row, the strawberry ripeness level can still be successfully identified. However, in the right image of the first row, the strawberry is severely occluded, and the instance segmentation model failed to detect the strawberry, resulting in the inability to recognize its ripeness subsequently. In the last image, the same strawberry was detected twice, resulting in duplicate detections. This is because the strawberry is occluded by the stalk, and the instance segmentation model mistakenly recognizes it as two instances, causing subsequent tasks to treat it as two objects for processing. In general, the overall performance of the model is largely affected by the segmentation performance. When the first-stage segmentation model failed to detect or misdetected objects, the model was unable to predict strawberry ripeness, so the predictions could not be reversed.

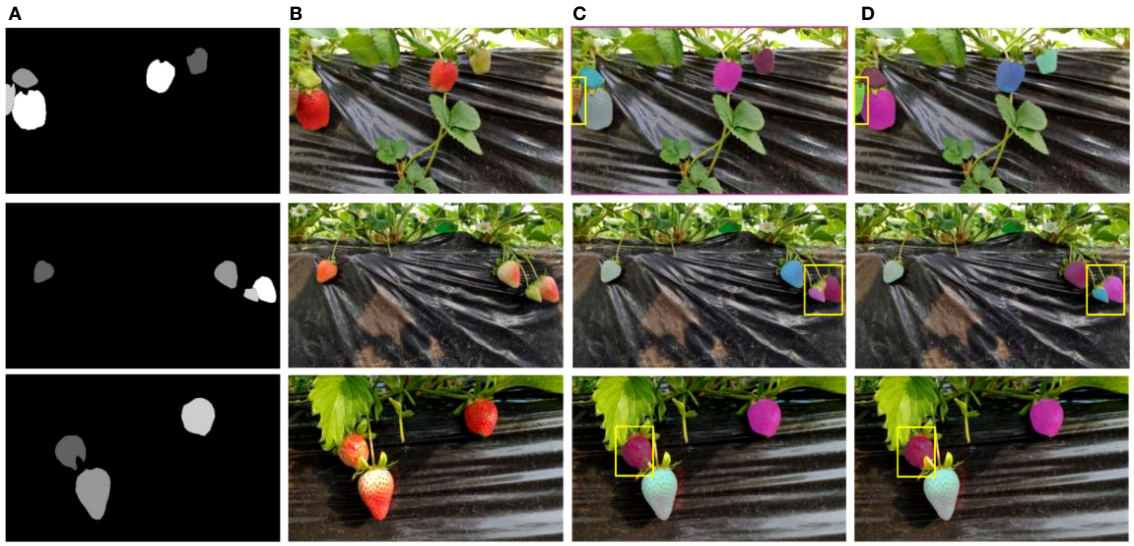


FIGURE 9
Strawberry segmentation results. The yellow rectangles indicate the area to be compared. **(A)** Ground truth. **(B)** Initial images. **(C)** ResNet50 results. **(D)** SCNet50 results.

TABLE 5 Mean IoU comparison of models under different occlusion areas of strawberries.

Model	Backbone	0~20%	20~50%
Mask R-CNN	ResNet50	0.896	0.849
	SCNet50	0.918	0.905

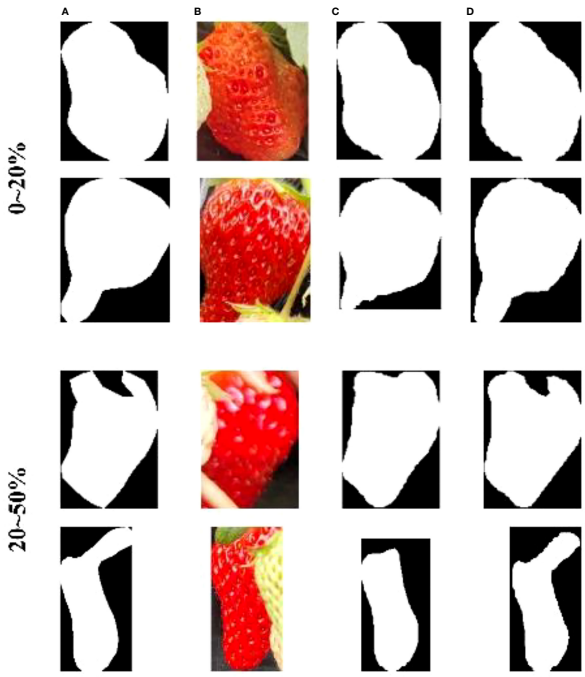


FIGURE 10
Segmentation examples under different occlusion areas. **(A)** Ground truth. **(B)** occlusion strawberries. **(C)** ResNet50 results. **(D)** SCNet50 results.

3.5 The effect of different sub-regions on classification results

Table 7 is the classification results of strawberry ripeness under the SVM classifier based on the color features of different sub-regions. In terms of the single sub-regions' effects, except for the B channel, R_3 consistently exhibits the highest classification accuracy. In terms of the combination effects of sub-regions, as the number of sub-regions increases, the feature information is more diverse and comprehensive. Consequently, this leads to enhanced classification accuracy for each single channel. In order to further analyze the specific contributions of each sub-region to different ripening stages of strawberries, we extracted the color feature values under the combined channel BGLaS. Subsequently, we utilized the SVM classifier to classify the ripeness. The number of correct classification labels was counted, as shown in Table 8. First of all, the sub-region with the highest classification accuracy is R_3, which is 68.15%. This is consistent with the result that R_3 in Table 7 basically maintains the highest accuracy in a single channel. In the White stage, the accuracy of R_2 demonstrates the highest performance, while in the Breaking and Turning-1 stages, the accuracy of R_1 exhibits the highest level of accuracy. The classification effect of Turning-2 mainly depends on R_3, which contributes the most to the classification effect of this stage. Ripe and Full ripe both bring the most obvious classification effect under R_4.

The increase of strawberry ripeness is basically accompanied by the continuous expansion of the surface red area from bottom to top, as shown in Figure 14. During the early stages of strawberry ripeness, the red area is small. The color change primarily occurs in the lower half of the strawberry, while the color of the upper half remains relatively unchanged. Therefore, the color differences of White, Breaking and Turning-1 in the sub-regions R_1 and R_2 are relatively large, which is conducive to the judgment of the three ripeness levels. In the later stages of strawberry ripening, the lower half of the strawberry basically turns red, and the green area of the upper half gradually diminishes. This color difference is also helpful in judging the ripeness of Turning-2, Ripe and Full ripe. Therefore, when considering Table 8, it becomes evident that R_1 and R_2 play a significant role in determining the first three ripeness levels. On the other hand, R_3 and R_4 exhibit greater influence in discerning the last three ripeness levels. In Table 8, the accuracy of each sub-region of the White stage is higher, because the whole surface of the strawberry in the White stage is light green. No matter under which sub-region, its color value is obviously different from other stages.

3.6 Comparison of different classification methods

To validate the superiority of the proposed feature extraction method, we compared it with the common manual feature extraction methods. Typical manual feature extraction methods can be divided into two categories: 1) taking each pixel as a feature value; 2) taking the pixel mean of the foreground target as a feature value. Table 9 shows the classification results of different strawberry color feature extraction method. Method 1 is to resize the strawberry block cropped by the rectangular frame to 30×40, while method 2 is to take the mean value of the segmented strawberry foreground pixels as the feature value. Table 9 clearly demonstrates that the accuracy of the proposed method is higher than other methods across all channels. The highest accuracies of method 1 and method 2 are 0.811 and 0.826, which are 0.055 and 0.040 lower than the proposed method respectively. Method 1 primarily emphasizes full-image pixel classification, placing excessive emphasis on pixel position information. This approach may result in inaccurate classification, particularly when dealing

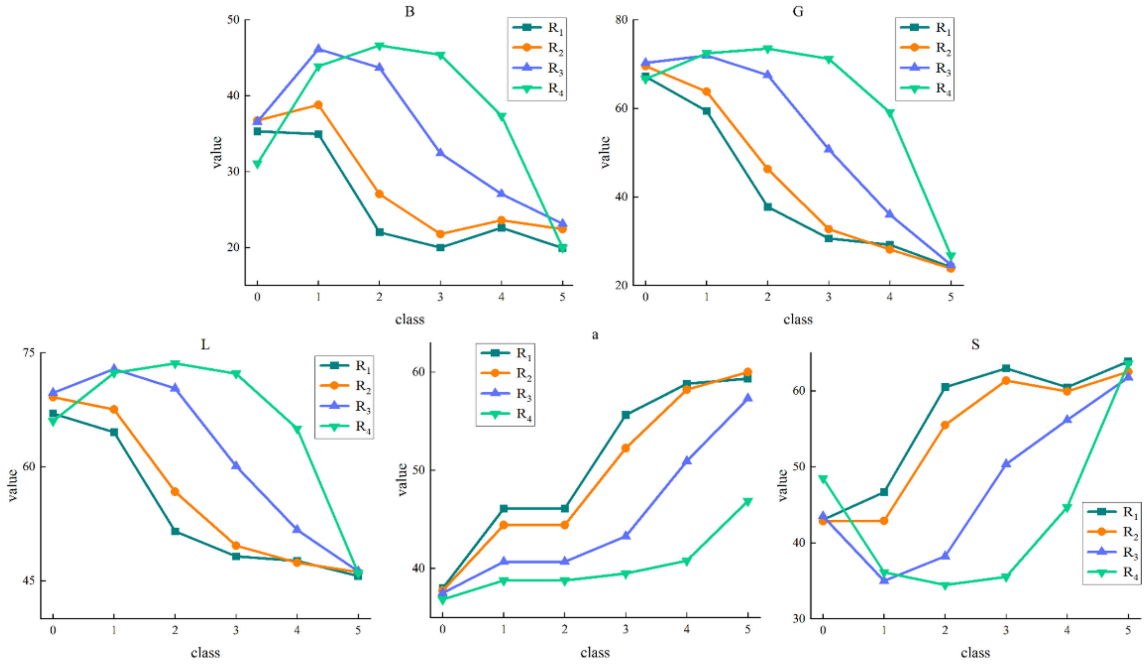


FIGURE 11
Variation trend of color feature values in strawberry sub-regions. 0 to 5 indicates increasing ripeness.

with horizontally arranged strawberries that undergo deformation during the resizing process. Method 2 primarily emphasizes foreground pixel classification and relies on the color mean value as a classification feature. However, it overlooks pixel position information, which ultimately results in inaccurate classification. While the color feature extraction based on region segmentation in the proposed method takes into account both the positional information of the red region as it changes with ripeness and the pixel-level information. Therefore, the proposed method can obtain more informative features for strawberry ripeness classification.

The fruit ripeness classification based on CNN is also a widely adopted method. Therefore, we conducted a comparison between the proposed method and commonly used CNN models. The parameter settings of CNN model training are consistent. The learning rate and batch size are 0.001 and 16, respectively. The model uses the SGD optimizer and iterates for 30 epochs to train the parameters. The learning optimization strategy adopts the MultiStepLR method, and the learning rate decays at the 18th, 24th, and 27th epoch respectively. Gaussian blur and horizontal flip data augmentation are randomly performed on the image during training. The experimental results are shown in Table 10. Except that the F1 score of the proposed method is lower than AlexNet and ResNet18 in the Turning-1 and Turning-2 stages, the rest of the ripeness stages show better classification results. The classification error rate of the proposed method is primarily concentrated in the Turning-1 and Turning-2 stages, because there are more strawberries in transitional ripeness stages between Turning-1 and Turning-2 stages. Their features are very similar, which can easily result in the classification results to swing between these two stages.

4 Discussion

In this study, we have developed a method that combines Mask R-CNN and region segmentation to accurately assess the ripeness of strawberries in the field. The method proposed in this paper is compared with existing research work (Table 11). In most cases, managing strawberry planting, including monitoring fruit growth status and predicting fruit yield, needs to be done in a natural

TABLE 6 Classification accuracy of different color channels.

	B	G	L	a	S	Ga	BGa	GaS	BGaS	BGLaS
LR	0.651	0.768	0.693	0.842	0.704	0.840	0.850	0.849	0.854	0.857
KNN	0.645	0.783	0.724	0.844	0.696	0.839	0.828	0.829	0.823	0.819
RF	0.622	0.791	0.705	0.846	0.659	0.860	0.860	0.856	0.861	0.849
SVM	0.639	0.770	0.710	**0.850**	0.697	0.854	0.863	0.859	0.863	**0.866**

Values in bold mean the highest classification accuracy under single channel and combined channel among all classifiers.

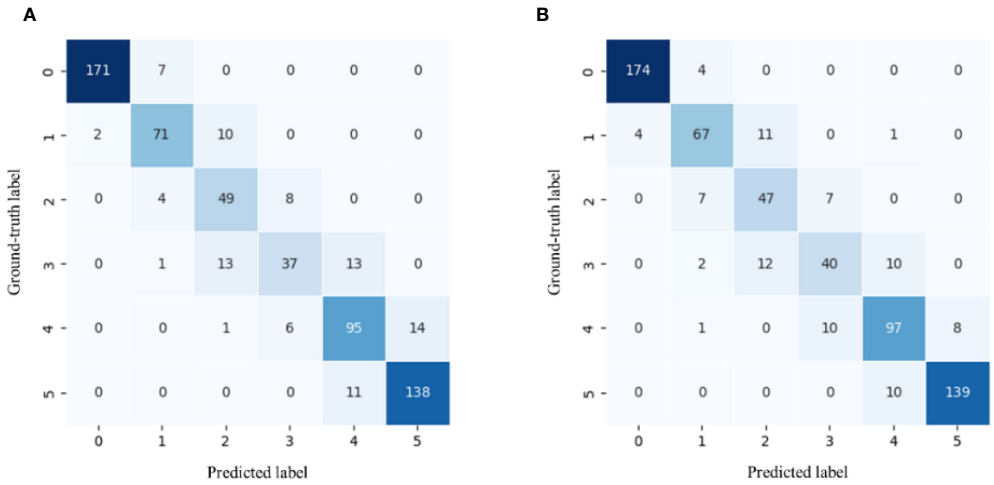


FIGURE 12
(A) RF confusion matrix. **(B)** SVM confusion matrix. 0 to 5 indicates increasing ripeness.

environment rather than indoors. In earlier studies, the majority of research was conducted within the confines of an indoor setting. This highly structured environment allowed for greater control, thereby facilitating the extraction of strawberry features and subsequent analysis(Zhang et al., 2016; Indrabayu et al., 2019; Su et al., 2021). Compared to the unstructured outdoor environment, the complexity of lighting, background similarity to fruit, overlapping fruit, and fruit occlusion by plants are some of the uncertain factors that can pose a challenge (Yu et al., 2019; Pérez-Borrero et al., 2020). The presence of these phenomena poses a challenge in precisely segmenting the target fruit from the surrounding environment, thereby impacting the subsequent research work. The significant improvement of AP in Table 4 is specifically reflected in the model's miss rate of strawberries and the

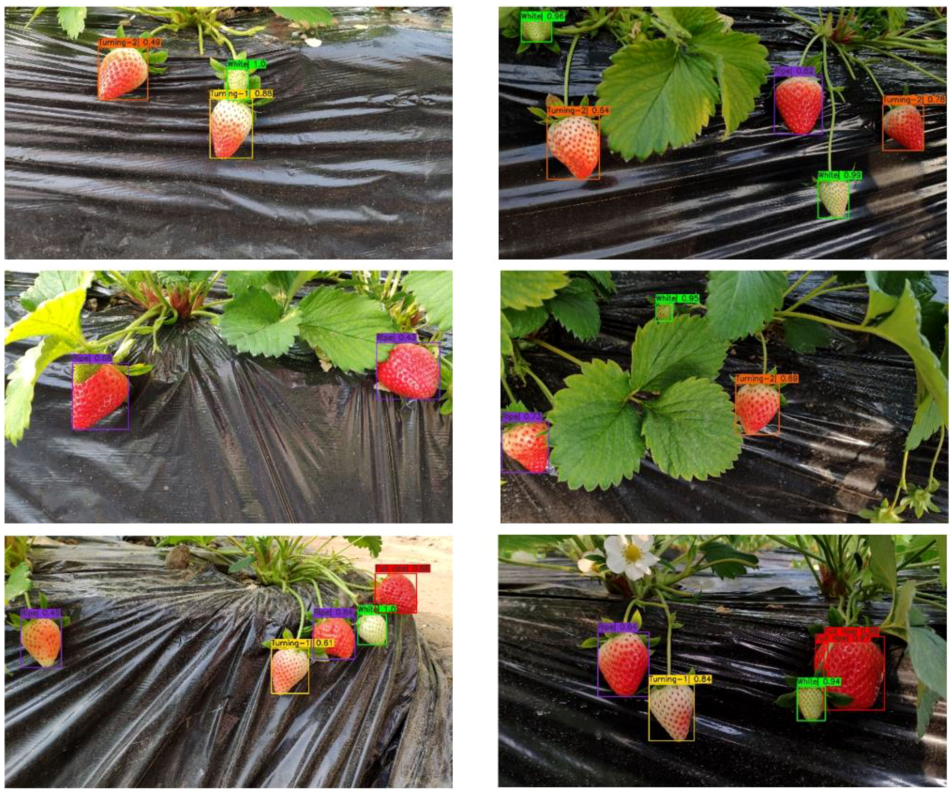


FIGURE 13
The visualization results of strawberry ripeness detection.

TABLE 7 Classification results of different sub-regions under single channel.

	R_1	R_2	R_3	R_4	R_3R_4	$R_1R_3R_4$	$R_2R_3R_4$	$R_1R_2R_3R_4$
B	0.488	0.493	0.481	0.487	0.588	0.630	0.625	0.639
G	0.601	0.604	0.621	0.593	0.694	0.766	0.768	0.770
L	0.524	0.539	0.553	0.551	0.639	0.682	0.699	0.710
a	0.590	0.642	0.710	0.690	0.776	0.846	0.840	0.850
S	0.510	0.521	0.522	0.502	0.625	0.693	0.671	0.697

TABLE 8 Contribution of different sub-regions to each ripeness stage.

Class (number)	R_1	R_2	R_3	R_4
White (178)	172(96.63%)	174**(97.75%)**	169(94.94%)	164(92.13%)
Breaking (83)	62**(74.70%)**	61(73.49%)	51(61.44%)	46(55.42%)
Turning-1 (61)	41**(67.21%)**	36(59.02%)	26(42.62%)	1(1.64%)
Turning-2 (64)	1(1.56%)	21(32.81%)	37**(57.81%)**	26(40.63%)
Ripe (116)	28(24.14%)	43(37.07%)	74(63.79%)	89**(76.72%)**
Full ripe (149)	124(83.22%)	127(85.23%)	124(83.22%)	136**(91.28%)**
Total (672)	428(63.69%)	437(65.03%)	485(68.15%)	433(64.43%)

Values in bold mean the highest classification accuracy in each ripeness stage.

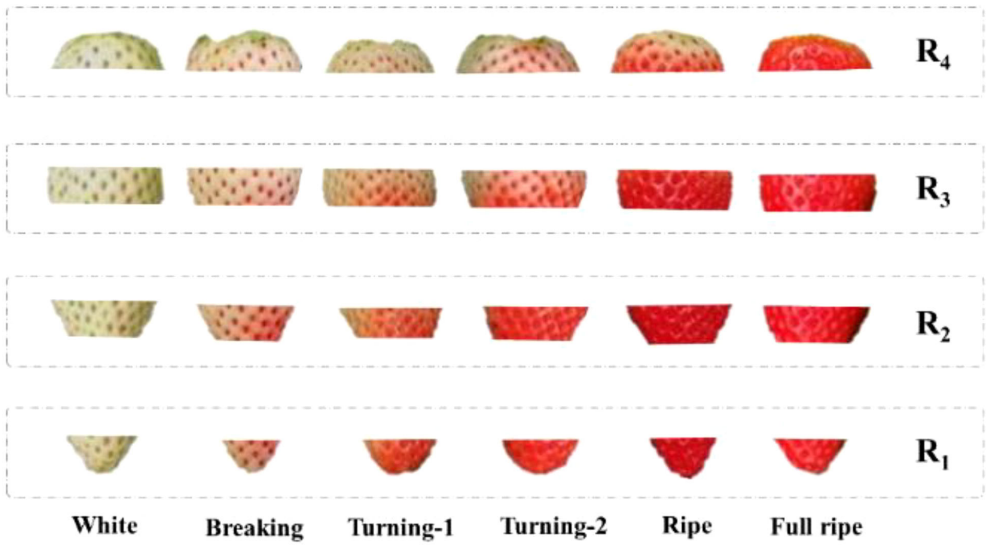


FIGURE 14
Examples of sub-regions at different ripeness levels.

integrity of the segmentation mask. Thanks to the unique architecture of self-calibrated convolution, the model shows the potential of greater adaptability in the face of complex field environments.

Strawberries undergo a brief veraison period and mature rapidly. By utilizing a more comprehensive categorization of ripeness stages, fruit farmers can obtain precise information on fruit growth, enabling them to efficiently seize crop management opportunities such as topdressing and harvesting. In this study, strawberries were categorized into six ripeness levels, providing more comprehensive information on their ripeness than previous studies. Due to the large similarity between some categories (such as

TABLE 9 SVM classification accuracy of different feature extraction methods.

	B	G	L	a	S	Ga	BGa	GaS	BGaS	BGLaS
Method 1	0.612	0.745	0.676	0.762	0.676	**0.811**	0.806	0.800	0.799	0.796
Method 2	0.520	0.692	0.614	0.786	0.561	0.800	0.812	0.821	0.821	**0.826**
Proposed	0.639	0.770	0.710	0.850	0.697	0.854	0.863	0.859	0.863	**0.866**

Values in bold mean the highest classification accuracy for each method.

TABLE 10 Test results of different classification methods.

Ripeness category	AlexNet				ResNet18				Proposed			
	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
White	0.94	0.99	0.96	0.848	0.99	0.93	0.96	0.856	0.98	0.98	0.98	0.866
Breaking	0.86	0.75	0.80		0.73	0.94	0.73		0.83	0.81	0.82	
Turning-1	0.69	0.74	0.71		0.79	0.69	0.79		0.67	0.77	0.72	
Turning-2	0.69	0.72	0.72		0.72	0.61	0.72		0.70	0.62	0.66	
Ripe	0.77	0.81	0.79		0.78	0.81	0.78		0.82	0.84	0.83	
Full ripe	0.93	0.87	0.90		0.92	0.93	0.93		0.95	0.93	0.94	

Acc means accuracy.

TABLE 11 Comparison of different ripeness identification methods.

Source	Classes	Environment	Model	Results
Zhang et al. (2016)	3	Laboratory	SVM	Accuracy: over 85%
Habaragamuwa et al. (2018)	2	Field	DCNN	AP: 88.03%, 77.21%
Indrabayu et al. (2019)	3	Laboratory	SVM	Accuracy: 85.64%
Shao et al. (2020)	3	Laboratory, Field	PLS-DA, LS-SVM	Accuracy: 91.7% ~ 96.7%
Su et al. (2021)	4	Laboratory	1D ResNet, 3D ResNet	Accuracy: 86.03%, 85.29%
Fan et al. (2022)	4	Field	YOLOv5	Accuracy: over 90%
Raj et al. (2022)	3	Laboratory, Field	SVM	Accuracy: over 98%, 71%
Ours	6	Field	Mask R-CNN,SVM	Accuracy: 86.6%

Turning-1 and Turning-2), it is difficult for the classifier to distinguish them, which eventually leads to a decrease in the overall accuracy (Table 10). This phenomenon is also evident in other studies on fruit ripeness. (Saranya et al., 2021; Chen et al., 2022). Categorizing strawberries into 2 to 3 ripeness levels enhances the distinctiveness of their characteristics, facilitating the classifier's judgment and contributing to the high accuracy achieved in previous studies (Habaragamuwa et al., 2018; Shao et al., 2020; Raj et al., 2022). However, the rough ripeness classification will make the strawberry interval span larger. This often leads to missed opportunities for timely topdressing during the intermediate stages of ripeness and the optimal timing for harvest under various sales patterns towards the end of ripeness. We devised a color feature extraction method that incorporates region segmentation, along with a classifier tailored to the feature data, resulting in precise classification of strawberries into six ripeness levels. The method we proposed not only enables the completion of multi-category ripeness distinction, but also ensures high accuracy. This provides important technical support for the precise harvesting operation of strawberries.

5 Conclusion

This study presents a fine recognition method for assessing strawberry ripeness, with the objective of addressing the current issue of coarse classification and emphasizing indoor experimental investigations. It can provide more accurate decision support for

strawberry harvest management. The achievement of fine recognition of strawberry ripeness in the field involves three stages. The first stage is to detect and segment strawberries from images with a deep learning model. We added self-calibrated convolutions to Mask R-CNN to improve the network segmentation effect, and the final AP and AP.50 were 0.937 and 0.979, respectively. The second stage is strawberry color feature extraction. Firstly, to extract relevant features, the change trend of feature values with ripeness was analyzed, leading to the selection of channels B, G, L, a, and S for feature extraction. Subsequently, the strawberry was divided into four sub-regions, and the feature values of each region were individually extracted under the aforementioned color channels. The third stage is ripeness classification. The feature values were input into different classification models for ripeness classification, and finally achieved the best results in the SVM classifier. The classification accuracy of SVM is 0.850 under single channel a and 0.866 under combined channel BGLaS. Through additional experiments, it was observed that sub-regions R_1 and R_2 primarily play a role in identifying strawberry ripeness in the White, Breaking, and Turning-1 stages. On the other hand, sub-regions R_3 and R_4 demonstrated significant contributions in identifying strawberry ripeness in the Turning-2, Ripe, and Full ripe stages.

In summary, the incorporation of self-calibrated convolutions enhances the model's robustness in field environments, leading to improved segmentation outcomes for strawberries. Additionally, the color feature extraction method based on region segmentation effectively captures the distinctive feature information among strawberries of varying ripeness levels, thus enhancing the classifier's ability to differentiate between strawberries at different stages of ripeness. The research findings demonstrate that this method can accurately identify multiple levels of ripeness for strawberries in field conditions, thereby providing more effective guidance for strawberry harvest management.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

CT designed the experiment, conducted data analysis, and wrote the manuscript. XW guided the experiment, provided research ideas, and improved the quality of the manuscript content. XN enhanced the logic and presentation of the Introduction. YeL and YiL processed experimental data and revised figure descriptions. DC and XM contributed to the revision of the manuscript content. SW reviewed and guided the manuscript. All authors contributed to the article and approved it for publication.

Funding

This work was funded by the National Key Research and Development Program of China [No. 2022YFD2001203], the project of agricultural machinery R & D, manufacturing, promotion, application and the integration, and the National Natural Science Foundation of China [No. 32201687]. This work was supported in part by College of Engineering, China Agricultural University (CAU).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abbaszadeh, R., Rajabipour, A., Sadrnia, H., Mahjoob, M. J., Delshad, M., and Ahmadi, H. (2014). Application of modal analysis to the watermelon through finite element modeling for use in ripeness assessment. *J. Food Eng.* 127, 80–84. doi: 10.1016/j.jfoodeng.2013.11.020

Aghilinategh, N., Dalvand, M. J., and Anvar, A. (2020). Detection of ripeness grades of berries using an electronic nose. *Food Sci. Nutr.* 8 (9), 4919–4928. doi: 10.1002/fsn3.1788

Azodanlou, R., Darbellay, C., Luisier, J. L., Villettaz, J. C., and Amado, R. (2004). Changes in flavour and texture during the ripening of strawberries. *Eur. Food Res. Technol.* 218 (2), 167–172. doi: 10.1007/s00217-003-0822-0

Chen, J., Mao, L., Mi, H., Zhao, Y., Ying, T., and Luo, Z. (2014). Detachment-accelerated ripening and senescence of strawberry (Fragaria × ananassa duch. cv. akihime) fruit and the regulation role of multiple phytohormones. *Acta Physiol. Plantarum* 36 (9), 2441–2451. doi: 10.1007/s11738-014-1617-6

Chen, S., Xiong, J., Jiao, J., Xie, Z., Huo, Z., and Hu, W. (2022). Citrus fruits maturity detection in natural environments based on convolutional neural networks and visual saliency map. *Precis. Agricul.* 23, 1515–1531. doi: 10.1007/s11119-022-09895-2

Dhakshina Kumar, S., Esakkirajan, S., Bama, S., and Keerthiveena, B. (2020). A microcontroller based machine vision approach for tomato grading and sorting using SVM classifier. *Microprocess. Microsyst.* 76, 103090. doi: 10.1016/j.micpro.2020.103090

Fan, Y., Zhang, S., Feng, K., Qian, K., Wang, Y., and Qin, S. (2022). Strawberry maturity recognition algorithm combining dark channel enhancement and YOLOv5. *Sens. (Basel)* 22 (2). doi: 10.3390/s22020419

Ge, Y. Y., Xiong, Y., and From, P. J. (2019). Instance segmentation and localization of strawberries in farm conditions for automatic fruit harvesting. *Ifac Papersonline* 52 (30), 294–299. doi: 10.1016/j.ifacol.2019.12.537

Habaragamuwa, H., Ogawa, Y., Suzuki, T., Shiigi, T., Ono, M., and Kondo, N. (2018). Detecting greenhouse strawberries (mature and immature), using deep convolutional

neural network. *Eng. Agricult. Environ. Food* 11 (3), 127–138. doi: 10.1016/j.eaef.2018.03.001

He, Y., Bose, S. K., Wang, W., Jia, X., Lu, H., and Yin, H. (2018). Pre-harvest treatment of chitosan oligosaccharides improved strawberry fruit quality. *Int. J. Mol. Sci.* 19 (8). doi: 10.3390/ijms19082194

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 2961–2969. doi: 10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778. doi: 10.1109/CVPR.2016.90

Huang, Y.-P., Wang, T.-H., and Basanta, H. (2020). Using fuzzy mask r-CNN model to automatically identify tomato ripeness. *IEEE Access* 8, 207672–207682. doi: 10.1109/access.2020.3038184

Indrabayu, I., Arifin, N., and Areni, I. S. (2019). "Strawberry ripeness classification system based on skin tone color using multi-class support vector machine," in *2019 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, pp. 191–195. doi: 10.1109/ICOIACT46704.2019.8938457

Le Louëdec, J., and Cielniak, G. (2021). 3D shape sensing and deep learning-based segmentation of strawberries. *Comput. Electron. Agric.* 190, 106374. doi: 10.1016/j.compag.2021.106374

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*), Honolulu, HI, USA, pp. 936–944. doi: 10.1109/CVPR.2017.106

Liu, J.-J., Hou, Q., Cheng, M.-M., Wang, C., and Feng, J. (2020). "Improving convolutional networks with self-calibrated convolutions," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 10093–10102. doi: 10.1109/CVPR42600.2020.01011

McInnes, L., Healy, J., and Melville, J. (2018). Umap: uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*. doi: 10.48550/arXiv.1802.03426

Moghimi, A., Aghkhani, M. H., Sazgarnia, A., and Sarmad, M. (2010). Vis/NIR spectroscopy and chemometrics for the prediction of soluble solids content and acidity (pH) of kiwifruit. *Biosyst. Eng.* 106 (3), 295–302. doi: 10.1016/j.biosystemseng.2010.04.002

Pérez-Borrero, I., Marín-Santos, D., Gegúndez-Arias, M. E., and Cortés-Ancos, E. (2020). A fast and accurate deep learning method for strawberry instance segmentation. *Comput. Electron. Agric.* 178, 105736. doi: 10.1016/j.compag.2020.105736

Raj, R., Cosgun, A., and Kulić, D. (2022). Strawberry water content estimation and ripeness classification using hyperspectral sensing. *Agronomy* 12 (2), 425. doi: 10.3390/agronomy12020425

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern. Anal. Mach. Intell.* 39 (6), 1137–1149. doi: 10.1109/tpami.2016.2577031

Saranya, N., Srinivasan, K., and Kumar, S. K. P. (2021). Banana ripeness stage identification: a deep learning approach. *J. Ambient Intell. Human. Comput.* 13 (8), 4033–4039. doi: 10.1007/s12652-021-03267-w

Shao, Y., Wang, Y., Xuan, G., Gao, Z., Hu, Z., Gao, C., et al. (2020). Assessment of strawberry ripeness using hyperspectral imaging. *Anal. Lett.* 54 (10), 1547–1560. doi: 10.1080/00032719.2020.1812622

Su, Z., Zhang, C., Yan, T., Zhu, J., Zeng, Y., Lu, X., et al. (2021). Application of hyperspectral imaging for maturity and soluble solids content determination of strawberry with deep learning approaches. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.736334

Tu, S., Xue, Y., Zheng, C., Qi, Y., Wan, H., and Mao, L. (2018). Detection of passion fruits and maturity classification using red-Green-Blue depth images. *Biosyst. Eng.* 175, 156–167. doi: 10.1016/j.biosystemseng.2018.09.004

Van de Poel, B., Vandendriessche, T., Hertog, M.L.A.T.M., Nicolai, B. M., and Geeraerd, A. (2014). Detached ripening of non-climacteric strawberry impairs aroma profile and fruit quality. *Postharvest Biol. Technol.* 95, 70–80. doi: 10.1016/j.postharvbio.2014.04.012

Yu, Y., Zhang, K., Yang, L., and Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN. *Comput. Electron. Agric.* 163, 104846. doi: 10.1016/j.compag.2019.06.001

Zhang, C., Guo, C., Liu, F., Kong, W., He, Y., and Lou, B. (2016). Hyperspectral imaging analysis for ripeness evaluation of strawberry with support vector machine. *J. Food Eng.* 179, 11–18. doi: 10.1016/j.jfoodeng.2016.01.002

Zhang, J., Wang, X., Yu, O., Tang, J., Gu, X., Wan, X., et al. (2011). Metabolic profiling of strawberry (Fragaria× ananassa duch.) during fruit development and maturation. *J. Exp. Bot.* 62 (3), 1103–1118. doi: 10.1093/jxb/erq343

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 10 October 2023
DOI 10.3389/fpls.2023.1241576

OPEN ACCESS

EDITED BY
Liangliang Yang,
Kitami Institute of Technology, Japan

REVIEWED BY
Yunchao Tang,
Guangxi University, China
Ebenezer Olaniyi,
Mississippi State University, United States

*CORRESPONDENCE
Miguel Molina-Rotger
miguel.molina@uib.es
Bartomeu Alorda-Ladaria
tomeu.alorda@uib.es

RECEIVED 16 June 2023
ACCEPTED 18 September 2023
PUBLISHED 10 October 2023

CITATION
Molina-Rotger M, Morán A, Miranda MA and Alorda-Ladaria B (2023) Remote fruit fly detection using computer vision and machine learning-based electronic trap. *Front. Plant Sci.* 14:1241576. doi: 10.3389/fpls.2023.1241576

Remote fruit fly detection using computer vision and machine learning-based electronic trap

Miguel Molina-Rotger[1*], Alejandro Morán[1], Miguel Angel Miranda[2,3] and Bartomeu Alorda-Ladaria[1,3,4*]

[1]Industrial Engineering and Construction Department, University of the Balearic Islands, Palma, Spain, [2]Biology Department, University of the Balearic Islands, Palma, Spain, [3]Institute for Environmental Agro-Environmental Research and Water Economics, University of the Balearic Islands, Palma, Spain, [4]Health Science and Technology Cross-cutting Department, Balearic Islands Health Research Institute (IdISBa), Palma, Spain

Introduction: Intelligent monitoring systems must be put in place to practice precision agriculture. In this context, computer vision and artificial intelligence techniques can be applied to monitor and prevent pests, such as that of the olive fly. These techniques are a tool to discover patterns and abnormalities in the data, which helps the early detection of pests and the prompt administration of corrective measures. However, there are significant challenges due to the lack of data to apply state of the art Deep Learning techniques.

Methods: This article examines the detection and classification of the olive fly using the Random Forest and Support Vector Machine algorithms, as well as their application in an electronic trap version based on a Raspberry Pi B+ board.

Results: The combination of the two methods is suggested to increase the accuracy of the classification results while working with a small training data set. Combining both techniques for olive fly detection yields an accuracy of 89.1%, which increases to 94.5% for SVM and 91.9% for RF when comparing all fly species to other insects.

Discussion: This research results reports a successful implementation of ML in an electronic trap system for olive fly detection, providing valuable insights and benefits. The opportunities of using small IoT devices for image classification opens new possibilities, emphasizing the significance of ML in optimizing resource usage and enhancing privacy protection. As the system grows by increasing the number of electronic traps, more data will be available. Therefore, it holds the potential to further enhance accuracy by learning from multiple trap systems, making it a promising tool for effective and sustainable fly population management.

KEYWORDS

precision agriculture, olive fruit fly pest, machine learning, support vector machine, random forest, computer vision, edge computing, remote sensing

1 Introduction

Precision Agriculture for pest management requires constant monitoring of the target pest population as well as continuous evaluation of environmental conditions like temperature and humidity. Bactrocera oleae (Gmelin), known as the olive fruit fly, is a serious pest in the olive industry. If environmental conditions favour the proliferation of this tephritidae, losses from this pest might exceed 100% of productivity in a year. As a result, developing a system capable of collecting field data is critical for precise pest management.

The traditional monitoring system is based on flytraps. Those traps kill specific species of fruit flies, which are then manually collected and identified. The number of flies trapped are checked manually usually every week during the fruit fly season and then fortnightly during the winter months. The number of hours spent in this check task is huge and due to the manually data collection frequency, the time to detect an infestation is too large for flash responses. Therefore, developing a monitoring station to automate this manual trap checking will produce many benefits Martins et al. (2019). In addition, several environmental and public health problems appear when insecticides and off-target sprays are used extensively without adequate management. Weather parameters like air temperature and humidity levels in the spraying area are critical to determine the moment to spray and the duration of this process. The adult fly population is the insecticide target, and the weather conditions are important to decrease or increase the spray process effectiveness. In this sense, automatically monitoring those parameters in real time using computer-based platforms is important to adjust the spray activity.

In general, agricultural scenarios seem to be one of the most promising application areas for wireless monitoring station deployments due to the necessity of improving the agro-food production chain in terms of precision and quality. This involves a careful system design, since a rural scenario consists of an extensive area devoid of an electrical power supply and available wired connections. Automatic monitoring stations technology is introduced in Precision Agriculture strategy (PA) to obtain accurate real time field information and make accurate and optimum decisions Bjerge et al. (2023); Fasih et al. (2023).

Plant pest control remains one of the main research objectives of modern agriculture Shah and Wu (2019). The widespread use of insecticides at the field level is still the most common practice for the control of plant pests in general and for the fruit flies in particular Dias et al. (2018). However, its use is being restricted by official authorities due to its impact on the environment, human health, and the development of resistance in target pests. The use of PA for pest control has been applied to improve the control and/or detection of several pests, as examples: particularly sensitive maps are used to drive variable insecticide application for the control of certain insect pests Reay-Jones et al. (2019); hyperspectral imaging is used to detect fruit fly infestation in fruits Ding et al. (2021); or GIS technologies are used to implement user support systems to take more precise decisions about treatments of insect pests in the Mediterranean areas Goldshtein et al. (2021). In all these cases, a continuum of more accurate monitoring data produces a more accurate assessment of pest presence which, together with geolocation information, improves understanding of the spatial and temporal distribution of pest effects. In fact, the fast access to the information about pests is mandatory to accurately manage pests and diseases in agriculture Grasswitz (2019).

Since the monitoring of fruit flies is dependent on fly identification, the first fruit fly identification platform was proposed by Pontikakos et al. (2012) as a combination of traditional manual inspection process and the computer-based platform for storing the trap checking results. The proposed computer-based platform can perform olive fruit fly evolution analysis and treatment prediction considering weather conditions. Although the manual trap inspection is also required, the automatic analysis of data combined with weather conditions allows determining the best period to apply the spray treatment and the areas to be considered in the treatment.

The second one is related to solve the identification process and reduce the time needed to check the fly traps. The authors in Bjerge et al. (2023); Fasih et al. (2023) describe a procedure to identify the fruit fly using image segmentation techniques using a camera as a sensor and some computing process to obtain the identification results. Although, the procedure is proposed using a MacPhil trap. In a MacPhil trap, the fly can be over or in the liquid introducing some additional difficulty for accurate fly identification process in comparison with using sticky traps. The sticky trap retains the flies on the surface of the trap plane and increases the possibilities to take an adequate photograph for identification purposes.

This is where computer vision and Artificial Intelligence (AI) come in. It can analyze the photo and identify the olive fly, reducing the time it takes to check the traps and automating the process. As a result, the farmer's workload is reduced. Advances in image identification techniques have paved the way for the use of AI in this field. Although Deep Learning (DL) is the most commonly used technique, Krizhevsky et al. (2017), and there are examples of their effectiveness, Victoriano et al. (2023); Uzun (2023), this article discusses classical machine learning (ML) approaches. This is because DL requires a large dataset to achieve good results, and such a dataset is currently unavailable. It is also computationally expensive. Therefore, the study will focus on the ML algorithms Random Forests (RF) and Support Vector Machines (SVM).

This work shows the design and implementation of a real time automated low-cost olive fruit fly smart trap, will now be referred to as e-trap throughout this article. The main novelty is the use of ML for image identification, in addition to the connection through a GPRS link with a cloud-based platform described in Miranda et al. (2019). In particular, it is explored how RF and SVM can improve efforts to reduce the use of pesticides against the olive fly to prevent crop loss and monitor it remotely.

2 Materials and methods

The smart trap approach consists of a photographic camera for image capture, a linux-based electronic system to implement the algorithms to recognize olive fly adults, a solar-based power system, and an ambient relative humidity/temperature sensor. The sensor

and picture data collected by the smart trap is processed and stored allowing *in-situ* access in case of communication lost.

The solar panel and the Stevenson screen for the humidity and temperature sensors are at the upper part of a metal pole see Figure 1A. The battery, transmitter system, and controller are included in a box just in the middle part, as Figure 1A shows. The controller system and the transmitter module are in the middle box for weather condition protection. In addition, Figure 1B shows the sticky trap supported by a metal pole, including a junction box with a camera installed in front of the trap. This camera is connected to the controller system for image capture and power supply. Finally, the lower part of the metal pole will be used to nail the pole on Earth and thus have a first fastening point to finish tying the pole to the strongest olive branches. In this way, the metal pole will be stable and tied up during the measurement period without disturbing the agricultural machines and workers between olive trees.

2.1 Sensors and camera

The designed prototype includes a temperature, a relative humidity sensor, and a camera serial interface (CSI). The two sensors (model DHT22) installed in the upper part of the pole will be connected and powered from the controller box. This sensor has enough resolution in both parameters, see Table 1. The DHT22 device provides a new value each 2 seconds with reduced energy consumption ratio. The controller system is designed to measure and save in local storage memory the temperature and humidity values each minute. But, only the maximum, minimum and the average values are transmitted to the cloud server every hour including the exact timestamp. This methodology reduces the amount of data to be sent to the server and filters the unwanted values (aberrant values or errors in communication with the sensor), storing the information on the station for post-analysis and maintenance purposes.

The camera used is a CMOS sensor Omnivision 5647 with removed IR filter (see Table 1 for camera specifications). It is connected and powered by the controller system using a CSI bus. The cable between camera and controller is 1.5 meters long, allowing to determine the most adequate position of sticky trap without restrictions of distances, see metal arm where sticky trap and camera are fixed in Figure 1B.

The camera is the most energy demanding device in the proposed e-trap system apart from the 4G modem. Therefore, it is powered on during the instant to take the photo, afterwards, it remains turned off. The instant when taking the photography can be adjusted considering the sun position and the amount of light available. The smart trap has been programmed by default, to take three photos when the sun is around the upper level, so the sunlight intensity will be the highest producing the highest image contrast. The three photos will be taken around midday hour with a delay of 30 minutes between each photo. In addition, users can change the timing of the photo at any time to capture the best quality photo depending on the locations and shadows on the sticky trap surface.

Photographies are taken only three times a day because this is not a real-time application. Here the goal is to infer and report the insect population without being on the field. In addition, since the system is not perfect, it is convenient to take several photographies, three in our case, to filter errors and increase the amount of training data.

FIGURE 1
Electronic components of the e-trap. **(A)** E-trap with solar panel, Stevenson screen to protect the temperature and relative humidity sensor, battery and electronics. **(B)** Camera placed in front of a Rimi® trap. **(C)** Battery and electronics.

TABLE 1 Specifications of sensor and camera elements.

Parameter	Value
Sensor voltage supply	3.3 Vdc ≤ Vcc ≤ 6 Vdc
Sensor output type	Digital
Temperature range	-40°C to 80°C
Temperature accuracy	± 0.5°C
Temperature resolution	0.1°C
Humidity range	0% to 100% RH
Humidity accuracy	2% RH
Humidity resolution	0.1% RH
Sensor measurement period	2 s
Camera resolution	2592 x 1944 pixels
Camera focus	Fixed focus
Camera dimensions	25 x 20 x 9 mm

2.2 Controller and communication system

The controller system is one of the most important parts of the smart trap. It manages sensor, camera, data transmission and performs the fly identification task. All these tasks require enough computer resources, low energy consumption and system flexibility. In this work a Raspberry Pi B+ is selected to supply the required hardware requirements in combination with the Raspbian OS lite version. The selected platform is flexible enough to manage all the tasks reducing the number of active processes and power consumption, while image processing software can be implemented using open-source resources like OpenCV, Bradski (2000).

The communication module consists of an Airlink GL8200 modem connected to the controller system using the serial port interface (SPI). The communication uses flux control to obtain maximum transmission velocity ratios (115200 bps). The modem module is compatible with standard AT commands and can allow server connections using standard internet protocols like File Transfer Protocol (SFTP), Hypertext Transfer Protocol (http) and Network Time Protocol (NTP) between others. The NTP protocol is used to maintain and update the local real time controller (RTC) enabling a time-based schedule of the tasks. The http protocol enables the connection with the remote server to store the sensor data and the fly count result on the remote database. In case of necessary, the SFTP protocol allows uploading images to the server for validation purposes with the penalty to increase the energy consumption available at the smart trap. In any case, a SD storage disk is used to save all sensor data, fly count and images. Therefore, the data will remain in the smart trap in case communication fails and can be accessed manually visiting the trap location during sticky trap maintenance.

2.3 E-trap firmware

The e-trap controller is designed using the Raspian Lite operating system implementing a time-scheduled management. The different e-trap tasks are executed using the Cron task manager embedded in the Unix systems. In this way, the e-trap is configured to work alone without expecting interaction from remote infrastructures.

The e-trap firmware is divided into five main tasks as shown in the functional diagram in Figure 2. All tasks are lunched using the Cron manager, Kernighan and Pike (1984). The first task, called "system", maintain the controller date and time updated, check the battery level, peripheral power supply management and rebooting-based strategy to avoid software issues. The second task, referred to as "collect", is related to sensor access, and collects temperature and humidity values from DHT22 sensor storing it timestamped in a local file using CSV format. The third task, named "capture", takes a picture adjusting the exposition time and white balance level to optimize the resolution and the quality of the picture. The fourth task, termed "identify", analyzes the obtained images, and try to identify the number of flies trapped. This identification process is explained in the next section. And finally, the fifth task, called "transfer", is responsible to establish LTE communications, to send the sensor data file to the remote server and to attend to the remote requirements (send the picture file or software update).

Each task of the e-trap software is launched by Cron daemon at different time during the day. Therefore, each task is implemented independently of the other tasks avoiding that one task stop the rest of tasks. In fact, meanwhile the Cron daemon is running, the tasks are initiated and terminated without interaction between them.

It is important to note that the "system" task is executed twice a day. The first time it reboots the controller to get a fresh system after one day of continuous operation. The second execution of the "system" task (@16:00) will shut down all peripherals not related to

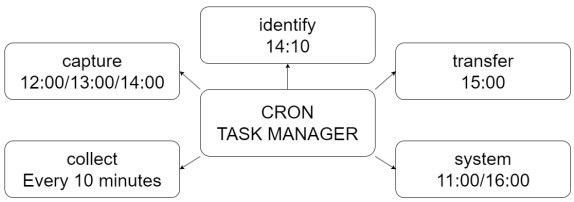


FIGURE 2
E-trap firmware flowchart showing the five main tasks and their execution times.

the collection task. With this procedure, the power consumption of Raspberry Pi platform is minimized until the next day's reboot.

3 Data collection and generation

3.1 Dataset collection

This article uses images of the two e-traps identified as N10 and N17. These traps were placed in the olive fields of the "Institut de Recerca i Formació Agroalimentària i Pesquera de les Illes Balears" (IRFAP) in Mallorca, Spain. The OV5647 camera, which is already integrated in the e-trap itself, was used to capture the images. The resulting images have a resolution of 1600 pixels wide by 1200 pixels high, 3 RGB channels, 24-bit depth, and were saved in.jpg format. Note that the physical position of the traps in the olive trees was similar but not exactly the same, resulting in differences in the final image. The dataset consists of a total of 62 images, 45 generated by N10 and 15 generated by N17. Figure 3 shows an example of an image taken by each of the traps and Table 2 shows all this data summarized.

By taking a photo every day until the sticky pad is replaced, the observation reveals the emergence of new flies alongside the already trapped flies that persist over time. Figure 4 shows how this allows us to know how the same olive fly is observed with different lighting, thus performing the data augmentation (DA) technique in an organic way and allowing the classifier model to learn which features have the highest priority in defining the fly for its correct classification. The application of this technique is common in the AI world, since it allows to face the problem of lack of data to train, and in the PA world it is no exception (Brilhador et al. (2019); Fawakherji et al. (2020)); (Shorten and Khoshgoftaar (2019)).

3.2 Dataset generation

The 45 images from N10 were used to train the classifier models. Classifier test was performed on the remaining 15 images from N17. This was advantageous because the classifier model never knew the training data and could even be given different e-trap positions and luminance conditions with respect to N10. In summary, it was possible to test whether a single e-trap could be used to generate a first scalable smart trap system capable of localizing and classifying the olive fruit fly.

After studying all the available images to train the classifier model, the dataset consisted of 501 olive flies, 368 flies of other species or very similar insects, and 611 different elements such as the bag or tube with the olive fly attractant, the brand of the adhesive panel, holes in the panel, other insects, shadows due to different lighting, trap identifier, etc., all of $32 \times 32 \times 3$ pixels. All of these were grouped into two groups, "olive fly"/"others", resulting in a data set with a ratio of 501 "olive fly" and 979 "others" samples. All these dataset values are summarized in Table 3.

A 9:1 ratio was used for training and validation of the models, i.e. 90% of the samples are used for training and 10% for validation. In addition, in order to have more working data, basic DA techniques that could be present in the nature of the project were applied: vertical image flipping, horizontal flipping, 90° rotation, and changes in the brightness and contrast of the images. These actions allowed us to enlarge each image up to $2^4 = 16$ new alternatives. In addition, it is worth highlighting these DA techniques based on basic image manipulations are considered "safe" for this application because the label is always preserved Shorten and Khoshgoftaar (2019).

Two conditions were set for this DA process: first, between zero and ten new images could be generated, this number being random for each sample. Second, each DA technique could occur with a 50% chance. In this way, the augmentation would not be homogeneous, thus preventing the model from learning repetitive patterns. This action eventually increased the training data set from 1332 to 8069 samples, and all AI models used it, so that the result comparisons for different models are not biased by the dataset.

Finally, the test images from N17 were simply labeled to match the image provided by the e-trap to simulate the real system process.

4 Machine learning classification models

As mentioned earlier, due to the size of the dataset, the final algorithms selected for this article were RF and SVM. These ML methods and their validation would be the focus of this section.

FIGURE 3
Example targets from N10 and N17 sticky traps.

TABLE 2 Dataset collection parameters.

Parameter	Value
e-traps count	2 (N15 & N17)
Sticky trap images count	45/15 (N10/N17)
Location	IRFAP, Mallorca, Spain
Resolution	1600 × 1200 × 3
Depth	24-bit
Format	.jpg

4.1 Random forest

Random Forest, introduced by Breiman (2001), is a supervised learning algorithm used for both classification and regression tasks. It is an ensemble method that combines multiple decision trees to make predictions. Each decision tree in the RF is built independently on a different subset of the training data, and the final prediction is made by aggregating the predictions of all the trees.

Here's how RF works:

1. Data Preparation Given a collection of training examples denoted as $[(x_i, y_i)]_{i=1}^{n}$, where x_i represents the input features and y_i represents the corresponding target labels, RF starts by randomly selecting subsets of the training data with replacement. These subsets are known as bootstrap samples.
2. Building Decision Trees: For each bootstrap sample, a decision tree is constructed independently. At each node of the decision tree, a feature subset is randomly selected, and the split that optimally separates the data based on some criterion (e.g., Gini impurity or entropy for classification, Jost (2006), mean squared error for regression, Langs et al. (2011)) is chosen. The tree continues to split the data until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples required to split further.
3. Ensemble Prediction: Once all the decision trees are built, predictions are made by each tree on unseen data. For classification tasks, the class with the majority of votes among the trees is selected as the final prediction. For regression tasks, the average of the predicted values from all the trees is taken.

RF offers several advantages over individual decision trees:

- Ensemble Effect: By aggregating predictions from multiple decision trees, RF reduces the risk of overfitting and provides more robust predictions.
- Feature Randomness: Randomly selecting a subset of features at each node helps to decorrelate the trees and capture different aspects of the data.
- Out-of-Bag Evaluation: As the trees are built on bootstrap samples, the instances left out in each sample (out-of-bag instances) can be used for validation without the need for an additional holdout set.

In summary, RF is a versatile and powerful algorithm that combines the predictions of multiple decision trees to achieve high accuracy and robustness in both classification and regression tasks. It is particularly effective when dealing with complex data and can handle a large number of features.

4.2 Support vector machines

Support vector machines (SVM), introduced by Vapnik and Chervonenkis (2015), are also supervised learning models used for classification and regression analysis. The term SVM typically does not refer to a linear SVM, but rather to the use of kernel methods, Sánchez A (2003).

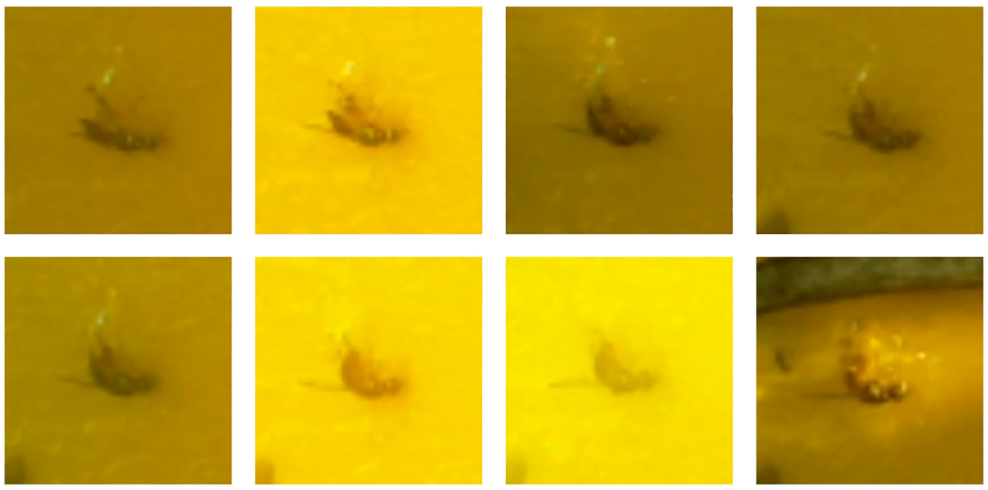


FIGURE 4
Example of the same olive fruit fly from 8 to 15 October on N17.

TABLE 3 Training, validation, and test set sizes for the cropped images. Note that the training size refers to the already augmented data and the percentages refer to the sum of these augmented samples.

Parameter	Total value	Train value (90%)	Validation value (10%)	Test value
Olive flies	501	451	50	6
Other species flies	368	332	36	17
Other elements	611	550	61	14

Given a collection of training examples denoted as $[(x_i, y_i)]_{i=1}^n$, and a kernel function denoted as K, each y_i belonging to the set [−1, +1] represents its categorization into one of two categories. An objective function of the SVM is used to solve the optimization problem defined as follows:

$$\max_{\alpha}\left[\sum_{i=1}^{n}\alpha_i + \sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j)\right] \quad (1)$$

subject to the constraints:

$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Here, the Lagrange coefficients α_i are involved, and the constant C is used to penalize training errors present in the samples.

An SVM training algorithm constructs a model that classifies new examples into one of two categories, acting as a non-probabilistic binary linear classifier. The SVM model represents the examples as points in a space in which they are mapped to ensure a clear gap that maximizes its width between the different categories. Then, new examples are projected into the same space and their categorization is predicted based on which side of the gap they fall. As mentioned in the introduction, the choice of the regularization parameters $\alpha_i$and the form of the kernel function $K(x_i, x_j)$ have a significant impact on the performance of the SVM. These factors are thoroughly considered and extensively discussed in the comparative experiments.

4.3 Model validation

When building a model, there are several parameters to consider, and depending on how they are combined, the results may vary. In addition, there is a stochastic variable in the selection of data that may or may not favor the final result.

Therefore, the techniques used in this article can be grouped into two. (i) Grid search, to find the combination of hyperparameters that give the best results. (ii) Cross validation, to perform the process k times with different combinations of data, thus validating that the response of the classifier model is general and not specific to a single combination of data.

The metrics used for validation were: confusion matrix, accuracy, precision, recall, f1-score, Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC).

4.3.1 Confusion matrix

Measures the performance of a classification model by summarizing the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions in tabular form.

4.3.2 Accuracy

This metric measures the proportion of correctly classified images out of the total number of images in the dataset.

$$Accuracy = ((TP + TN)/TotalImages) * 10$$

4.3.3 Precision

It measures the proportion of correctly predicted positive instances out of all instances predicted as positive.

$$Precision = TP/(TP + FP)$$

4.3.4 Recall

The recall metric measures the ability of a model to correctly identify positive instances out of all the instances that are actually positive.

$$Recall = TP/(TP + FN)$$

4.3.5 F1-Score

The F1 score is a metric that combines precision and recall to provide a single measure of a model's performance in classification tasks, including image classification. It takes into account both the false positives and false negatives to assess the balance between precision and recall. The F1 score is calculated by

$$F1score = 2 * (Precision * Recall)/(Precision + Recall)\,.$$

4.3.6 ROC curve

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR represents the recall or sensitivity (correctly predicted positive instances), while the FPR represents the proportion of negative instances incorrectly classified as positive.

4.3.7 AUC

The AUC measures the performance of a model in terms of its ability to discriminate between positive and negative instances across different classification thresholds.

5 Image approach: fruit fly detection

Identifying the olive fruit fly in the e-trap images involved a number of challenges. The first was the lack of images available to train and validate the AI model. The second was the ability to distinguish the olive fruit fly from other fly families or dark elements that might appear in the images. Finally, the third was related to the processing power and energy consumption allocated for inference, in this case the target device was a Raspberry Pi B+.

The usual way to perform this process of object detection on an image is usually done by applying convolutional neural networks (CNNs). An example of this is the recent publication by Jia et al. (2023), where they apply the YOLOX-m network for the localization of different green fruits, such as green apple and green persimmon, among the leaves of trees, which can also be green. Other examples include the recognition and counting of bananas by Wu et al. (2021, 2023). The reason for applying this technique is mainly due to its ability to extract physical and temporal features from the images. However, in this paper, the CNNs path is discarded because the challenges mentioned in the previous section become clearly latent. State of the art CNNs require large datasets to train the model, which has not been available so far, and the computational process is expensive for some devices such as a Raspberry Pi B+ without external aids like a hardware accelerator.

The working dataset is considered small compared to the usual benchmarks for these tasks. For example, MNIST with 60,000 training images, CIFAR-10 and CIFAR-100 with 50,000 images each or Imagenet with 1.2 million training images (LeCun et al. (1998); Krizhevsky and Hinton (2009); Deng et al. (2009)).

Due to this challenge, in this article it was decided to finally apply classification methods based on traditional ML techniques. Although such models are mainly used for tabular data, present less overfitting when working with small amounts of data. In addition, since the model complexity is usually lower, in general, power consumption is lower too. Table 4 shows the different models tested in a first step. It is observed that for the same set of training data and all the metrics of the ML models are clearly superior to those of the DL models. Therefore, it was decided to investigate the different ML models in more detail.

The use of ML techniques for image processing is not new, Wang et al. (2021) concluded that traditional ML has a better solution effect on small sample data sets. Researchers such as Mekha and Teeyasuksaet (2021) have already studied the use of different ML algorithms for the detection of diseases in rice leaves, concluding that the application of RF was the one that gave them the best results. Another example are Liu et al. (2017), which proposes the use of the SVM algorithm for image classification in remote locations, as in our case, instead of using DL.

Performing fly detection with traditional ML methods was a new challenge. Some pre-trained DL models for object detection already have this built-in function, capable of locating and classifying objects, as well as understanding the overlap between different possible locations of the same object (Milioto et al. (2018); Prasetyo et al. (2020); Rong et al. (2022)). In our case, the solution was to first apply image processing that takes advantage of the contrast between the yellow background of the trap and the dark color of the fly to distinguish where the different elements to be classified appear. Finally, all that remained was the ML classification process for each of the elements found.

Since RF and SVM gave the top-2 better performance metrics compared to other models, it was decided to combine them to improve classification performance. Therefore, it is validated that the element is an olive fruit fly if both models assert that the element is an olive fruit fly.

Figure 5 shows the logical flow. First, the image is captured. Second, the image is processed by segmenting the trap to avoid possible false positives and locating the elements that appear in the e-trap. Third, each element is classified one by one by applying RF and SVM. Finally, if both validate the classification, it is marked on the image.

6 Results

This section presents the results of the study. In the previous points, it was mentioned that CNNs are not able to provide accurate results due to the small training dataset. Therefore, classical ML solutions are compared with CNNs solutions.

6.1 Machine learning and deep leaning results

As mentioned above, the challenges of the project were: mainly how to deal with the limited training data available, and also whether it is possible to develop an accurate classifier model taking into account the low computational capacity of the

TABLE 4 Olive fly classification performance metrics for different traditional ML and DL approaches.

Model	Type	Accuracy	Precision	Recall	F1-score	AUC
Random Forest	ML	0.85	0.84	0.85	0.85	0.85
SVM	ML	0.81	0.80	0.80	0.80	0.80
Decision Tree	ML	0.75	0.78	0.75	0.75	0.77
VGG16	DL	0.59	0.68	0.69	0.39	0.54
MobileNet	DL	0.59	0.68	0.69	0.39	0.54
Xception	DL	0.58	0.68	0.69	0.38	0.53

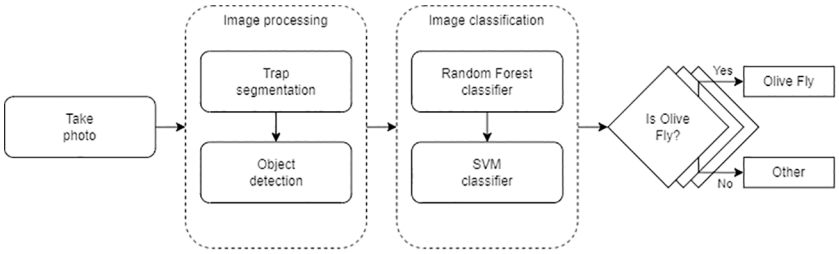


FIGURE 5
Inference pipeline.

Raspberry Pi B+. Table 1 shows the metrics of the different models proposed in the first phase of the project.

As evident from the analysis, there are six evaluated models, comprising three classical ML algorithms and three CNN models. The ML algorithms are the already mentioned RF and SVM, and also the Decision Tree algorithm, which already includes the RF, as mentioned above.

On the other hand, the CNNs include the VGG16, Mobilenet, and Xception models (Simonyan and Zisserman (2014); Howard et al. (2017); Chollet (2017)). Models that are widely used for image classification due to their good results. For example, the work of Subramanian and Sankar (2022), where they compare this CNN model and others for coconut maturity detection. Or the work of Sehree and Khidhir (2022) that classifies olive trees from unmanned aerial vehicle images.

Looking at Table 4, the superiority of the ML becomes evident, maintaining an accuracy of no less than 75%, compared to the DL, which does not achieve more than 60% accuracy in any case due to the limited availability of data.

As mentioned in section 3.2 *Dataset Generation*, the validation data come from N10, so the metrics will always tend to be higher than the test metrics, which comes from e-traps unknown to the model. Table 4 also shows the AUC value, DL models tend to be around 0.5, which could lead us to think that they are doing a random classification.

At this point, it was decided to take the two best results and test them as if the system was already in production.

6.2 Random forest and support vector machines analysis

Figure 6 shows the results of the two-week evolution of trap N17 from no flies to six flies. The Figure 6A refers to the true positives (TP), i.e. the correct classification of the olive fly by the different models. And the Figures 6B, C refer to the false classifications of the fly, the false positives (FP) refer to the elements that the model classified as flies and they are not, and the false negatives (FN) refer to the elements that are flies and the model discarded them. The final hyperparameters used in RF were: max depth of 20, min samples split equal to 5, and 3 estimators. And the final SVM hyperparameters were A polynomial kernel, C equal to 0.1, and gamma equal to 1.

6.2.1 RF classifier

This model tends to classify most items that resemble an olive fruit fly as "Olive Fly". After examining the images, one may conclude this is because the RF model is not able to differentiate whether a fly belongs to the olive fruit fly species or not, so its FP rate tends to rise and conversely the FNs are very low.

6.2.2 SVM classifier

The graphs show how this model is more cautious about RF in determining whether an object is an olive fly or not. Therefore, its FP rate is lower, but it increases the FN discriminating flies that were correct.

6.2.3 RF+SVM classifier

Finally, combining the two models allows for more accurate classification. The FNs go down even further, in exchange for the fact that if an item is claimed to be a olive fly, it is much more likely to be so.

7 Discussion

In this study, an intelligent system capable of detecting the olive fly using non-invasive techniques was developed. Two models were created with an accuracy of 62.1% for RF and 86.4% for SVM, Figure 7A, using only the data of two traps, one for training and the other to validate the models.

While RF would be the first to warn of a possible fly infestation. SVM proved to be more conservative in stating whether or not there was a fly in the sticky trap. In addition, a third option was also presented too, the combination of both models to be able to combine the best of each and achieve a higher accuracy of 89.1%, as shown in Figure 7A.

It has been shown that it is also possible to control the olive fly using classical ML techniques. Allowing deploy this intelligent systems faster than if the detection were performed using CNN techniques. And consequently understand the status of the crops before and remotely observe the evolution of the fly population, Figure 6A. In addition, the robustness achieved using ML is reflected in Figure 7B. Here, the performance of both models is shown when trying to classify only flies, regardless of the species. As can be seen, the accuracy of both algorithms increases to 91.9% for RF and 94.5% for SVM.

Therefore, this project demonstrates the application of ML on an e-trap system that facilitates the control tasks to the experts, being able to reduce the number of times they should go to the fields to make the manual count of the flies, as well as providing additional information

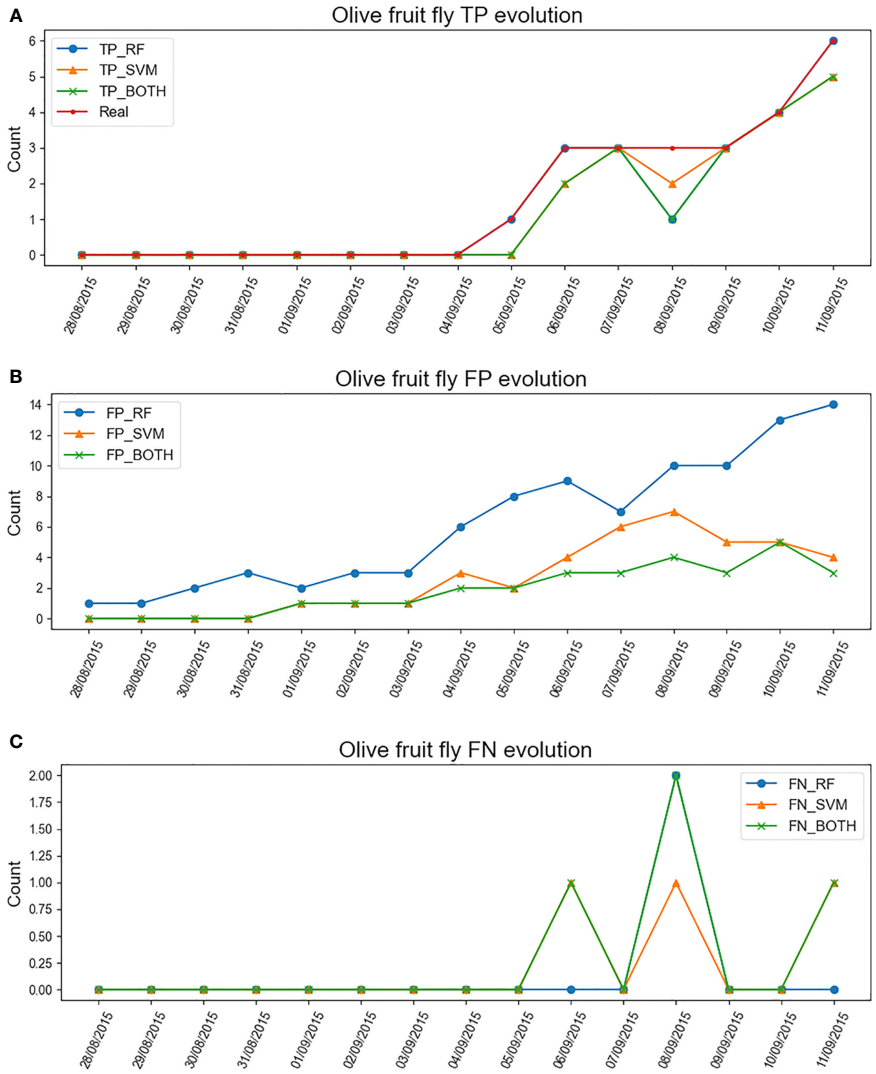


FIGURE 6
Evolution of the number of olive flies detected in the sticky trap as a function of time using different ML classifiers for the N17 e-trap. **(A)** TP evolution of RF classifier, SVM classifier and their combination together with the real count. **(B)** FP evolution of the same classifiers. **(C)** FN evolution of the same classifiers.

not to go blindly. Thus providing an improvement compared to the previous article of this same project of Miranda et al. (2019).

This opens a horizon for new challenges where, if the size of the data set and the computational capabilities of the system are not optimal, as is often the case in specific systems such as the trap described, combined ML techniques can be explored for image classification on remote devices.

In addition to the benefits described above, the application of ML strategies opens up new possibilities for the system. Once the model is trained, the device performs the prepreocessing and inference on the image data, but only the prediction is exchanged with the server. In this regard, it is also worth mentioning the advantages in terms of privacy, e.g. there is no risk related to identifying people in images sent to the server. Since no images are shared with the server, it also represents an improvement in terms of privacy. Moreover, these models are relatively small compared to state-of-the-art neural networks and might be running on small IoT devices, such as Raspberry Pi B+ used in this case, or even smaller very low power microcontroller boards. Overall, it implies a reduction in power and energy consumption and an increase in battery life. All this is possible by making a more efficient use of bandwidth.

Finally, it is important to note that the data source used has come from a single e-trap system, so the system has the potential to increase the accuracy of the results as the system of nodes grows while each e-trap system can learn specific details of the conditions that make it unique.

8 Conclusions

The main contributions of this study are threefold: development of an intelligent system for efficient crop monitoring, demonstrating superior performance of ML methods over DL for this particular case study, and further improving performance using a simple model ensembling approach.

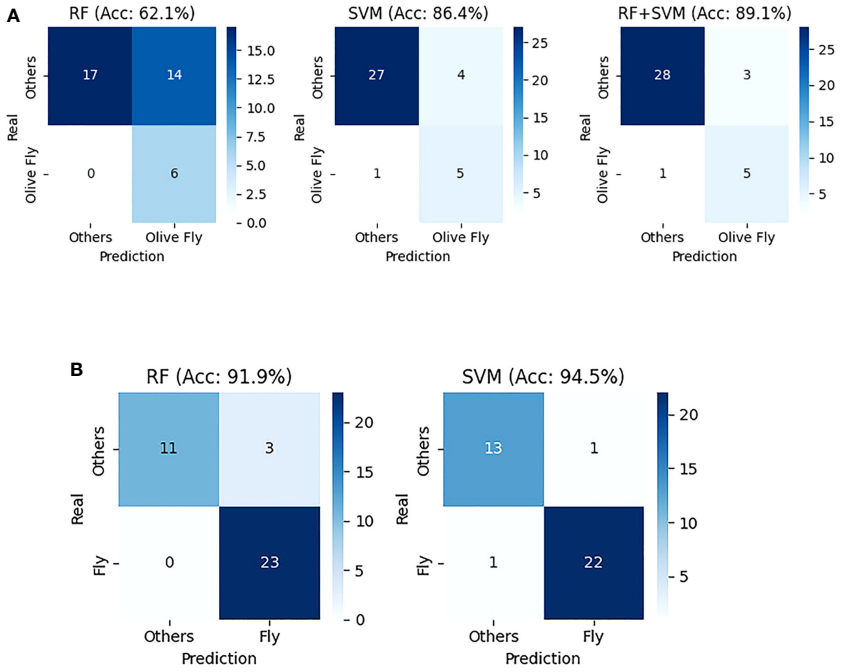


FIGURE 7
RF and SVM metrics on the N17 image from the 11th of October. **(A)** Confusion matrix comparison of RF, SVM, and "RF+SVM" models for olive fly classification. **(B)** Confusion matrix comparison of RF and SVM models for classification of all fly species.

An intelligent system capable of detecting the olive fly using non-invasive techniques was successfully developed. The system is capable of monitoring the fly and olive fly population using image processing and ML techniques. This enabled experts to remotely monitor the status and evolution of the fly population, thereby reducing the need for manual fly counts in the fields.

Since a relatively small dataset was available, the application of classical ML techniques worked better compared to a transfer learning approach using pre-trained DL models. The study revealed that classical ML models (RF and SVM) outperformed CNN solutions in this case. Despite the scarcity of images, these models demonstrated good accuracy, making them an attractive option for resource-constrained applications. In particular, the RF and SVM models reported an accuracy of 62.1% and 86.4% for the olive fly detection task, respectively. In addition, the RF and SVM approaches reported an accuracy of 91.9% and 94.5%, respectively, when classifying only flies, regardless of the species.

Finally, the model performance was further improved by combining both RF and SVM models. RF was found to be more sensitive in detecting a potential fly infestation, while SVM demonstrated a more cautious approach in stating whether a fly was present in the sticky trap. As a result, combining both models led to an increased accuracy of 89.1% for the olive fly detection task.

In conclusion, this research showcases the successful implementation of ML in an e-trap system for olive fly detection, providing valuable insights and benefits. The combination of RF and SVM models demonstrated promising results, offering more efficient crop monitoring and control tasks to the experts. The potential for using small IoT devices for image classification opens up new possibilities, emphasizing the significance of ML in optimizing resource usage and enhancing privacy protection. As the system grows by increasing the number of e-traps, more data will be available. Therefore, it holds the potential to further enhance accuracy by learning from multiple e-trap systems, making it a promising tool for effective and sustainable fly population management.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

MM-R, MM and BA-L contributed to conception of the study. MM-R and BA-L contributed to the design of the study MM-R organized the datasets and performed the experimental analysis. MM-R wrote the first draft of the manuscript. MM-R, MM, AM, and BA-L wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work has been partially sponsored and promoted by the Comunitat Autonoma de les Illes Balears through the Direcció General de Recerca, Innovació I Transformació Digital and the Conselleria de Economia, Hisenda I Innovació and by the European Union- Next Generation UE (BIO/016 A.2). Nevertheless, the views and opinions expressed are solely those of the author or authors, and do not necessarily reflect those of the European Union or the

European Commission. Neither the European Union nor the European Commission are to be held responsible.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bjerge, K., Alison, J., Dyrmann, M., Frigaard, C. E., Mann, H. M., and Høye, T. T. (2023). Accurate detection and identification of insects from camera trap images with deep learning. *PloS Sustainabil Transform* 2, e0000051. doi: 10.1371/journal.pstr.0000051

Bradski, G. (2000). The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25 (11), 120–123.

Breiman, L. (2001). Random forests. *Mach. Learn.* 25, (11), p. 120–123. doi: 10.1023/A:1010933404324

Brilhador, A., Gutoski, M., Hattori, L. T., de Souza Inacio,´, A., Lazzaretti, A. E., and Lopes, H. S. (2019). "Classification of weeds and crops at the pixel-level using convolutional neural networks and data augmentation," in *2019 IEEE latin american conference on computational intelligence (LA-CCI)* (Guayaquil, Ecuador: IEEE), 1–6.

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (Honolulu, USA: IEEE Computer Society), 1251–1258.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition* (Miami, Florida: Ieee), 248–255.

Dias, N. P., Zotti, M. J., Montoya, P., Carvalho, I. R., and Nava, D. E. (2018). Fruit fly management research: A systematic review of monitoring and control tactics in the world. *Crop Prot.* 112, 187–200. doi: 10.1016/j.cropro.2018.05.019

Ding, G., Qiao, Y., Yi, W., Fang, W., and Du, L. (2021). Fruit fly optimization algorithm based on a novel fluctuation model and its application in band selection for hyperspectral image. *J. Ambient. Intell. Humanized. Computing.* 12, 1517–1539. doi: 10.1007/s12652-020-02226-1

Fasih, S. M., Ali, A., Mabood, T., Ullah, A., Hanif, M., and Ahmad, W. (2023). "Fruit fly detection and classification in iot setup," in *International conference on computational science and its applications* (Athens, Greece: Springer), 593–607.

Fawakherji, M., Potena, C., Prevedello, I., Pretto, A., Bloisi, D. D., and Nardi, D. (2020). "Data augmentation using gans for crop/weed segmentation in precision farming," in *2020 IEEE conference on control technology and applications (CCTA)* (Sheraton Downtown, Canada: IEEE), 279–284.

Goldshtein, E., Gazit, Y., Hetzroni, A., Timar, D., Rosenfeld, L., Grinshpon, Y., et al. (2021). Long-term automatic trap data reveal factors affecting diurnal flight patterns of the mediterranean fruit fly. *J. Appl. Entomo.* 145, 427–439. doi: 10.1111/jen.12867

Grasswitz, T. (2019). Integrated pest management (ipm) for small-scale farms in developed economies: Challenges and opportunities. *insects* 10, 179. doi: 10.3390/insects10060179

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv. preprint. arXiv:1704.04861*. doi: 10.48550/arXiv.1704.04861

Jia, W., Xu, Y., Lu, Y., Yin, X., Pan, N., Jiang, R., et al. (2023). An accurate green fruits detection method based on optimized yolox-m. *Front. Plant Sci.* 14, 1411. doi: 10.3389/fpls.2023.1187734

Jost, L. (2006). Entropy and diversity. *Oikos* 113, 363–375. doi: 10.1111/j.2006.0030-1299.14714.x

Kernighan, B. W., and Pike, R. (1984). *The UNIX programming environment* Vol. 270 (Hoboken, New Jersey, U.S.: Prentice-Hall Englewood Cliffs, NJ).

Krizhevsky, A., and Hinton, G. (2009). *Learning Multiple Layers of Features from Tiny Images*. (Toronto, Ontario: University of Toronto). Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Langs, G., Menze, B. H., Lashkari, D., and Golland, P. (2011). Detecting stable distributed patterns of brain activation using gini contrast. *NeuroImage* 56, 497–507. doi: 10.1016/j.neuroimage.2010.07.074

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Liu, P., Choo, K.-K. R., Wang, L., and Huang, F. (2017). Svm or deep learning? a comparative study on remote sensing image classification. *Soft. Computing.* 21, 7053–7065. doi: 10.1007/s00500-016-2247-2

Martins, V. A., Freitas, L. C., de Aguiar, M. S., de Brisolara, L. B., and Ferreira, P. R. (2019). "Deep learning applied to the identification of fruit fly in intelligent traps," in *2019 IX Brazilian symposium on computing systems engineering (SBESC)* (Natal, Brazil: IEEE), 1–8.

Mekha, P., and Teeyasuksaet, N. (2021). "Image classification of rice leaf diseases using random forest algorithm," in *2021 joint international conference on digital arts, media and technology with ECTI northern section conference on electrical, electronics, computer and telecommunication engineering* (IEEE), 165–169.

Milioto, A., Lottes, P., and Stachniss, C. (2018). "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," in *2018 IEEE international conference on robotics and automation (ICRA)* (Brisbane, Australia: IEEE), 2229–2235.

Miranda, M. Á., Barceló, C., Valdés, F., Feliu, J. F., Nestel, D., Papadopoulos, N., et al. (2019). Developing and implementation of decision support system (dss) for the control of olive fruit fly, bactrocera oleae, in mediterranean olive orchards. *Agronomy* 9, 620. doi: 10.3390/agronomy9100620

Pontikakos, C. M., Tsiligiridis, T. A., Yialouris, C. P., and Kontodimas, D. C. (2012). Pest management control of olive fruit fly (bactrocera oleae) based on a location-aware agro-environmental system. *Comput. Electron. Agric.* 87, 39–50. doi: 10.1016/j.compag.2012.05.001

Prasetyo, E., Suciati, N., and Fatichah, C. (2020). "A comparison of yolo and mask r-cnn for segmenting head and tail of fish," in *2020 4th international conference on informatics and computational sciences (ICICoS)* (Semarang, Indonesia: IEEE), 1–6.

Reay-Jones, F. P., Greene, J. K., and Bauer, P. J. (2019). Spatial distributions of thrips (thysanoptera: Thripidae) in cotton. *J. Insect Sci.* 19, 3. doi: 10.1093/jisesa/iez103

Rong, M., Wang, Z., Ban, B., and Guo, X. (2022). Pest identification and counting of yellow plate in field based on improved mask r-cnn. *Discrete. Dynamics. Nat. Soc.* 2022, 1–9. doi: 10.1155/2022/1913577

Sánchez, A. V. D. (2003). Advanced support vector machines and kernel methods. *Neurocomputing* 55, 5–20. doi: 10.1016/S0925-2312(03)00373-4

Sehree, N. A., and Khidhir, A. M. (2022). Olive trees cases classification based on deep convolutional neural network from unmanned aerial vehicle imagery. *Indonesian. J. Electrical. Eng. Comput. Sci.* 27, 92. doi: 10.11591/ijeecs.v27.i1.pp92-101

Shah, F., and Wu, W. (2019). Soil and crop management strategies to ensure higher crop productivity within sustainable environments. *Sustainability* 11, 1485. doi: 10.3390/su11051485

Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. big. Data* 6, 1–48. doi: 10.1186/s40537-019-0197-0

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv. preprint. arXiv:1409.1556*. doi: 10.48550/arXiv.1409.1556

Subramanian, P., and Sankar, T. S. (2022). "Coconut maturity recognition using convolutional neural network," in *Computer vision and machine learning in agriculture, volume 2* (Singapore, Singapore: Springer), 107–120.

Uzun, Y. (2023). *An intelligent system for detecting mediterranean fruit fly [medfly; ceratitis capitata (wiedemann)]*. (Aksaray, Turkey: Aksaray Üniversitesi Fen Bilimleri Enstitüsü).

Vapnik, V. N., and Chervonenkis, A. Y. (2015). "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of complexity: festschrift for alexey chervonenkis* (New York, USA: Springer Publishing company), 11–30.

Victoriano, M., Oliveira, L., and Oliveira, H. P. (2023). "Automated detection and identification of olive fruit fly using yolov7 algorithm," in *Iberian conference on pattern recognition and image analysis* (Alicante, Spain: Springer), 211–222.

Wang, P., Fan, E., and Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition. Lett.* 141, 61–67. doi: 10.1016/j.patrec.2020.07.042

Wu, F., Duan, J., Chen, S., Ye, Y., Ai, P., and Yang, Z. (2021). Multi-target recognition of bananas and automatic positioning for the inflorescence axis cutting point. *Front. Plant Sci.* 12, 705021. doi: 10.3389/fpls.2021.705021

Wu, F., Yang, Z., Mo, X., Wu, Z., Tang, W., Duan, J., et al. (2023). Detection and counting of banana bunches by integrating deep learning and classic image-processing algorithms. *Comput. Electron. Agric.* 209, 107827. doi: 10.1016/j.compag.2023.107827

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 18 October 2023
DOI 10.3389/fpls.2023.1274231

OPEN ACCESS

EDITED BY
Liangliang Yang,
Kitami Institute of Technology, Japan

REVIEWED BY
Ruirui Zhang,
Beijing Academy of Agricultural and Forestry Sciences, China
Jiangtao Qi,
Jilin University, China

*CORRESPONDENCE
Heming Li
liheming@sdmu.edu.cn

RECEIVED 08 August 2023
ACCEPTED 19 September 2023
PUBLISHED 18 October 2023

CITATION
Cheng H and Li H (2023) Identification of apple leaf disease via novel attention mechanism based convolutional neural network.
Front. Plant Sci. 14:1274231.
doi: 10.3389/fpls.2023.1274231

Identification of apple leaf disease via novel attention mechanism based convolutional neural network

Hebin Cheng and Heming Li*

School of Intelligence Engineering, Shandong Management University, Jinan, China

Introduction: The identification of apple leaf diseases is crucial for apple production.

Methods: To assist farmers in promptly recognizing leaf diseases in apple trees, we propose a novel attention mechanism. Building upon this mechanism and MobileNet v3, we introduce a new deep learning network.

Results and discussion: Applying this network to our carefully curated dataset, we achieved an impressive accuracy of 98.7% in identifying apple leaf diseases, surpassing similar models such as EfficientNet-B0, ResNet-34, and DenseNet-121. Furthermore, the precision, recall, and f1-score of our model also outperform these models, while maintaining the advantages of fewer parameters and less computational consumption of the MobileNet network. Therefore, our model has the potential in other similar application scenarios and has broad prospects.

KEYWORDS

apple leaf disease, classification, deep learning, attention mechanism, multi-scale feature extraction

1 Introduction

Apple is one of the most popular and widely grown fruits worldwide and has been cultivated by humans for over 2000 years. Apple fruit is rich in vitamins and minerals, with high nutritional value, and is an indispensable part of a healthy diet. However, the production of apples is also hindered by various diseases, which can seriously affect the yield and quality of apples. Traditional plant disease identification, management, and prevention rely on the experience of farmers and local agricultural technicians. When these measures are insufficient, it is impossible to accurately identify the diseases and timely intervene, causing great losses to apple production. In the past decade, with the continuous development and progress of machine learning (ML), especially the advancement of deep learning (DL) technology, the accuracy of identifying leaf diseases has been continuously

improved, paving the way for more efficient and real-time disease detection. Kamilaris and Prenafeta-Boldú (2018); Pardede et al. (2018).

The disease recognition of plant leaves is essentially an image classification problem that requires accurate capture of disease features, comparison with other types of diseases, and classification. Traditional ML methods typically use image processing and classifier for plant disease recognition. The image processing methods include extracting the color and texture of disease spots through grayscale values or performing pixel-level segmentation of disease spots. Deng et al. (2019) Support vector machine (SVM), Mokhtar et al. (2015) k-means clustering, Naive Bayes, etc. Ma et al. (2018) are most widely used classifier. Tradition ML has good recognition accuracy for diseases with certain characteristics. Singh et al. (2016) However, the generalization of these methods is poor, limited by the inability to recognition of nonlinear data and the difficulty of feature extraction. Once the processing object changes, the model cannot perform reasonable classification.

Convolutional neural network (CNN) automatically extracts features directly from the original image, greatly improving the efficiency of image classification. Therefore, with the emergence of CNN, especially the success of AlexNET in the competition of ImageNet LSVRC-2010, Krizhevsky et al. (2017); Shin et al. (2021) a series of DL models have been proposed, such as GoogleNet, Inception, VGG, ResNet, DenseNet, etc. Not surprisingly, these DL networks have also been used by researchers in plant disease detection. For example, Fuentes et al. present a deep-learning-based model to detect diseases and pests in tomato plants. They proposed a two-stage model which combines the meta-architecture (faster R-CNN) with feature extractors such as VGG and ResNet. Their system can effectively recognize nine different types of diseases and pests in complex surroundings. Fuentes et al. (2017) Khan, et al. utilized a hybrid method -a segmentation method which followed pre-trained deep models to achieve the classification accuracy of 98.60% on public datasets. Khan et al. (2018) Ferentinos compared some DL networks such as AlexNet, GoogLeNet, and VGG et al. and reported a 99.53% accuracy with VGG16 on the extended PlantVillage dataset. Ferentinos (2018) Arsenovic et al. proposed a novel two-stage architecture of a neural network which focused on a real environment plant disease classification. Their model achieved an accuracy of 93.67%. Arsenovic et al. (2019) Too, et al. compared many DL architecture and evaluated the best performance of DenseNet-121 in the experiment. Too et al. (2019). Shoaib et al. utilized the Inception Net model in the research work. For the detection and segmentation of disease-affected regions, two state-of the-art semantic segmentation models, i.e., U-Net and Modified U-Net, are utilized in their work too. Shoaib et al. (2022) At the same time, in the segmented field of apple leaf disease detection, a number of research achievements have also emerged. Hasan et al. (2022) For example Jiang et al. proposed an INAR-SSD (incorporating Inception module and Rainbow concatenation) model that achieves a detection accuracy of 78.80% mean Average Precision (mAP) on the apple leaf disease dataset, while maintaining a rapid detection speed of 23.13 frames per second (FPS) Jiang et al. (2019). Sun et al, proposed a novel MEAN-SSD (Mobile End AppleNet based SSD algorithm) model, which can achieve the detection performance of 83.12% mAP and a speed of 12.53 FPS. Sun et al. (2021).

MobileNet is a lightweight network proposed by Google and is widely used by researchers. Howard et al. (2017); Wang et al. (2021); Xiong et al. (2020) In MobileNet v1, depthwise separable convolution was first proposed, which combines depthwise convolution and pointwise convolution in the module. The computational complexity was successfully reduced to 1/9 of that of ordinary convolution. Therefore it greatly reduces computational parameters and improves the speed of model computation. Sandler et al. (2018) In MobileNet v2, the interest manifold is captured by inserting a linear bottleneck in the convolution module instead of the original nonlinear activation function. Kavyashree and El-Sharkawy (2021) The researchers also proposed the inverted residual structure, which expands dimensions through an expansion layer. The depthwise separable convolutions are used to extract features, and projection layers are used to compress data, making the network smaller again. Through this structure, the dimensionality and computational speed of convolutions are balanced, enhancing the performance of the network. In MobileNet v3, the Squeeze-and-Excitation (SE) attention mechanism is further introduced. The SE module is added to the inverted residual structure, and the activation function is updated. Howard et al. (2019) Compared to MobileNet v2, the computational speed and accuracy of MobileNet v3 have been further improved.

In recent years, more Transfer learning (TL) strategies are used in DL. Chen et al. (2020); Coulibaly et al. (2019) These DL models require a large amount of labeled data to achieve good performance. However, in many real-world scenarios, obtaining such a large amount of labeled data may be expensive, time-consuming, or impractical. TL enables the utilization of pre-existing large-scale datasets, such as ImageNet or COCO data sets, and transfers the knowledge obtained from them to the target tasks. On the other hand, DL models consist of multiple layers that learn the hierarchical representation of data. Early layers capture general low-level features (such as edges and textures), while later layers capture high-level semantic features. By using TL, we can reuse low-level and intermediate features learned from pre-trained models as feature extractors. This reduces the need to train these layers from scratch and allows us to focus on training only the top layers specific to our tasks. In the training process of our model, we also adopted the method of TL and achieved very good results.

In this article, we propose a deep learning model named mobileNet-MFS, where MFS is the abbreviation for multi-fused spatial. The main contributions of our work include:

1. A novel fused spatial channel attention (FSCA) mechanism is proposed, which considers both channel and spatial connections of the input layer. We use it to replace the Squeeze-and-Excitation(SE) attention mechanism in the MobileNet v3 architecture and significantly improve the performance of the model.
2. In order to include multi-dimensional information in neural networks, a multi-scale feature extraction module was applied in our network, which fused image features

through convolutions of different dimensions. Research has shown that this module has successfully improved the model's accuracy.

3. Our proposed MobileNet-MFS model has better performance than the original version of MobileNet v3, demonstrating advantages in accuracy, computational speed, parameter size, and other aspects compared to MobileNet VIT, EffientNet, ShuffleNet, DenseNet in diagnosing apple leaf diseases.

2 Methodology

2.1 Network architechture

The network architecture of our model(MobileNet-MFS) is shown in Figure 1A. The design of the model inherits the main modules of MobileNet v3, but in order to obtain better diagnostic efficacy, many modifications were also made to the model. The main body of the model is consistent with MobileNet v3, which consists of a two-dimensional convolutional layer, a series of bottleneck layers with different dimensions, a two-dimensional convolutional layer, a pooling layer, and a one-dimensional convolutional layer in sequence. Through this series of modules, feature information on plant disease-affected areas is extracted, and diseases are classified into 9 types through 1×1 convolution. However, at the front end of the model, in order to further explore the feature information that cannot be captured in the original MobileNet v3, we introduced a multi-scale feature extraction module. The most important change is that we have proposed a new FSCA attention mechanism to replace the SE attention mechanism module used in MobileNeT v3. The FSCA mechanism will be explained in detail in the following chapters.

As shown in Figure 1B, in MobileNet-MFS, the most basic module is the bottleneck layer, which is composed of an inverted residual network containing depthwise separated convolutions. It replaces the standard convolution operation with a depthwise convolution followed by a pointwise convolution. This reduces computational complexity and model size while maintaining accuracy. In addition to depthwise separated convolution, the bottleneck layer also includes expansion convolution, which mainly serves to increase the number of channels in the input feature map using a 1x1-sized convolutional kernel. Projection convolution is a 1x1 convolutional kernel with a significantly smaller number of output channels than the input channels, thus achieving the goal of limiting the size of the model. When the input and output channels are the same, a residual network can be used. The bottleneck layer of the inverted residual structure formed by the above convolution operations is finally activated using ReLU or h-swish function.

2.2 Attention module

Although CNN is very powerful in image expression, they are deficient in expressing spatial information. Therefore, the attention mechanism has been introduced into MobileNet v3, which can improve the learning ability of the model by assigning weights to images. In the original version of MobileNet v3, the SE attention module is placed in the middle of the bottleneck layer, Hu et al. (2020) giving an updated set of weight values through two fully connection layers and the activation function. However, the SE attention module only cares about the dependencies between channels and ignores location information, which is crucial for generating spatially selective attention maps. Therefore, we propose our FSCA attention mechanism to replace the SE module.

The FSCA attention mechanism considers both spatial and channel information of the input layer, thus more effectively guiding the model to focus on effective positions in the image. As shown in Figure 2, the FSCA attention mechanism consists of two

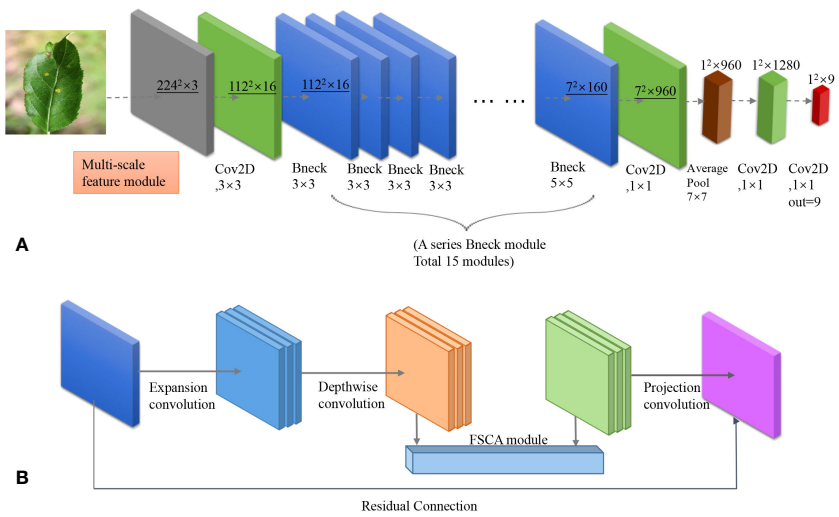


FIGURE 1
(A) Network structure of MobileNet-MFS. **(B)** Detailed composition of a single bottleneck module.

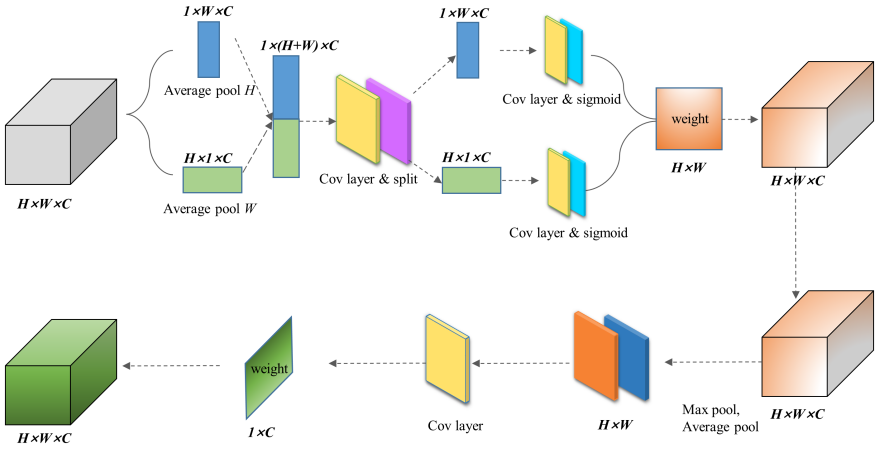


FIGURE 2
Network architecture of FSCA attention mechanism module.

concatenated modules. The first module mainly focuses on aggregating features in the spatial directions of X and Y. By averaging pooling in the X and Y directions and performing concat operation, a 1 × (H+W) × C dimensional array is obtained. Furthermore, we normalized the array through convolution, separated it, and activated it with a sigmoid function to obtain a set of weights containing information in the X and Y directions. Afterward, the weights are multiplied with the original data to obtain a set of directional perception feature layers. These transformations allow the attention module to capture long-term dependencies along one spatial direction and preserve precise positional information along another spatial direction, which helps the network locate interested targets more accurately.

The second module focuses on channel attention. In this module, we will take the maximum and average values of the input feature layers on the channels of each feature point. Afterward, we stack these two values and adjust the number of channels using a convolution with a channel count of 1. Then, we take a sigmoid function and obtain the weights of each feature point in the input feature layer (between 0 and 1). After obtaining this weight, we multiply it by the original input feature layer.

By concatenating and multiplying the two steps, we obtain our FSCA attention mechanism, which focuses on both the X and Y dimensions of input and the fusion of information of channels. Therefore, the obtained results are more comprehensive. Since our attention mechanism fused both spatial and channel information, we named it FSCA attention mechanism, which references CBAM Woo et al. (2018) and CA Hou et al. (2021) attention mechanism. In the following experiments, it was demonstrated that the FSCA mechanism helped our model better identify the characteristics of apple leaf diseases.

2.3 Multi-scale feature extraction

For apple leaf diseases, there are two main characteristics that are not easily extracted by machines. One is that there is a significant difference in the size of the disease on the leaves, such as Powdery Mill Draw and Grey spot lesions. Another type of disease is that its color or other details may vary depending on the scope of the disease, such as Grey spot and Rust lesions.

The above features involve dimensions of different sizes and are not easily captured by MobileNet V3, which mainly uses 3 × 3 and 5 × 5 convolution operations. In order to enable the machine to capture more features from different dimensions, Li et al. (2020) we have added a multi-scale feature extraction module to the front end of the input layer.

The structure of this module is shown in Figure 3. Four dimensions of convolution: 1 × 1, 3 × 3, 5 × 5, and 7 × 7 were applied in the module. After the image is convoluted, it is merged into a new feature map and then placed ahead of the network. Through such feature extraction, the accuracy of disease classification was improved.

3 Experimental results

3.1 Dataset

The images of apple leaves were collected from both laboratory and outdoor environments, with a total of eight diseases. These leaves were divided into nine categories, and each photo was labeled with the disease type. Our data mainly comes from PlantVillage, PPCD2020, PPCD2021, and ATLDSD datasets. PlantVillage is mainly from laboratory environments, while images from the PPCD2020 and PPCD2021 are collected in natural environments. The total number of samples is 15250, including 12204 for the training set and 3046 for the testing set. The sample ratio for the training and testing sets is 4:1.

As shown in Figure 4, there are a total of eight apple diseases in our sample, namely Alternaria leaf spot, Brown spot, Frogeye leaf spot, Grey spot, Mosaic, Powdery Mildew, Rust and Scab. The number of samples was collected in Table 1. Both Brown spot and Mosaic form large spots on the leaves, but the former will first cause

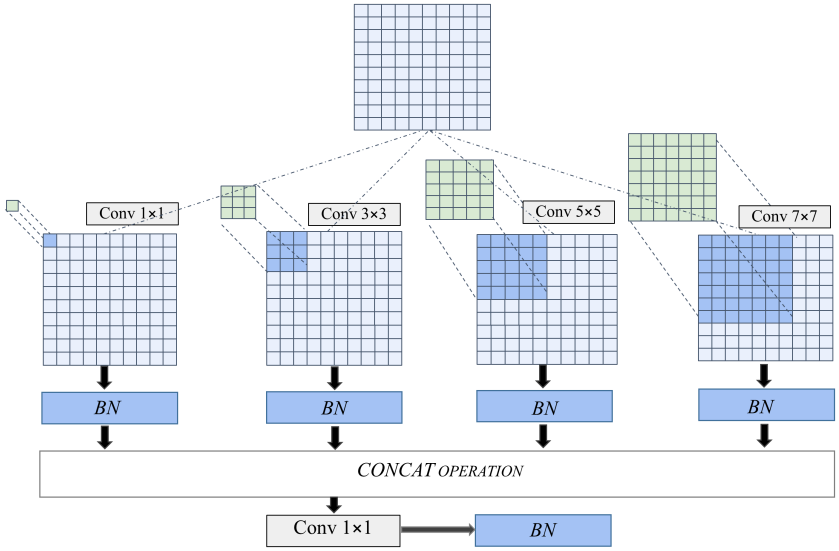


FIGURE 3
Compositions of multi-feature extraction module.

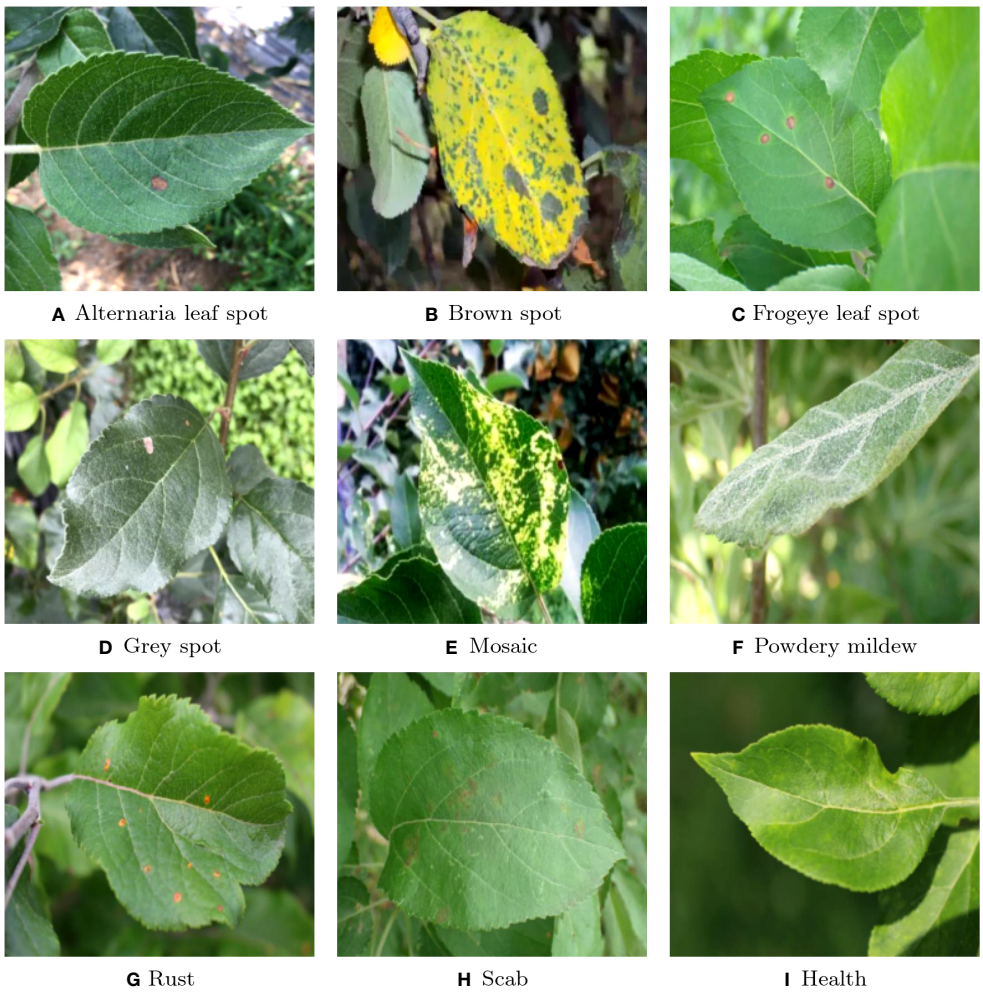


FIGURE 4
Classification of the samples: **(A)** Alternaria leaf spot; **(B)** Brown spot; **(C)** Frogeye leaf spot; **(D)** Grey spot; **(E)** Mosaic; **(F)** Powdery mildew; **(G)** Rust; **(H)** Scab; **(I)** Health.

TABLE 1 Number of samples from different diseases.

Types	Training Sample	Test Sample	Total Sample	Total (data augumation)
Alternaria leaf spot	526	131	657	1578
Brown spot	354	88	442	1062
Frogeye leaf spot	2544	635	3179	7632
Grey spot	285	71	356	855
Health	704	175	879	2112
Mosaic	316	79	395	948
Powdery mildew	947	236	1183	2841
Rust	2202	550	2752	6606
Scab	4326	1081	5407	12978
Total Number	12204	3046	15250	36612

the diseased parts of the leaves to turn yellow in a large area. Powdery Mildew can turn the veins of the leaves white and stain the leaves with white spots. Many other plants also suffer from similar diseases, such as strawberries. Other diseases can cause various types of spots on the leaves, such as Rust causing red spots on the leaves, while Gray spots causing gray spots, and Frogeyes causing yellow-brown spots on the center, similar to those on the outer ring of a frog's eye. In order to distinguish these different types of spots, neural networks need to first be able to capture these spots and further distinguish the different features of color and shape in the spots.

3.2 Evaluation metric

Accuracy is the most commonly used indicator, which represents the proportion of the true value of a model in the overall population. However, measuring the quality of a model cannot be solely based on accuracy. Some other indicators also reflect the quality of the model. For example, precision focuses on the model's ability to avoid false positives, while recall focuses on the model's ability to identify all positive instances. At the same time, when the dataset of the model is imbalanced, the f1-score balances the results of recall and precision, which better reflects the advantages and disadvantages of the model. The area under curve (AUC) shows the trade-off between the true positive rate and the false positive rate. Higher AUC values indicate better discriminability of the model. Therefore, accuracy is used with other performance metrics like precision, recall, f1-Score, and AUC. The definition of accuracy is:

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN} \tag{1}$$

where TN = true negative, FN = false negative, TP = true positive, and FP = false positive.

The expression of precision, recall, and f1-score are equations (2–4), respectively.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

4 Result

4.1 Accuracy and Loss

The accuracy and loss values of the model are shown in Figure 5. By analyzing the images, we can conclude that the accuracy of training and testing has improved to over 97% after 20 epochs. For the training data, the loss is around 0.5, while for the test data, the loss stabilizes below 0.1 after 20 epochs. When epochs approach 80, the model achieved a maximum accuracy of 98.7%.

4.2 Confusion matrix

The confusion matrix of the experiment is shown in Figure 6, where the horizontal and vertical coordinates represent the disease predicted by the model and the real disease respectively. Therefore, when the prediction is consistent with the actual situation, the axis data of the matrix will be added by one. When the predicted disease is inconsistent with the actual disease, the increased value of the matrix appears in the nondiagonal region. Take 'Rust' as an example, 534 cases of Rust were accurately identified, but 4 cases were misdiagnosed as Frogeye, 2 cases were misdiagnosed as health, and 10 cases were misdiagnosed as Scab. The 10 misdiagnosed cases were also the most common in the model, due to the similarity in size and color between rust and scab. Next, we want to further modify the model to better distinguish between the two diseases.

4.3 ROC

We have depicted the Receiver Operating Characteristic (ROC) curve of each disease, as shown in Figure 7. It should be noted that

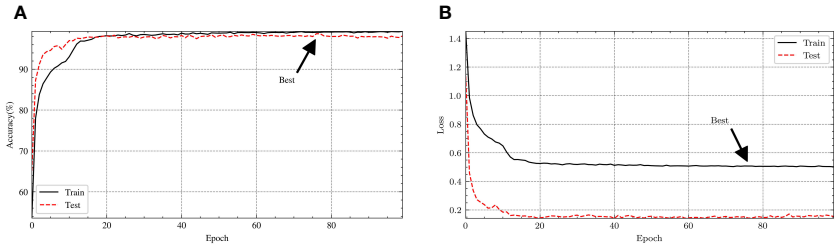


FIGURE 5
The **(A)** accuracy and **(B)** loss curve of the experiment.

the true positive rate of various diseases is high, resulting in a very steep ROC curve. The curve of Gery spot is different from several other diseases, as it initially reaches around 0.95. When the false positive rate reaches over 0.6, the true positive rate further increases to over 0.98. The steep ROC curve shows that the model can distinguish various diseases very well. In contrast, the ROC of general models is only diagonal.

4.4 Comparison with other attention mechanisms

In order to visually display the impact of different attention mechanisms, we calculated and compared the accuracy of different attention mechanisms (SE, ECA, CBAM, CA, FSCA, MFS) within the MobileNet v3 framework. As shown in Figure 8, our proposed FSCA attention mechanism and combined multi-scale MFS attention mechanism grow rapidly with epochs but are slightly slower compared to other types. But when the epoch increases to 20, their stability and maximum value are the best. In contrast, the fluctuation amplitude of other attention mechanisms is relatively large, while the accuracy of the MFS and the FSCA mechanism fluctuates at the highest point, demonstrating special stability.

4.5 Comparison with other CNNs

The accuracy of different CNNs and MobileNet-MFS are also compared. As shown in Figure 9, the light gray curve represents the accuracy curve of MobileNet-MFS. Compared with other models, it also rises very quickly and gradually reaches its high-level platform after 20 epochs. At the 28th epoch, MobileNet-MFS has an accuracy of around 98%, which is better than other models at the same epoch. Finally, when the epoch reaches 75, the MobileNet-MFS reaches its maximum accuracy of 98.7%, surpassing all other models.

In order to comprehensively compare our model with other classic models, we calculated several indicators such as precision, recall, f1 score, and AUC. These indicators can measure the model's

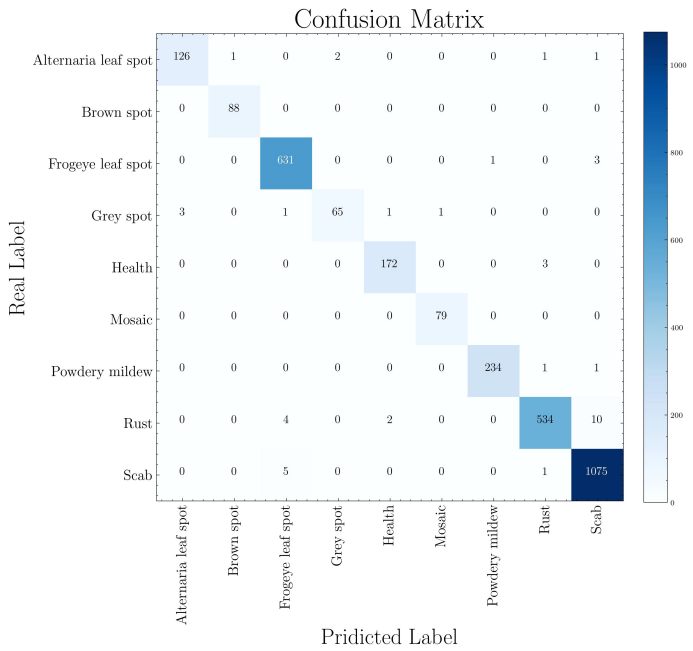


FIGURE 6
Confusion matrix of disease classification.

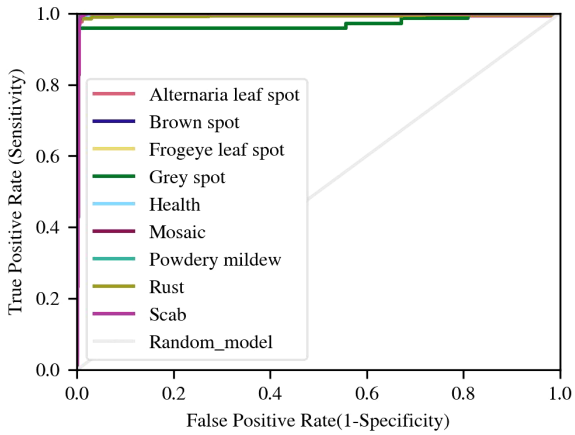


FIGURE 7
ROC curves of disease samples.

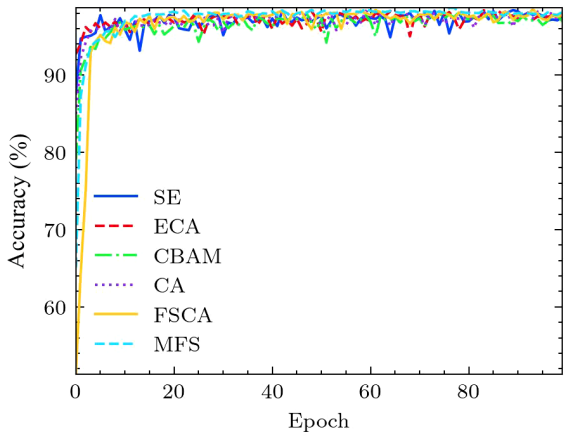


FIGURE 8
Comparison of accuracy curves for different attention mechanisms.

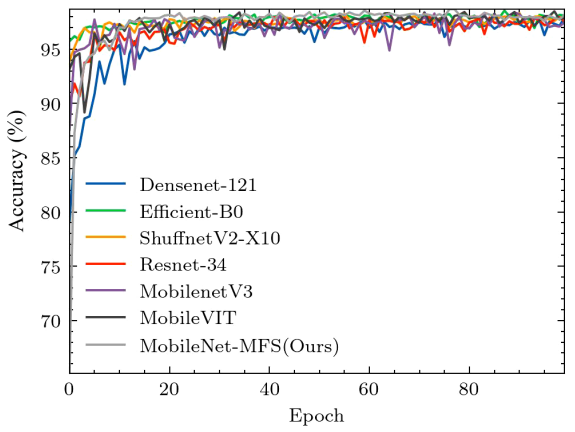


FIGURE 9
Comparison of accuracy curves for different models.

capabilities from different aspects. From Table 2, we can note that the MobileNet-MFS has the highest metrics in precision, recall and f1-score. However, in terms of AUC, it is not as good as a group of models such as EfficientNet-B0 and MobileNet-VIT.

Regarding the comparison of model performance, in addition to the above indicators, it is also necessary to consider the computational resources used by the models. MobileNet-MFS is based on MobileNet v3 and belongs to a lightweight CNN. The lightweight of the model will help it be applied to a wider range of scenarios. In addition, the computational complexity of the model is also a very important indicator, and the FLOPs provide an effective method to measure the computational complexity of the model. The indicators provided in Table 3 help us measure various aspects of the model more comprehensively. Taking into account parameter counts, memory size, and FLOPs counts, The MobileNet-MFS has more advantages over EfficientNet-B0, ResNet-34, and DenseNet-121, consuming slightly more computing resources than MobileNet v3, but not as streamlined as ShuffleNet v2.

In summary, through the comparison of various indicators, parameter quantities, and computational complexity, we can conclude that although many excellent models have emerged for image classification, MobileNet-MFS is still a state-of-the-art model.

5 Discussions

Finally, we utilized Gradient-weighted Class Activation Mapping (GRAD-CAM) to extract network recognition feature maps of images. Through these feature maps, we can more intuitively see the model's recognition of image features. As shown in Figure 10A, the Alternaria leaf spot on the leaf is very well and directly identified. From Figures 10B, C, it should be noted that the lesion areas on the Rust and Gray spot leaves with multiple spots have also been simultaneously observed, without any omissions or misjudgments. As shown in Figure 10D, the large area of yellow on the brown spot was well captured by our model, and the spots on the brown spot were also given special attention. These figures demonstrate the model's excellent feature capture ability.

The error case of MobileNet-MFS is also checked, and these images are selected from the library. As shown in Figures 11A, B, the Rust lesion can be accurately captured by our model. However, the leaves in Figures 11C, D with Frogeyes disease were mistakenly identified by the model as Rust-infected leaves. It can be deduced that these erroneous cases are due to the many similarities in the characteristics of these two diseases, and this discrimination error should be very difficult for CNNs.

From the perspective of incorrect images, it is actually difficult for the human eye to distinguish between the two situations. We cannot rule out that the database itself may still have misclassification in some cases. Without proper management, the error rate of the human eye itself is within the range of 5% -10%. If artificial intelligence is well-trained, it can surpass human recognition ability. Therefore, considering randomness, we believe that certain errors are inevitable.

Simply comparing accuracy, our work is inferior to some recent work. However, on the one hand, our dataset differs from theirs, as a large proportion of the images in our dataset are collected from the natural environment. On the other hand, the parameters and

TABLE 2 Precision, Recall, F1-Score and AUC for different models.

Model	Precision	Recall	F1-Score	AUC
MobileNetV3	0.982257	0.982272	0.982245	0.996483
Densenet121	0.978340	0.978332	0.978258	0.998184
EfficientB0	0.985624	0.985555	0.985524	**0.998827**
ShuffnetV2 X10	0.981947	0.981944	0.981842	0.998230
Resnet34	0.979438	0.979317	0.979282	0.997757
MobileVIT	0.984214	0.984242	0.984185	0.998727
MobileNet-MFS	**0.986198**	**0.986211**	**0.986156**	0.996105

TABLE 3 Comparison of operational and parameter performance among different models.

Model	TOP-1 Accuracy (%)	Parameters Count (Millions)	Memory Size (MB)	FLOPs Count (MFLOPs)
MobileNet-MFS	98.69	4.96	51.30	251.94
MobileNetV3	98.39	4.21	50.39	226.44
Densenet121	97.90	6.96	147.10	2881.60
EfficientB0	98.56	4.02	79.40	398.03
ShuffnetV2 X10	98.33	1.26	20.84	149.58
Resnet34	98.09	21.29	37.61	3673.72
MobileVIT	98.49	1.94	–	743.48

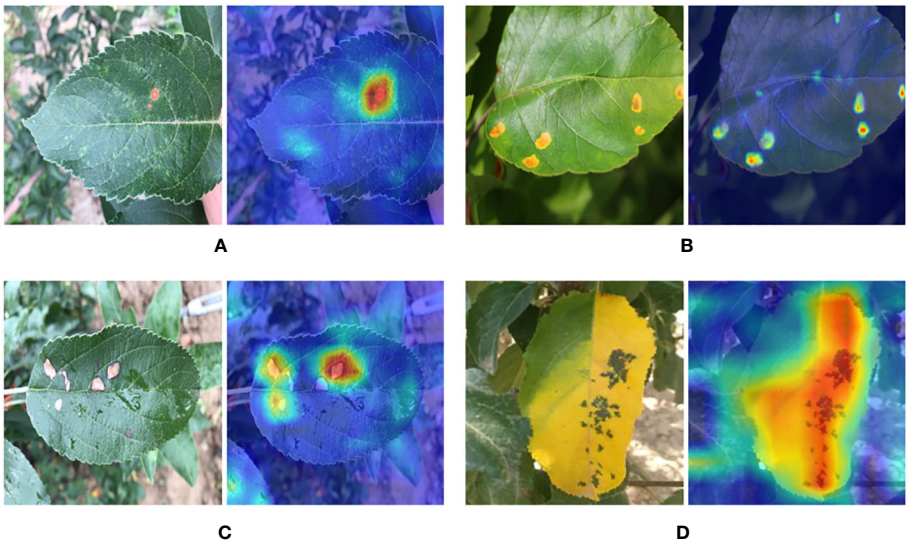


FIGURE 10
Heat map display of feature extraction of leaf disease sites: **(A)** Alternaria leaf spot **(B)** Rust **(C)** Grey spot **(D)** Brown spot.

operation time of our model are also different. Although 98.7% is a high-level score for the classification of leaf diseases, the images in our dataset have been well processed, so they cannot fully restore the real usage scenarios. We have not yet processed images taken in orchard environment, therefore it is the weakness of our work. Our next step is to develop a network that can process drone and robot camera images, remove unclear and messy backgrounds, and make accurate classifications on mobile devices.

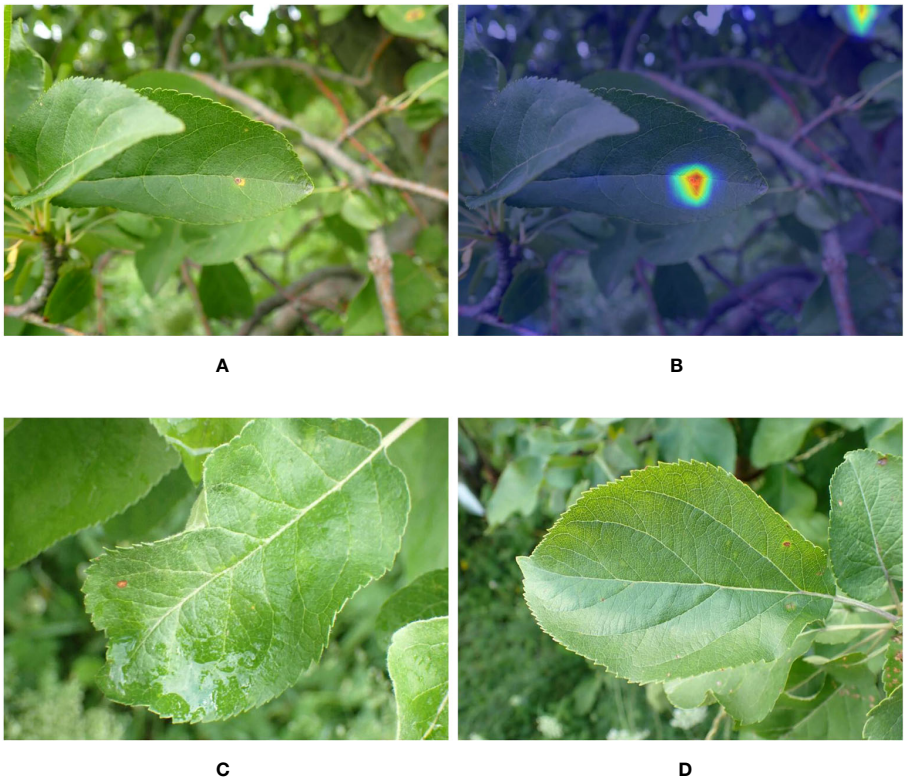


FIGURE 11
(A) Leaves with Rust disease. **(B)** Heat map of feature extraction of the Rust lesion site. **(C, D)** Mistakenly identified leaves with Frogeye disease.

6 Conclusions

The identification of apple leaf diseases is very difficult, thanks to the development of deep learning, a series of models have shown great achievement in identifying leaf diseases. On the basis of these works, we have improved MobileNet v3 by modifying its attention mechanism, taking into account the influence of dimension and space. At the same time, we have added a multi-scale feature extraction module to further improve the performance of the network. By comparing with similar models, we found that our proposed MobileNet-MFS showed the best performance in terms of accuracy and stability. This also indicates that our proposed attention mechanism and multi-scale module have effectively improved the feature capture ability of the model for leaf diseases, and there is also hope for their application in other aspects. We also calculated the ROC and confusion matrix of the model, which shows that the model is very good at resolving various diseases. Finally, we reviewed the feature extraction graph of the model through GRAD-CAM and analyzed the error cases. Compared to previous models, the model is more efficient mainly due to the mutual cooperation of two aspects. FSCA and multi-scale respectively increase the model's feature discovery ability and the implementation of more scale features, both of which are crucial for getting more accurate classifications. This work indicates that the MobileNet-MFS is a very effective model for distinguishing apple leaf diseases, and the FSCA attention mechanism used in this model is also worthy of further application in other scenarios.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

HC: Methodology, Software, Visualization, Writing – review & editing. HL: Investigation, Visualization, Writing – original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The authors thank financial support from Key Projects in Shandong Province for Undergraduate Teaching Reform Research (Z2022150).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., and Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry* 11, 939. doi: 10.3390/sym11070939

Chen, J., Chen, J., Zhang, D., Sun, Y., and Nanehkaran, Y. (2020). Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* 173, 105393. doi: 10.1016/j.compag.2020.105393

Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., and Traore, D. (2019). Deep neural networks with transfer learning in millet crop images. *Comput. Industry.* 108, 115–120. doi: 10.1016/j.compind.2019.02.003

Deng, L., Wang, Z., and Zhou, H. (2019). "Application of image segmentation technology in crop disease detection and recognition," in *Computer and computing technologies in agriculture XI*. Eds. D. Li and C. Zhao (Cham: Springer International Publishing), 365–374.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* 145, 311–318. doi: 10.1016/j.compag.2018.01.009

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17 (9), 2022. doi: 10.3390/s17092022

Hasan, R. I., Yusuf, S. M., Mohd Rahim, M. S., and Alzubaidi, L. (2022). Automated masks generation for coffee and apple leaf infected with single or multiple diseases-based color analysis approaches. *Inf. Med. Unlocked.* 28, 100837. doi: 10.1016/j.imu.2021.100837

Hou, Q., Zhou, D., and Feng, J. (2021). "Coordinate attention for efficient mobile network design," in *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Nashville, TN, USA, 13708–13717. doi: 10.1109/CVPR46437.2021.01350

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., et al. (2019). "Searching for mobilenetv3," in *2019 IEEE/CVF international conference on computer vision (ICCV)*, Seoul, South Korea, 1314–1324. doi: 10.1109/ICCV.2019.00140

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. arXiv:1704.04861.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2020). Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2011–2023. doi: 10.1109/TPAMI.2019.2913372

Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7, 59069–59080. doi: 10.1109/ACCESS.2019.2914929

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.* 147, 70–90. doi: 10.1016/j.compag.2018.02.016

Kavyashree, P. S. P., and El-Sharkawy, M. (2021). "Compressed mobilenet v3:a light weight variant for resource-constrained platforms," in *2021 IEEE 11th annual computing and communication workshop and conference (CCWC)*, NV, USA, 0104–0107. doi: 10.1109/CCWC51732.2021.9376113

Khan, M. A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., et al. (2018). Ccdf: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep cnn features. *Comput. Electron. Agric.* 155, 220–236. doi: 10.1016/j.compag.2018.10.013

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Li, Z., Yang, Y., Li, Y., Guo, R., Yang, J., and Yue, J. (2020). A solanaceae disease recognition model based on se-inception. *Comput. Electron. Agric.* 178, 105792. doi: 10.1016/j.compag.2020.105792

Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., and Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Comput. Electron. Agric.* 154, 18–24. doi: 10.1016/j.compag.2018.08.048

Mokhtar, U., Ali, M. A. S., Hassenian, A. E., and Hefny, H. (2015). "Tomato leaves diseases detection approach based on support vector machines," in *2015 11th international computer engineering conference (ICENCO)*, Cairo, Egypt, 246–250. doi: 10.1109/ICENCO.2015.7416356

Pardede, H. F., Suryawati, E., Sustika, R., and Zilvan, V. (2018). "Unsupervised convolutional autoencoderbased feature learning for automatic detection of plant diseases," in *2018 international conference on computer, control, informatics and its applications (IC3INA)*, Tangerang, Indonesia, 158–162. doi: 10.1109/IC3INA.2018.8629518

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 4510–4520. doi: 10.1109/CVPR.2018.00474

Shin, J., Chang, Y. K., Heung, B., Nguyen-Quang, T., Price, G. W., and Al-Mallahi, A. (2021). A deep learning approach for rgb image-based powdery mildew disease detection on strawberry leaves. *Comput. Electron. Agric.* 183, 106042. doi: 10.1016/j.compag.2021.106042

Shoaib, M., Hussain, T., Shah, B., Ullah, I., Shah, S. M., Ali, F., et al. (2022). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1031748

Singh, A., Ganapathysubramanian, B., Singh, A. K., and Sarkar, S. (2016). Machine learning for highthroughput stress phenotyping in plants. *Trends Plant Sci.* 21, 110–124. doi: 10.1016/j.tplants.2015.10.015

Sun, H., Xu, H., Liu, B., He, D., He, J., Zhang, H., et al. (2021). Mean-ssd: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. *Comput. Electron. Agric.* 189, 106379. doi: 10.1016/j.compag.2021.106379

Too, E. C., Yujian, L., Njuki, S., and Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* 161, 272–279. doi: 10.1016/j.compag.2018.03.032

Wang, Y., Wang, H., and Peng, Z. (2021). Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Syst. Appl.* 178, 114770. doi: 10.1016/j.eswa.2021.114770

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "Cbam: Convolutional block attention module," in *Computer vision – ECCV 2018*. Eds. V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (Cham: Springer International Publishing), 3–19.

Xiong, Y., Liang, L., Wang, L., She, J., and Wu, M. (2020). Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset. *Comput. Electron. Agric.* 177, 105712. doi: 10.1016/j.compag.2020.105712

frontiers | Frontiers in Plant Science

TYPE Review
PUBLISHED 02 November 2023
DOI 10.3389/fpls.2023.1268167

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Changji Wen,
Jilin Agricultural University, China
Kamil Dimililer,
Near East University, Cyprus

*CORRESPONDENCE
Dan Popescu
dan.popescu@upb.ro

RECEIVED 27 July 2023
ACCEPTED 11 October 2023
PUBLISHED 02 November 2023

CITATION
Popescu D, Dinca A, Ichim L and Angelescu N (2023) New trends in detection of harmful insects and pests in modern agriculture using artificial neural networks. a review.
Front. Plant Sci. 14:1268167.
doi: 10.3389/fpls.2023.1268167

New trends in detection of harmful insects and pests in modern agriculture using artificial neural networks. a review

Dan Popescu[1]*, Alexandru Dinca[1], Loretta Ichim[1] and Nicoleta Angelescu[2]

[1]Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania, [2]Faculty of Electrical Engineering, Electronics, and Information Technology, University Valahia of Targoviste, Targoviste, Romania

Modern and precision agriculture is constantly evolving, and the use of technology has become a critical factor in improving crop yields and protecting plants from harmful insects and pests. The use of neural networks is emerging as a new trend in modern agriculture that enables machines to learn and recognize patterns in data. In recent years, researchers and industry experts have been exploring the use of neural networks for detecting harmful insects and pests in crops, allowing farmers to act and mitigate damage. This paper provides an overview of new trends in modern agriculture for harmful insect and pest detection using neural networks. Using a systematic review, the benefits and challenges of this technology are highlighted, as well as various techniques being taken by researchers to improve its effectiveness. Specifically, the review focuses on the use of an ensemble of neural networks, pest databases, modern software, and innovative modified architectures for pest detection. The review is based on the analysis of multiple research papers published between 2015 and 2022, with the analysis of the new trends conducted between 2020 and 2022. The study concludes by emphasizing the significance of ongoing research and development of neural network-based pest detection systems to maintain sustainable and efficient agricultural production.

KEYWORDS

insect detection, pest detection, precision agriculture, image processing, deep learning, artificial neural networks

1 Introduction

The adoption of artificial intelligence (AI) and integrated structures has rapidly become multidisciplinary and spread across various fields, dominating research areas and plans in previous years (Zhang, 2022). Thanks to the technological advancements in the field of AI and more importantly in the field of deep learning (DL), a multitude of domains enjoy

notable results for various associated tasks (Abade et al., 2021). This advance has brought such technologies to the fore with great success, its upward trajectory and continued development being supported by a range of technological, financial, and educational resources (Kumar and Kukreja, 2022). The integration of AI and integrated structures has significantly impacted insect pest detection, offering innovative solutions to this pressing agricultural and environmental concern. This evolution is driven by advancements in DL and supported by substantial resources, ultimately resulting in the development of highly efficient and sustainable techniques for insect pest detection and management (LeCun et al., 2015).

Considering the agricultural field, these techniques have enjoyed great popularity and started to be adopted on a large scale, where human labor does not have the necessary time and speed to analyze the data in a timely manner and to cover considerable areas in the monitoring area (De Cesaro Júnior & Rieder, 2020). Often, these features are more than useful and relevant to every operation, and early detection, monitoring, and classification deliver results to match (Ampatzidis et al., 2020). Due to this aspect, automation areas have been successfully introduced and are based on thorough research and massive development and optimization techniques (Ahmad et al., 2022). Technological advances, particularly in deep learning (DL), have been critical in the identification of insect pests. These breakthroughs have resulted in tremendous progress in correctly detecting and managing insect pests in agriculture and other industries. In recent years, the use of artificial intelligence (AI) and integrated structures has spread to a variety of disciplines, with a special emphasis on insect pest identification. This integrative approach has gained prominence in research agendas, altering how we address pest-related concerns.

Abbreviations: ACC, Accuracy; AI, Artificial Intelligence; ANN, Artificial Neural Network; API, Application Programming Interface; BPNN, Back-Propagation Neural Network; CAD, Computer Aided Diagnosis; C-GAN, Conditional Generative Adversarial Network; CNN, Convolutional Neural Network; CSA, Channel-Spatial Attention; DA, Dragonfly Algorithm; DB, Database; DC-GAN, Deep Convolutional Generative Adversarial Network; DCNN, Deep Convolutional Neural Network; DL, Deep Learning; DS, Dataset; F1, Dice Coefficient (F1 Score); FPN, Feature Pyramid Network; GaFPN, Global Activated Feature Pyramid Network; GAM, Global Activated Module; GAN, Generative Adversarial Network; IoT, Internet of Things; IPM, Integrated Pest Management; KNN, K-Nearest Neighbor; LSTM, Long-Short Term Memory; mAP, Mean Average Precision; MBD, Maryland Biodiversity Database; ML, Machine Learning; MLP-ANN, Multilayer Perceptron Artificial Neural Network; MSR, Multi-scale super-resolution; NIN, Network in Network; NMS, Non-Maximum Suppression; ORB, Oriented Rotated Brief; PRE, Precision; PSSM, Position-sensitive score map; R-CNN, Deep region based convolutional neural network; ReLU, Rectified Linear Units; RGB, Red, Green, Blue; ROI, Region of Interest; RPN, Region Proposal Network; SANN, Smart Agriculture Neural Network; SEN, Sensitivity; SMOTE, Synthetic minority over-sampling technique ; SOTA, State-of-the-art; SPE, Specificity ; SSD, Single Shot Detector; SVM, Support Vector Machine; UAV, Unmanned Aerial Vehicle; YOLO, You Only Look Once; ZF, Zeiler and Fergus Model.

Intelligent and precision techniques are necessary for farmers, especially for automation, because they reduce the complexity of pest detection and counting estimation, compared to a process done manually by farmers or authorized auxiliary persons, this process being expensive and requiring a lot of time execution (Ahmad et al., 2018; Apolo-Apolo et al., 2020; Thakare and Sankar, 2022). Solutions based on DL and the automation of the processes involved in crop management prove to be effective, with high coverage and low costs (Iost Filho et al., 2019). At the same time, it helps the process of detecting and managing pests in a timely manner, without resorting to highly invasive solutions and representing effective measures (Mavridou et al., 2019).

Considering the chemical treatment applied with pesticides, the amounts administered become directly proportional to the degree of infestation and do not present sustainability characteristics, as they are present or required in modern development areas. Pest populations cause massive, considerable damage to crops of various types and sizes. This highlights an important point because agriculture is the most significant economic branch in many countries (Cardim Ferreira Lima et al., 2020). Monitoring, managing, and protecting crops from insect pests is an important step and an area of thorough research (Zhu et al., 2020). In an unfortunate setting, the productivity and production volume of agricultural areas is strongly affected by the appearance and presence of pests and their widespread (Ahmad et al., 2022). The identification and monitoring of pests, mostly represented by insects, and careful management of crops are of interest in agricultural development. Many times, the management of these pests takes place in poorly managed processes, without clear expertise, and often based on invasive, non-sustainable, and polluting solutions (Wen & Guyer, 2012). Modern models and techniques based on AI and DL, especially image processing and convolutional neural networks (CNNs), are very useful and effective in the so-called precision agriculture (PA) or integrated pest management (IPM) (Mavridou et al., 2019). The way to combine automatic or supervised image acquisition using drones and digital cameras with the emphasized developments of models based on CNNs was a great success (Du et al., 2022; Zhang et al., 2022).

The continuous progress of DL models has brought to the fore several notable applications for pest management and PA in general. CNNs, as part of DL, represent a state-of-the-art around image analysis and are mainly and successfully used for the development of classification, object detection, or segmentation tasks (Wang et al., 2017; Zhang et al., 2020). In principle, the convolution techniques and the mathematical models present among them make possible the existence and continuous expansion of the previously mentioned techniques and even their strong development, modification, or optimization. Starting from an initial and innovative step, these types of techniques have been developed and researched along the way, having today a series of remarkable architectures with adequate performance in various tasks (Tian H. et al., 2020; Zhang et al., 2022). The study (Nanni et al., 2022) addresses the problem of automatic identification of invasive insects to combat crop damage and losses. The authors created ensembles CNNs using various topologies optimized with different Adam variants for pest identification. The best ensemble,

combining CNNs with various Adam variants, achieved impressive results, surpassing human expert classifications on several known datasets. With the awareness that agricultural pests severely impair food crop quality, the importance of agriculture as an economic backbone is underlined in (Sanghavi et al., 2022). Machine learning models have been employed to handle pest categorization and detection, however they suffer when dealing with insects that have similar traits but live in diverse environments. The paper offers an enhanced deep learning model named Hunger Games Search-based Deep Convolutional Neural Network (HGS-DCNN) for efficient insect identification with improved accuracy to address this difficulty. The process of recognizing and classifying insects, addressing several challenges, was proposed by the authors in paper (Xia et al., 2018) locating information on an insect quickly as part of a complex backdrop, precisely recognizing insect species, especially when they are highly similar within the same species (intra-class) and across species (inter-class) and identifying differences in the appearance of the same insect species at various stages of development. These issues are crucial in the field of insect recognition and categorization.

Starting with a motivation area, we highlighted IPM and PA for this study. There are several problems facing the current agricultural sector in terms of production management, security, and the negative impact of external and biological/natural factors (Csillik et al., 2018; Ronchetti et al., 2020). Speaking of the agricultural area, the desire for sustainability has brought to the fore a series of characteristics represented by IPM and a series of actions for the areas where it can be applied. Basically, IPM represents a collection of good practices to attract attention and give rise to effective approaches in the fight against pest populations and for the optimal and timely management of the associated effects (Cardim Ferreira Lima et al., 2020). IPM has developed over the years based on up-to-date, well-verified information and gradual adoption. A series of studies developed and researched this topic in detail for the construction of PA areas, with innovative and well-documented techniques (Velusamy et al., 2022). Moreover, the desire for sustainability quickly accentuated this. The accuracy of the information, the continuous monitoring, and the effective IPM documentation make possible the emergence and continuous support of good practices that can be successfully applied to the development of the agricultural field (Ronchetti et al., 2020; Misango et al., 2022). In principle, the adoption of IPM is done for the adequate control of pests and to reduce them and their effects to a tolerable level. On the other hand, the IPM effect also has a considerable positive impact on the environment and the population. The desire for adoption is primarily emphasized by the decrease in the amounts of pesticides used after prior monitoring. The effects of pests, their presence, and plant diseases represent a serious threat to agricultural production and the resulting food security due to the agricultural sector (Misango et al., 2022; Wu et al., 2019). The IPM objective is to create a combination of actions associated with good practices to develop specific solutions for each agricultural area and culture. Although IPM notions and application methods are not relatively new techniques, a considerable number of studies have emerged to identify the status and trends of the agricultural sector regarding the existence of these good practices that IPM wishes to highlight.

As highlighted by the authors (Damos, 2015) the management of pests in a sustainable or ecological way brings into question the reduction of pesticides and the adoption of alternatives for the control and development of production in a safe and ecological way. Being a basic field, agriculture represents a sector that has enjoyed a series of changes over time marked by automation, modern crop management and monitoring models, and various smart methodologies. Research developed by the authors (Deguine et al., 2021) shows the impact and evolution of IPM practices over the last five or six decades. Data needed for the area of crop profiles, pesticides, and strategy plans for the safe management of agricultural areas were noted by (Bouroubi et al., 2022) to highlight an educational basis for decision-making and risk assessment. data creation and documentation were noted as necessary and examples of databases and applications that can be used for continuous and quality information with high availability were highlighted. The need for access to data and the influence of IPM adoptions were also noted by the authors (Tong et al., 2022) for the agricultural production area. Here, several mechanisms and factors for the adoption of good practices by farmers and the attached IPM notions, as well as research trends in these directions, have been noted. In a more advanced framework, the authors of the meta-analysis (Sekabira et al., 2022) emphasized socio-economic factors with impact in the combined area of IPM and climate-smart CS-IPM. To ensure the sustainability of agricultural ecosystems, the authors analyzed and noted the strategic determinants for the adoption of smart innovations in the case of modern agriculture and environmental policies. CS-IPM involves a range of practices and techniques that are tailored to local conditions and needs. These include crop diversification, conservation agriculture, integrated pest management, and the use of climate-resilient crop varieties.

Modern agriculture has great potential and is aided today by several powerful working and monitoring technologies to increase productivity, efficiency, and the eco-friendliness that can be attached. Precision farming techniques and advanced methodologies have helped to increase food security and environmental sustainability (Wen and Guyer, 2012). Analyzing the papers highlighted for this study, there is a general trend of massive adoption of technological processes or automation in the agricultural area as part of the idea and methods involved in PA. It uses data and precision farming tools such as sensors, drones, and precision planting equipment to gather information about soil, weather, and crop growth, and then use that information to make precise, data-driven decisions about planting, fertilizing, harvesting crops or pest detection and management (Popescu et al., 2020). This can help farmers to increase yields, reduce costs, and improve the efficiency of their operations, being a major advantage to achieve modern targets such as sustainability and ecological production.

CNN-based systems for insects and pest detection have been successfully applied to a range of crops, including vegetables, fruits, and grains. In addition to identifying insects, CNNs can also detect damage caused by insects, such as holes and discoloration on plant leaves. This information can be used to quantify the severity of insect infestations and to guide pest management strategies. Digital images of plants and crops are obtained using cameras or drones

equipped with high-resolution sensors. These images are then analyzed using CNNs, that has been shown to be highly effective at image classification and object detection tasks. However, the models used for monitoring need training and validation of insect pest datasets and innovative optimizations. Examples of digital images for this topic, illustrating several known insect pests, are shown in Figure 1: A) Aulacophora indica, B) Bemisia tabaci, C) Sesamia inferens, D) Cicadella viridis, E) Cnaphalocrocis medinalis, F) Trigonotylus caelestialium, G) Emposca flavenscens, H) Pieris rapae, I) Ostrinia nubilalis, J) Epitrix fuscula, K) Halyomorpha halys, and L) Cydia pomonella.There are often problems in the highly accurate detection of insects of interest, as they are part of the natural setting where the conditions in which these insects are captured are not optimal – accurate detection is hindered by lighting conditions, various artifacts, or obturations of various types (leaves, flowers, branches, fruits). Based on these limitations, there has been continuous research and development aimed at creating innovative techniques for extracting information of interest from digital images that illustrate real contexts.

The presence of natural factors with a negative impact on performance inclined toward the development of research based on concrete work methods. The general workflow for insect detection and monitoring in modern agriculture using neural networks is composed of the following phases: a) Data collection, b) Data processing, c) NN training, and d) Validation and testing.

To start developing a system for insect pest detection using digital images and CNNs, the first step is to collect relevant data consisting of images of insects and crops, which would need to be labeled and categorized to identify the type of insects encountered (Partel et al., 2019; Nanni et al., 2022). The next step is data preprocessing, which involves removing noise, distortion, or other anomalies from the collected data (Du et al., 2022). This can include resizing images, adjusting brightness and contrast, and data augmentation (Ahmad et al., 2022). A great feature extraction

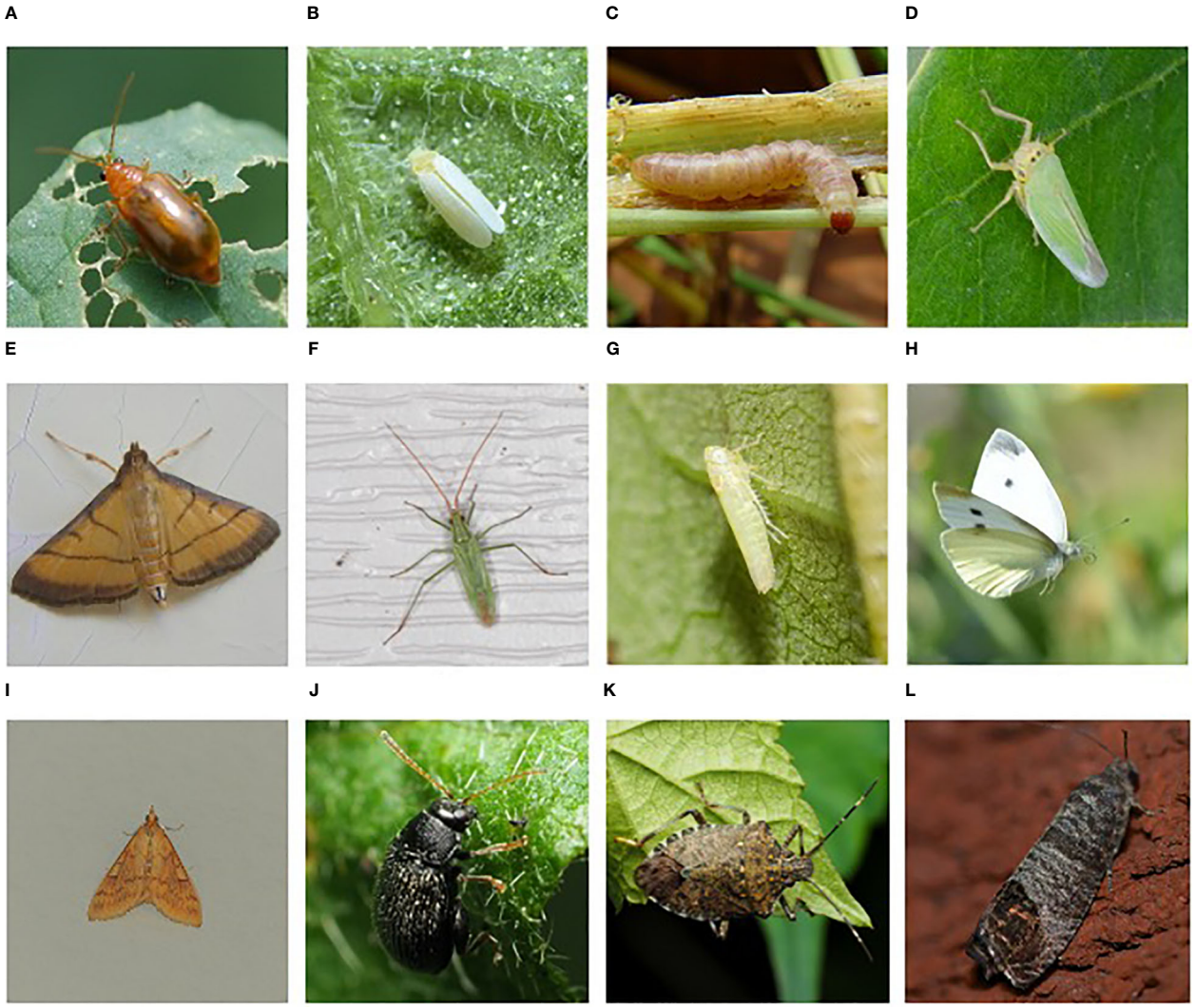


FIGURE 1
Examples of harmful insects for agriculture: **(A)** Aulacophora indica, **(B)** Bemisia tabaci, **(C)** Sesamia inferens, **(D)** Cicadella viridis, **(E)** Cnaphalocrocis medinalis, **(F)** Trigonotylus caelestialium, **(G)** Emposca flavenscens, **(H)** Pieris rapae, **(I)** Ostrinia nubilalis, **(J)** Epitrix fuscula, **(K)** Halyomorpha halys, **(L)** Cydia pomonella (Xie et al., 2018), (https://www.dlearningapp.com/web/DLFautoinsects.htm).

model can make use of DL techniques to focus attention on insect pests (Li et al., 2020; Li et al., 2022). From the preprocessed dataset, a representative subset of images needs to be selected for training CNNs to identify and classify insect pests in the images and adjust internal weights to improve accuracy (Khanramaki et al., 2021). Once trained, the CNN must be validated and tested on a separate dataset to evaluate its accuracy and identify any issues that need to be addressed. Since manual classification and detection are time-consuming automation using CNNs is preferred (Butera et al., 2021).

This paper wants to present a detailed review of the methods of automatic identification of populations of harmful insects by involving algorithms in the field of neural networks (NNs). Recently, it has been observed that the use of digital tools and services for the early and automatic detection of populations of harmful insects represents an impact factor on agricultural areas. Moreover, the optimization of agricultural processes in combination with these tools offers optimal and high-performance solutions. To facilitate reading the article, a list of abbreviations is given in Annex 1.

The presentation of the selected studies brings to the fore a series of key, modern methods related to the topic attached to the paper. Pest detection methods have made significant advancements over the years, but there are still several challenges and areas that need improvement in existing approaches. These challenges often include accuracy and reliability, data quality and quantity, integration with pest management, automation and scalability, real-time detection and species and diversity. Many current methods for pest detection still suffer from high rates of false positives (identifying non-pests as pests) or false negatives (failing to detect pests when they are present). On the other hand, developing accurate machine learning models for pest detection often requires large amounts of high-quality labeled data, which can be expensive and time-consuming to obtain. Imbalanced datasets, where certain pests are rare or hard to find, can lead to biased models that perform poorly on underrepresented pests. Pest species can be highly diverse, and methods that work for one pest may not be effective for others. Developing generalized detection methods that can adapt to different pests is a challenge.

2 Materials and methods

2.1 Investigation of references

The paper considered method workflow from PRISMA guidelines (Page et al., 2021) for insect detection and monitoring in agriculture based on NNs by investigating articles published between 2015 and 2022. This review article aims to provide an overview of the new trends and advancements in CNN research for insect pest detection in agriculture between 2015 and 2022. To select the papers for this review, the focus was primarily on papers that contribute to the development of CNN-based systems for insect pest detection in agriculture. Specifically, papers that propose novel CNN architectures, explore the use of transfer learning for insect pest detection, or apply CNNs to new insect pest detection tasks were prioritized. The selected papers demonstrate the power of CNNs in various applications for insect monitoring in modern agriculture, including object detection, segmentation, and recognition.

The research databases used in this review were: Web of Science, Scopus, and IEEE. Following the Prisma flow diagram (Figure 2), several criteria were attached for searching and extracting articles of interest. Although there was an initially large number of papers identified for the topic of this review, the initial selection criteria extracted approximately 354 relevant studies in the first instance. Of all these, only 138 were chosen based on the final criteria related to new periods, new trends, attachment in top publications, and innovation. An initially large number of diverse research for the modern agricultural area and a considerable evolution in recent years are observed.

Searches for important terms and evolution as article numbers during the last years in the Web of Science, Scopus, and IEEE Xplore DBs between 2015 and 2021 with AND connector are presented in Figure 3: A) (CNN) AND (agriculture) AND (image processing), B) (CNN) AND (agriculture) AND (insects), C) (CNN) AND (agriculture) AND (pest detection), D) (image processing) AND (pest detection), E) (CNN) AND (pest detection), and F) (CNN) AND (insects). The graphs highlight the strong increase in the number of research articles in the connected fields in recent years regarding the use of CNN.

2.2 Datasets used

A robust image database (DB) is crucial for DL classification and detection because it is the foundation upon which a model is trained (Ding & Taylor, 2016). The larger and more diverse the dataset is, the better the ML model's performance will be. A robust image dataset allows a DL model to learn a general representation of the objects or classes it is supposed to recognize. The more diverse the dataset, the better the model will be at recognizing new images that it has not seen before. Also, it enables an ML model to achieve higher accuracy in classification and detection tasks. When the dataset is comprehensive and covers a wide range of scenarios, the model can learn more accurately how to identify objects and classify them.

Insect pest databases were used in agricultural monitoring applications to track and identify the presence of insect pests that can damage crops (Turkoglu et al., 2022). These databases are typically created by agricultural organizations, universities, and research institutions that have collected data on the life cycles, presence, behavior, and distribution of various insect pests. In this regard, analyzing the papers selected for this study, several ways to construct datasets in training and validating models used for insect detection and identification were observed. Several public databases have been used by researchers in their studies to measure the performance of the implemented architectures and to test the defined models against the obtained results. Table 1 presents a summary of the most known and frequently used databases for modern insect pest monitoring applications in agriculture.

Insect pest image databases often include images of insects at different life stages, including larvae and adult stages (Zhang S. et al.,

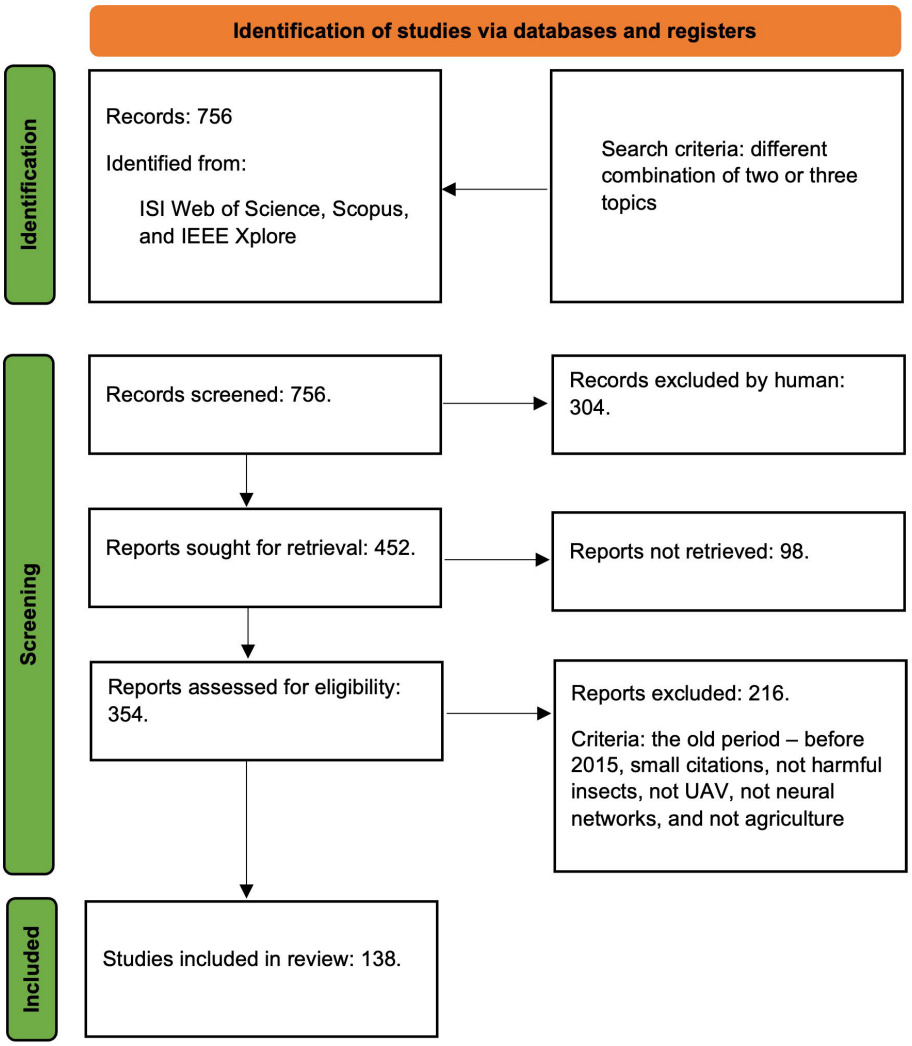


FIGURE 2
PRISMA 2020 flow diagram for this study.

2022). These images could be accompanied by additional information, such as the insect's common name, scientific name, and the types of crops or plants that the insect pest is known to damage. This is an important aspect because the primary purpose of an insect image database is to provide a visual reference for identifying insect pests in the field. Insect pest image databases can be used as educational resources to help people learn about the different types of insect pests and their impact on agriculture and the environment (Shi et al., 2020).

One of the databases that is highlighted in the present study and that was used by the researchers in the selected papers is the IP102 DB. As presented in the acronym, it contains 102 classes of common insect pests with hierarchical taxonomy and broadly totals around 72,222 images (see Table 1). The database is regularly updated and maintained by a team of experts in the field of entomology. It covers a wide range of insect orders. Each entry in the IP102 Insect Database includes information on the insect's scientific name, common name, description, habitat, diet, life cycle, behavior, and distribution, all being presented in high-quality images and illustrations, making it easy to identify different species.

The authors on IP102 DB note that existing image datasets primarily focus on everyday objects like flowers and dogs, limiting the applicability of advanced deep learning techniques in agriculture. To address this gap, they introduce a comprehensive dataset called IP102 for insect pest recognition. The authors conducted baseline experiments on the IP102 dataset using both handcrafted and deep feature-based classification methods. Their findings revealed that the dataset poses challenges related to inter-class and intra-class variance, as well as data imbalance. They anticipate that IP102 will serve as a valuable resource for future research in practical insect pest control, fine-grained visual classification, and addressing imbalanced learning challenges in this domain.

The Maryland Biodiversity Database (MBD) (Maryland Biodiversity Database, 2022) is another important database, and it has been used in various research works for the insect pest monitoring area. This database is a vast and valuable public

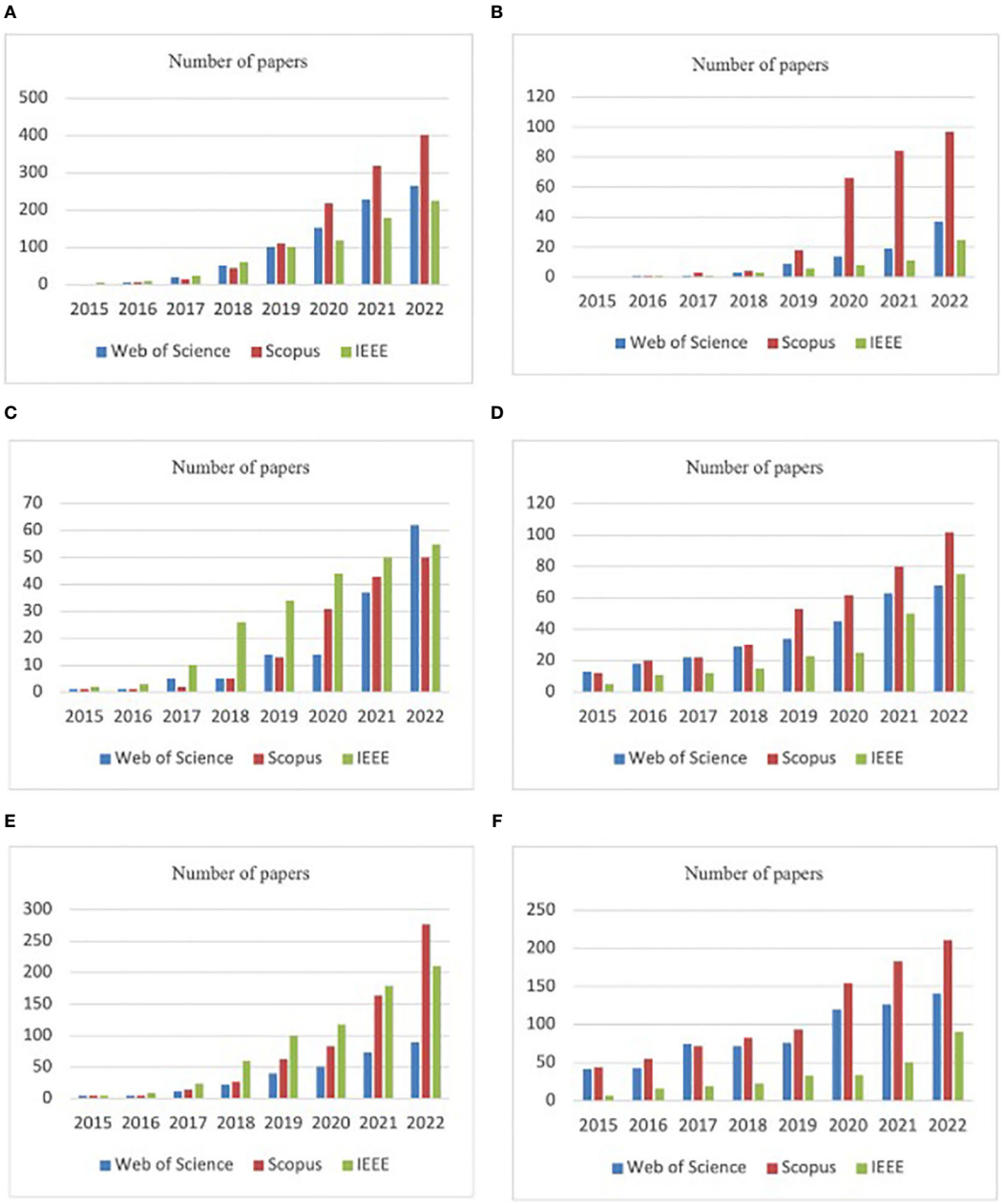


FIGURE 3
Searches for important terms in the Web of Science, Scopus, and IEEE Xplore DBs between 2015 and 2021 with AND connector: **(A)** (CNN) AND (agriculture) AND (image processing), **(B)** (CNN) AND (agriculture) AND (insects), **(C)** (CNN) AND (agriculture) AND (pest detection), **(D)** (image processing) AND (pest detection), **(E)** (CNN) AND (pest detection), **(F)** (CNN) AND (insects).

resource that can serve as an important tool in researching various information for the insect pest area and querying it can create diverse datasets. The MBD database provides ecological information about species, including their habitats and interactions with other organisms. This can be useful for understanding the ecological context of insect pests, their host plants, and their natural predators. Its strengths lie in providing detailed species records and distribution data, facilitating ecological context for organisms, and supporting research on insect pests and native species. Researchers and conservationists benefit from its wealth of information to assess biodiversity impact and pest behavior. While not specialized in pest monitoring, MBD enhances pest management by offering a broader understanding of local ecosystems. This collaborative database stands as a crucial asset in safeguarding Maryland's natural heritage and aiding scientific research. Scientists studying insect pests or conducting research on entomology can use the MBD to access data on insect species' distributions and occurrences. Pest management strategies often require a comprehensive understanding of the local ecosystem. MBD can provide context by offering information on the diversity of species that may interact with or be affected by insect pests.

TABLE 1 Insect DSs frequently used in agriculture applications.

DS name	Availability/Link	Classes/Observation	Number of images	Papers
IP102	Publicly/ https://github.com/xpwu95/IP102	102/Common pest species with a hierarchical taxonomy	75 222	(Ayan et al., 2020), (Butera et al., 2021), (Kasinathan et al., 2021), (Nanni et al., 2022), (Wang et al., 2022), (Wu et al., 2019)
Maryland	Publicly/ https://www.marylandbiodiversity.com/	20 600 species/ Cataloging living things	671 983	(Popescu et al., 2022)
AgriPest	Publicly/ https://github.com/liuliu66/AgriPest	14/Common pest species	49.7 K and 264.7 K annotated	(Wang et al., 2022)
Deng	Publicly/ https://doi.org/10.1016/ j.biosystemseng.2018.02.008	10 species of tea plants insect pests	NA	(Deng et al., 2018) (Teng et al., 2022)
NBAIR	Publicly/ https://www.nbair.res.in/databases National Bureau of Agricultural Insect Resources	40/field crop insect images	NA	(Cardim Ferreira Lima et al., 2020) (Thenmozhi & Srinivasulu Reddy, 2019)
RGBInsect	Publicly/ http://rgbinsect.cn/	10/stored-grained insects	3757	(Li et al., 2019) (Li et al., 2020)
Xie 1	Publicly/ http://www2.ahu.edu.cn/pchen/web/ insectRecognition.htm	24/field crop insect images	60 per species	(Cardim Ferreira Lima et al., 2020) (Xia et al., 2018), (Xie et al., 2015)
Xie 2	Publicly/ https://www.dlearningapp.com/web/ DLFautoinsects.htm	40/field crop insect images	4500	(Cardim Ferreira Lima et al., 2020), (Ayan et al., 2020), (Nanni et al., 2022), (Xie et al., 2018)
MDP2018	Private/Multi-Class Pests Dataset 2018 https://doi.org/10.1109/ACCESS.2019.2909522	16/Insect pests	88 670	(Liu et al., 2019)
LLPD-26	Private/ https://doi.org/10.3389/fpls.2022.810546	26/insect pests	18 585	(Teng et al., 2022)
Pest24	Publicly/ http://aisys.iim.ac.cn/zhibao.html	24/field crop insect images	25 378	(Wang et al., 2020) (Wang et al., 2022)
iDigBio	Publicly/ https://www.idigbio.org/	NA/Biodiversity specimens and resources	NA	(Valan et al., 2019)
Turkey-PlantDataset	Publicly/ https://github.com/mturkoglu23/ PlantDiseaseNet	15/Plant disease and pest images	4 447	(Turkoglu et al., 2022)
CPAF Dataset	Publicly/ https://drive.google.com/drive/folders/ 1GR4S2eqahZrLTmZlPphyfcIX5fkkV36?usp=sharing	20/insect species	73 635	(Wang et al., 2020)

AgriPest introduces a domain-specific benchmark dataset for tiny wild pest detection in agriculture. This dataset contains over 49.7K images and 264.7K annotated pests, making it the largest of its kind. It aims to enhance the application of deep learning in agriculture by providing standardized data for pest detection research. AgriPest also defines sub-datasets, including challenges like pest detection and population counting, and validation subsets for various real-world scenarios. The authors build practical pest monitoring systems based on deep learning detectors and evaluate their performance using AgriPest. This dataset and associated code will be publicly available, facilitating further research in pest detection and precision agriculture.

Crafted to serve as a robust resource for training deep learning models in pest detection, Pest24 is another important DB which offers a vast repository of meticulously annotated images of agricultural pests. This paper addresses the challenges of real-time pest population monitoring in precision agriculture using AI technology. It introduces a large-scale standardized dataset called Pest24, comprising 25,378 annotated images of agricultural pests collected from automatic pest traps and imaging devices. The dataset covers 24 categories of common pests in China. On the other hand, the study applies various advanced deep learning detection methods, such as Faster RCNN, SSD, YOLOv3, and Cascade R-CNN, to detect these pests and achieves promising results for real-time field crop pest monitoring. The authors aim to advance accurate multi-pest monitoring in precision agriculture and provide a valuable object detection benchmark for the machine vision community.

The analysis of Pest24 highlights three key factors influencing pest detection accuracy: relative scale, number of instances, and object adhesion. Due to the scarcity of multi-target pest image big data, Pest24 holds great importance as a resource for advancing

intelligent field crop pest monitoring. Characterized by its large-scale data, small relative object scales, high object similarity, and dense distribution, Pest24 presents unique challenges for deep learning-based object detection methods and is poised to drive progress in pest detection for precision agriculture while serving as a specialized benchmark for the computer vision community. Beyond its application in precision agriculture, Pest24 serves as an invaluable benchmark for the machine vision community, fostering advancements in specialized object detection. Future work aims to expand the dataset with more diverse multi-pest images from various practical.

Xie1 and Xie2 databases were other important resources in creating databases or testing and training the architectures defined in various works (Table 1). Because these datasets are not large some authors have often resorted to augmentation techniques to increase the size of these datasets. The Xie2 dataset also called D0 contains 40 classes of insect pests represented in 4508 RGB images of 200 x 200px resolution.

Although there are several public databases illustrating and grouping various common classes of insect pests, most of the authors used their own datasets in solving the problems specific (Bhoi et al., 2021; Rajeena et al., 2022). Creating proprietary databases for insect pest detection or monitoring using NNs can help improve the accuracy and specificity of pest detection systems, while also providing flexibility and cost-effectiveness (Segalla et al., 2020; Hong et al., 2021). From the point of view of flexibility, creating its own database offers absolute control of the data that is attached to train the NNs for insect pest monitoring. This is about how the data set can be adjusted as needed to meet the changing need for insect pest families and environmental changes that may occur rapidly. Complete control of the specificity of pest populations was discussed in several works to describe the specificity zone (Khanramaki et al., 2021). By creating proprietary databases, specific insect pests can be tailored and described regarding each context and interest in pest recognition and monitoring. This can help ensure that the NNs are able to accurately identify and differentiate between the specific insect pests, rather than simply providing a general detection of any insect in the image (Liu and Wang, 2020; Xu et al., 2022).

It is very important that the data set describes a real context to solve real problems with increased accuracy. What was observed in this regard as part of the present study in relation to the performances obtained by the authors in various works was a tendency to create robust datasets in increasing performances. The larger the database used for training and validating NNs, the higher the accuracy of the created models can be (Knyshov et al., 2021; Liu et al., 2022). The database used is determined by the precise study objectives and the sort of data required. Researchers interested in insect pest recognition, for example, may pick IP102 or Pest24, but those needing ecological context may prefer MBD. AgriPest is appropriate for precision agricultural research. Collectively, these databases help to advance pest detection and agricultural research. When paired with these different datasets, CNNs provide a very effective tool for insect pest study and control. They can help to increase pest detection accuracy, understand pest behavior in ecological contexts, and improve real-time monitoring and control tactics in precision agriculture. Researchers and practitioners may use these datasets to create more effective and efficient pest-related solutions in agriculture.

In another scenario, from a cost point of view, creating own database can be a cost-effective alternative. Test and training data creation solutions can capture data using low-cost methods like phones or digital cameras, which is a pretty good starting point. Where the data set is acquired using drones, high-fidelity cameras, robots, or specialized human resources, the cost of acquiring and creating the reference data set for pest monitoring can increase commensurately with the size and quality of data acquired (Xing et al., 2019; Tian H. et al., 2020; Genaev et al., 2022).

The organization of the data set represents another aspect noted by the authors in the development of models for harmful insect and pest detection in modern agriculture. In general, for training and evaluation using CNNs for pest detection and identification, the dataset division commonly includes training and validation sets or training, validation, and testing sets. The most common ratio observed in the last split was 70% for training, 20% for validation, and 10% for testing (Huang et al., 2022). The other ratio could include 80% for training and 20% for testing (Du et al., 2022; Zhang S. et al., 2022), or 70% with 30% respectively (Ahmad et al., 2022).

Regarding the dataset, the authors also followed techniques like data augmentation (Du et al., 2022; Zhang et al., 2023). Data augmentation in the context of CNNs is the process of producing additional training examples by applying various changes to existing pictures in the training dataset (Albanese et al., 2021). Geometric changes such as random rotation, horizontal and vertical flips, random cropping, and transformations such as brightness modifications or color jitter are examples of frequent transformations used for data augmentation in CNNs (Padmanabhuni and Gera, 2022). Adding random cropping can assist the model in learning to distinguish things that are not centered in the image (Genaev et al., 2022).

For data augmentation, some of the new trends include synthetic data generation to increase the number of samples if the number of representatives of a class is insufficient (Abbas et al., 2021). Using generative models to create synthetic images is one novel method of data augmentation. Augmentation through synthetic data generation is a novel technique of generating new training data using computer algorithms rather than gathering real-world data (Huang et al., 2022). The purpose of this method is to enhance the quantity and variety of the dataset, which can improve the performance of ML models (Divyanth et al., 2022). Synthetic data generation could address issues such as imbalanced datasets, lack of data privacy, and limited data availability (Lu et al., 2019). For the topic of agricultural pests, this can be done in a variety of ways. There are several methods for creating synthetic data for CNNs (Karam et al., 2022), including generative adversarial networks (GANs), deep learning picture synthesis, data augmentation, and data interpolation (Padmanabhuni and Gera, 2022). Conditional GAN was used by (Abbas et al., 2021) to generate synthetic images for tomato pests and to improve the performances. Another performance improvement was noted by (Divyanth et al., 2022) by creating an artificially generated dataset using GAN.

TABLE 2 Modality of image acquisition.

Image acquisition vector	Agricultural crop/images	Performances	Papers
Human operators (with camera or smartphone)	Oil palm/8000 Eggplant/NA NA/563 Fruits/365	ACC: 89% R^2 = 0.85 to 0.95 ACC: 94.3% F1 Score: 83.8%	(Ahmad et al., 2021) (Bereciartua-Pérez et al., 2022) (Cochero et al., 2022) (Genaev et al., 2022)
Pheromone-based traps and cameras	Apple orchard/8000 Apple/300 Vegetables/1789 Forest/50 Greenhouse/400	ACC: 97.9% training ACC: 97% training, 93% validation F1 Score: 83.8% ACC: 95.3% - 97.89% F1 Score: 90% - 92%	(Albanese et al., 2021) (Brunelli et al., 2020) (Guo et al., 2021) (Hong et al., 2021) (Rustia et al., 2020)
UAV	Forest/4710 Rice/NA Weeds, Potato, Grapes/600 NA/500 Maize/5691 Eucalyptus/4930	PRE: 70% ACC: 80% ACC: 90% PRE: 85%, F1 Score: 55% ACC: 97.59% - 98.77% ACC: 98.45%	(Aota et al., 2021) (Bhoi et al., 2021) (Bouroubi et al., 2018) (De Cesaro Júnior et al., 2022) (Dai et al., 2021) (Dos Santos et al., 2022)
Terrestrial vehicles and camera	Pomelo orchard/510	ACC: 95.83%	(Partel et al., 2019) (Tian G. et al., 2020)

For the testing phase, the acquisition of digital images from real contexts can be noted. This was pursued by the authors to test the NN architectures they created and optimized against the real contexts, using pest images in the field (Brunelli et al., 2020). For modern agriculture, there are some ways of acquiring digital images, using various systems and techniques (Terentev et al., 2022). The present study identified four important directions that describe image acquisition vectors and were grouped and described in Table 2. Based on the analyzed references, the performances obtained using the created databases were also noted. In this sense, satisfactory results are observed, and at the same time, it is important to note that the methods of image acquisition are done in an optimized framework and represent a strong point attached to the research areas in this field. For the modern agricultural area, the acquisition of data for the creation of models and automatic solutions in pest monitoring represents an extensive process that can include several resources (Nanni et al., 2022).

Table 2 summarizes the most common data gathering methods, with UAVs and pheromone traps emerging as the most popular options. This section will evaluate the benefits and drawbacks of different techniques. The integration of ML and deep learning DL for automated data processing, with a special focus on remote sensing and sensory data for complete area mapping, is an emerging research field. It is worth mentioning that remote sensing, as investigated by (Stefas et al., 2016; Ahmad et al., 2021), has several applications in fields such as agriculture and forestry.

Unmanned Aerial Vehicles (UAVs) are gaining remarkable traction across diverse domains, with agriculture and environmental monitoring being prominent beneficiaries. One of their vital applications lies in the realm of pest detection and management within agricultural crops Mu et al., 2018. UAVs offer versatile data acquisition methods, including high-resolution imagery and sensory capabilities (Tian H. et al., 2020; Cochero et al., 2022). Equipped with high-resolution cameras, UAVs excel at capturing images and videos of crops, facilitating the identification of insect pests (Tian H. et al., 2020; Cochero et al., 2022). Subsequently, these images can undergo automated pest detection using ML algorithms (Preti et al., 2021). Moreover, UAVs can be equipped with sensors for detecting specific chemicals in the air or on plant surfaces, thus enabling pest identification, as well as treatment efficacy monitoring (Velusamy et al., 2022). To combat identified pests, certain UAVs are equipped with precision sprayers, targeting affected areas with minimal chemical usage and environmental impact (Iost Filho et al., 2019; Li C. et al., 2022). Thermal cameras mounted on UAVs provide valuable temperature data, aiding in pinpointing stressed or pest-infested crop areas due to temperature differences (Yuan & Choi, 2021). UAVs also use multispectral cameras, such as infrared and hyperspectral imaging, in addition to typical RGB images, which considerably improves the accuracy of pest detection models (Terentev et al., 2022). Another current technique employs lidar sensors to collect high-resolution 3D pictures of agricultural fields, allowing for the identification of pest-infested areas (Dong et al., 2018; López-Granados et al., 2019). Lidar imaging also provides information about crop dimensions, growth patterns, and prospective yield (Johansen et al, 2018; Ampatzidis et al., 2020).

Nonetheless, there are several drawbacks to the UAV-based strategy. UAVs, in general, have limited payload capacity, restricting their ability to carry large amounts of equipment and sensors. Furthermore, UAV flight durations are limited, often ranging from 20 to 30 minutes depending on the type and payload. As a result, covering large regions may demand numerous flights, which can be both time-consuming and costly (Dong et al., 2020). While UAVs excel in collecting high-resolution photographs of crops and insect pests, image analysis algorithms' accuracy may be limited, necessitating professional analysis. Furthermore, the use of UAVs for data collecting is vulnerable to weather and legal limitations. These variables might limit the capacity to collect insect pest data during certain seasons or geographical locations. Many nations have tight UAV laws that include flying limitations as well as criteria for permissible equipment and sensors (Csillik et al., 2018).

Another way of data acquisition for insect monitoring in modern agriculture is based on pheromone traps (Table 1). Data acquisition using pheromone traps is a useful tool for monitoring and controlling insect pests in agriculture and forestry. Pheromone traps are placed in strategic locations throughout a crop. The number and placement depend on the type of insect pest being targeted and the size of the area being monitored. Pheromone traps need to be checked regularly to ensure that they are working properly. Digital cameras can be attached to the pheromone traps to capture images of the trapped insects. The traps should be monitored regularly, typically every 1-2 weeks. During each monitoring visit, the traps are checked for trapped insects, and the digital cameras are checked to ensure they are functioning correctly.

While pheromone traps can be an effective tool for monitoring and managing insect pests, there are some disadvantages to their use. They are only effective against insect pests that are attracted to specific pheromones (Cardim Ferreira Lima et al., 2020). On the other hand, their effectiveness is limited to the area in which they are placed (Toscano-Miranda et al., 2022). Pheromone traps can attract not only the target insect pest but also non-target species that are attracted to the same pheromones. Additionally, pheromone traps can give an incomplete or inaccurate representation of the population of insect pests. This is because some individuals of the pest species may not respond to the pheromone lure or may be located outside the trapping area. This can lead to incorrect decisions about pest management strategies.

In agriculture, insect pest identification and monitoring are key parts of precision farming because they have a direct influence on crop production, quality, and overall agricultural sustainability. CNNs, in conjunction with specialist databases, provide significant promise for tackling the issues connected with insect pest control. Deep learning architectures, object identification, categorization and taxonomy, and real-time monitoring may be essential elements of neural networks applied to these datasets, as demonstrated in the selected research. With the use of CNNs and specialized datasets, insect pest identification and monitoring have entered a new age. This synergy has the potential to lead to more accurate, timely, and environmentally conscious pest management solutions. Continuous research, multidisciplinary cooperation, and an emphasis on practical application are required to fully achieve this promise. As technology advances, the future of insect pest identification and monitoring in agriculture remains bright, with the potential to greatly contribute to global food security and sustainable agriculture practices.

2.3 Neural networks used in insect detection, segmentation, and classification

Automated monitoring systems use sensors and cameras to detect and identify insect pests (Amorim et al., 2019). These systems can be connected to the internet, allowing farmers to receive real-time information about pest populations. There are many solutions and methodologies based on image processing, DL, and NNs. CNNs are particularly well suited for tasks involving the detection of small objects, such as insects, within an image. In this scenario, CNNs are a powerful tool for pest detection and have been shown to achieve high accuracy in many applications. One key advantage of CNNs for pest detection is their ability to handle complex images. For example, a CNN can be trained to detect pests in images that contain multiple objects, different backgrounds, and varying lighting conditions (Fang et al., 2020). Additionally, CNNs can be trained on a large dataset of images, which can help improve the accuracy of the model. Another advantage of CNNs for monitoring crops for pest detection is their real-time ability (Ayan et al., 2020). On the other hand, one of the most significant advancements in this field is the development of transfer learning, where a pre-trained CNN model is fine-tuned on a smaller dataset of pest images. Some of the most used NNs for insect and pest detection and classification are presented in Figure 4.

VGG Net (Visual Geometry Group) (Simonyan & Zisserman, 2014) is a key architecture used in insect detection and monitoring, especially VGG-16 and VGG-19. The architecture is widely used in computer vision applications such as object detection and image segmentation (Popescu et al., 2022). The architecture for VGG-16 (Ramadhan & Baykara, 2022) is shown in Figure 4A, and it was the most used for insect detection and classification tasks. The convolutional layers are responsible for extracting features from the input image, while the pooling layers reduce the spatial dimensions of the feature maps to reduce computation time. The fully connected layers are used to classify the features extracted by the convolutional and pooling layers. The most used, VGG-16 model has a total of 16 layers and the VGG-19 has 19 layers being a modified version of VGG-16 with the addition of the new three convolutional layers.

Residual Network (ResNet) is another CNN family used in insect monitoring for modern agriculture. ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 are variants of the CNN architecture that was introduced in 2015 by researchers at Microsoft (He et al., 2016). The key innovation of ResNet is the use of "residual connections," or shortcut connections, that allow the network to learn identity mapping and make it easier to train very deep networks. This is shown in Figure 4B as a residual block example. According to the investigated papers, ResNet-50 was the most used for insect detection and classification tasks, and the basic architecture is shown in Figure 4C.

The R-CNN (Region-based CNN) architecture is a type of object detection model that uses a combination of CNNs and region proposal algorithms to detect objects within an image (Ren et al., 2015) and was also used for insect monitoring. It is a two-stage process that first generates a set of region proposals and then uses a CNN to classify and refine the proposals. The first stage of the R-CNN architecture is the region proposal algorithm, which generates a set of regions or "proposals" that may contain an object of interest. These regions are then passed to the second stage of the R-CNN architecture, which is the CNN. This is used to classify and refine the regions generated by the region proposal algorithm and it is done by extracting features from each region and passing them through a series of convolutional and fully connected layers. In this context, another architecture often used for insect detection tasks was Faster R-CNN. This is a type of object detection

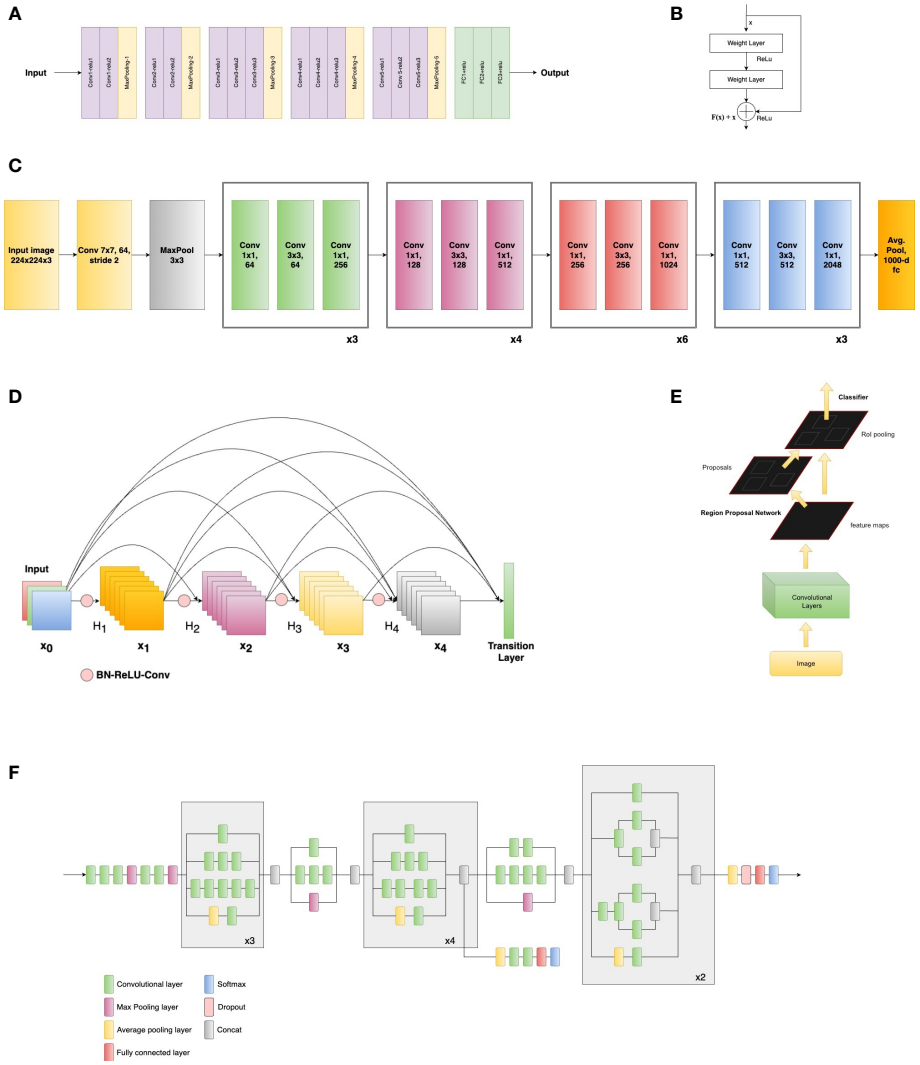


FIGURE 4
Examples of neural networks used: **(A)** VGG-16 architecture (adapted from Simonyan and Zisserman 2014, **(B)** Residual block example (adapted from He et al., 2016), **(C)** Example architecture of ResNet-50 (adapted from He et al., 2016), **(D)** Dense block example (adapted from Huang et al., 2017), **(E)** Faster RCNN architecture (adapted from Ren et al., 2015), **(F)** Inception V3 architecture (adapted from Szegedy et al., 2016).

model that uses a CNN to extract features from an image and then uses a region proposal network (RPN) to propose regions that may contain objects. The feature extractor typically uses a pre-trained CNN, such as VGG or ResNet, to extract features from the input image. The main advantage of Faster R-CNN over other object detection models is its efficiency, as it shares computation between the RPN and the classifier. The Faster R-CNN architecture, adapted from (Ren et al., 2015) is presented in Figure 4E.

The Inception CNN architecture (Szegedy et al., 2015) is also representative of insect classification and detection. This deep CNN architecture utilizes a combination of convolutional, pooling, and inception modules to efficiently learn hierarchical representations of visual data. The novel aspect is that it includes a series of components named Inception modules that apply a combination of convolutional and pooling layers at different scales, allowing the network to efficiently capture and learn both the high-level and low-level features of the image. This review highlighted that the most used architecture from this family, for insect detection and classification was the InceptionV3 (Szegedy et al., 2016). Following the structure and features presented previously, the basic scheme of Inception V3 can be viewed in Figure 4F (adapted from Szegedy et al., 2016).

Dense Convolutional Network (DenseNet) is another CNN family used for insect detection and classification. This neural network architecture is characterized by dense layers Huang et al., 2020. Each layer is connected to every other layer in the network (Huang et al., 2017). This creates a dense network of connections, which allows for a more efficient flow of information and a greater capacity for learning. A dense block is shown in Figure 4D, adapted from (Huang et al., 2017). One of the main advantages of DenseNet architecture is its ability to effectively handle large amounts of data and complex patterns (Huang et al., 2017).

YOLO (You Only Look Once) is another state-of-the-art family that is widely used in the modern agricultural sector for real-time

insect detection and monitoring. YOLO (Redmon et al., 2016) is an object detection algorithm that uses a single stage to perform object detection. Unlike other object detection algorithms that rely on region proposals, YOLO uses a grid of cells to divide the image into smaller regions and predicts the object class and location for each cell. The algorithm is trained on large datasets, such as the COCO (Common Objects in Context) or ImageNet, and has been designed to be fast and accurate. Different variants from this family were used: YOLOv2 (Redmon & Farhadi, 2017), YOLOv3 (Redmon & Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), YOLOv5s, YOLOv5m, and YOLOv5l (Ultralytics, 2020).

A synthetic presentation of NNs used for insect and pest detection and classification in agricultural applications is given in Table 3. Based on the information from Table 3, the graph in Figure 5 describes the evolution over the last three years of the most used neural networks for insect monitoring in modern agriculture.

This study primarily centers its focus on exploring emerging trends in CNNs for insect pest detection and monitoring through the innovative application of new combinations, while also acknowledging classic CNN models as reference points. This approach aligns with the prevalent practice in the field, where most studies strive to strike a balance between pioneering CNN architectures and established, foundational models. This dual perspective, embracing innovation while respecting tradition, mirrors a common practice observed in the contemporary studies within the deep learning community. Researchers understand that leveraging the strengths of

TABLE 3 CNN used in insect and pest detection.

CNN family/ References	Representatives/ configuration	Function	Performances	Papers
AlexNet 5	AlexNet	Classification	ACC: 80.3% - 91.31.%, F1 score: 96%	[(Khanramaki et al., 2021), (Li et al., 2019), (Malathi and Gopinath, 2021), (Xu et al., 2022), (Divyanth et al., 2022)
CapsNet 2	CapsNet/modified	Classification	ACC: 82.4%, PRE: 75.41%	(Xu et al., 2022), (Zhang S. et al., 2022)
CNN 8	CNN	Classification	ACC: 91.5% - 98,6% F1 score: 95%	(Chodey & Shariff, 2021), (Hossain et al., 2019), (Espinoza et al., 2016), (Kasinathan et al., 2021), (Sharma et al., 2020), (Singh et al., 2021)
	BPNN	Classification	ACC: 91%	(Zhu et al., 2020)
DenseNet 8	DenseNet 121	Detection and classification	ACC: 88.06% - 99.1%	(Abbas et al., 2021), (Sanghavi et al., 2022), (Zhang & Chen, 2020), (Shi et al., 2020)
	DenseNet 169	Detection	mAP: 92.3%	(Butera et al., 2021)
	DenseNet 201	Detection and classification	ACC: 79.01%, 95.52%	(Nanni et al., 2022), (Singh et al., 2021)
	Weakly DenseNet-16	Classification	ACC: 93.42%	(Xing et al., 2019)
EfficientNet 4	EfficientNet	Detection	ACC: 97.89% - 99.1%	(Dai et al., 2021), (Sanghavi et al., 2022), (Takimoto et al., 2021)
	EfficientNet B0	Detection	ACC: 94.25%	(Nanni et al., 2022)
EfficientDet 1	EfficientDet D0	Detection	ACC: 95.3% - 97.9%	(Hong et al., 2021)
GoogLeNet 1	GoogLeNet with Inception modules	Classification	ACC: 91.02%	(Malathi and Gopinath, 2021)
Inception 10	Inception v3	Classification	ACC: 75.3% - 99.04% mAP: 71%	(Ayan et al., 2020), (Fang et al., 2020), (Hansen et al., 2019), (Rajeena et al., 2022), (Sanghavi et al., 2022), (Singh et al., 2021), (Wang et al., 2020), (Liu et al., 2022)
	Inception ResNetv2	Detection	ACC: 91.14%	(Khanramaki et al., 2021), (Singh et al., 2021)
LeNet 3	LeNet5	Classification	ACC: 93.1% - 96.1%, PRE: 94%	(Albanese et al., 2021), (Ding & Taylor, 2016) (Segalla et al., 2020)
MobileNet 10	MobileNet	Detection and classification	ACC: 82.10% - 97.39%	(Ayan et al., 2020), (Singh et al., 2021), (Xing et al., 2019)

(Continued)

TABLE 3 Continued

CNN family/ References	Representatives/ configuration	Function	Performances	Papers
	MobileNetv2	Detection	ACC: 81.32% - 96.29%	(Albanese et al., 2021), (Hong et al., 2021), (Nanni et al., 2022), (Rajeena et al., 2022), (Xing et al., 2019), (Zhang & Chen, 2020)
	MobileNetv3	Detection	mAP: 92.66%	(Butera et al., 2021)
	Optimized MobileNet	Classification	ACC: 95.04%	(Rimal et al., 2022)
NASNet/1	NASNetMobile	Classification	ACC: 73.46%	(Singh et al., 2021)
Perceptron/1	Multi-layer perceptron	Detection	ACC: 98.45%	(Dos Santos et al., 2022)
R-CNN/13	Cascade R-CNN	Detection	mAP: 70.83%	(Dos Santos et al., 2022)
	Faster R-CNN	Detection and classification	ACC: 60,2% - 99% F1: 85.5% - 99.5% mAP: 65.58% - 89.1%	(Ahmad et al., 2021), (Alsanea et al., 2022), (Butera et al., 2021), (Du et al., 2022), (Guo et al., 2021), (Hong et al., 2021), (Li et al., 2019), (Liu et al., 2019), (Wang et al., 2022), (Shi et al., 2020)
	Mask R-CNN	Detection and segmentation	PRE: 85%	(De Cesaro Júnior et al., 2022)
	MSR-RCNN/ResNet-50 backbone	Detection	mAP: 67.4%	(Teng et al., 2022)
RegNet 1	RegNet	Detection	ACC: 98.07%	(Dai et al., 2021)
ResNet 31	ResNet/modified	Detection	ACC: 95.83%	(Tian G. et al., 2020),
	ResNet 18/modified	Detection	ACC: 60.3%	(Roosjen et al., 2020)
	ResNet 34	Detection	ACC: 94.3%, 91.2%	(Cochero et al., 2022), (Malathi and Gopinath, 2021)
	ResNet 50	Classification	ACC: 43.99% - 99.04% F1 score: 55% - 92.6% mAP: 74,24% - 88.5%	(Ayan et al., 2020), (Bereciartua-Pérez et al., 2022), (Butera et al., 2021), (De Cesaro Júnior et al., 2022), (Fang et al., 2020), (Dai et al., 2021), (Khanramaki et al., 2021), (Li et al., 2019), (Liu et al., 2019), (Liu et al., 2022), (Malathi and Gopinath, 2021), (Nanni et al., 2022), (Rajeena et al., 2022), (Sanghavi et al., 2022), (Wang et al., 2020), (Wang et al., 2022), (Xu et al., 2022)
	ResNet 53	Detection	mAP: 77.29%	(Lv et al., 2022)
	ResNet 101	Detection	mAP: 85.53% - 99.5%	(Hong et al., 2021), (Li et al., 2019), (Liu et al., 2019), (Lv et al., 2022), (Wang et al., 2022), (Zhang & Chen, 2020), (Shi et al., 2020)
	ResNet 152	Detection	ACC: 96.31%	(Zhang & Chen, 2020)
	ResNeXt-50	Classification	ACC: 86.5%	(Li C. et al., 2022)
RetinaNet 3	RetinaNet	Detection	mAP: 65.03% - 94.77%	(Li et al., 2020), (Wang et al., 2022)
	RetinaNet50	Detection	mAP: 86.40%	(Hong et al., 2021)
SqueezeNet 1	SqueezeNet	Classification	ACC: 94.02%	(Ayan et al., 2020)
ShuffleNet 2	ShuffleNet v1	Classification	ACC: 83.58%	(Xing et al., 2019)
	ShuffleNet v2	Classification	ACC: 83.58%	(Xing et al., 2019)
SSD 3	SSD	Detection	PRE: 70%	(Aota et al., 2021)
	SSD with MobileNetv2	Detection	mAP: 84.54%	(Hong et al., 2021)
	SSD/with VGG-16 and ResNet-50	Detection	mAP: 63.38%	(Wang et al., 2022)
VGG 23	VGG16/modified	Classification	ACC: 67% - 97.9% R^2 = 0.85 to 0.95	(Albanese et al., 2021), (Ayan et al., 2020), (Bereciartua-Pérez et al., 2022), (Khanramaki et al., 2021), (Knyshov et al., 2021), (Kusrini et al., 2021), (Li et al., 2019), (Nazri et al., 2018), (Rajeena et al., 2022), (Sanghavi et al., 2022), (Singh

(Continued)

TABLE 3 Continued

CNN family/ References	Representatives/ configuration	Function	Performances	Papers
				et al., 2021), (Valan et al., 2019), (Wang et al., 2020), (Wu et al., 2019), (Xing et al., 2019), (Zhang S. et al., 2022)
	VGG16/modified	Detection	F1:95.25%, mAP: 78.20%, PRE: 99%	(Segalla et al., 2020), (Wang et al., 2022), (Shi et al., 2020)
	VGG19	Classification	ACC: 74.07% - 99.02%	(Ayan et al., 2020), (Fang et al., 2020), (Rajeena et al., 2022), (Singh et al., 2021)
	VGG19/improved + RPN	Detection	mAP: 89.22%	(Xia et al., 2018)
Xception 4	Xception	Classification	ACC: 74.07% - 97.98 PRE: 77%	(Ayan et al., 2020), (Fang et al., 2020), (Rajeena et al., 2022), (Singh et al., 2021), (Kuzuhara et al., 2020)
YOLO 14	YOLO	Detection	ACC: 88.06% - 92.50%	(Shi et al., 2020), (Zhong et al., 2018)
	YOLOv3/improved	Detection	PRE: 77%, mAP: 77.29%, F1: 87% - 90%	(Kuzuhara et al., 2020), (Liu and Wang, 2020) (Lv et al., 2022), (Partel et al., 2019), (Rustia et al., 2020)
	Tiny-YOLOv3	Detection	F1 Score: 90% - 92%	(Rustia et al., 2020)
	YOLOv4	Detection	F1 Score: 55% - 83.8%	(Genaev et al., 2022), (Takimoto et al., 2021)
	YOLOv5	Detection	ACC: 98.45%, mAP: 77.0% -99.2%	(Bereciartua-Pérez et al., 2022), (Dos Santos et al., 2022), (Zhang Y. et al., 2022), (Zhang et al., 2023)
ZF Net 2	ZF Net	Detection	mAP: 88.5%, 75.46%	(Li et al., 2019), (Liu et al., 2019)

both new and classic CNN models can yield comprehensive insights and solutions, ultimately driving the field forward.

Regarding key trends and advancements, CNNs continued to be a popular choice for image-based insect pest detection. Researchers were developing and fine-tuning CNN architectures to achieve higher accuracy in recognizing and classifying pests from images. Transfer learning techniques were becoming increasingly important in this domain. Researchers were pre-training CNN models on large

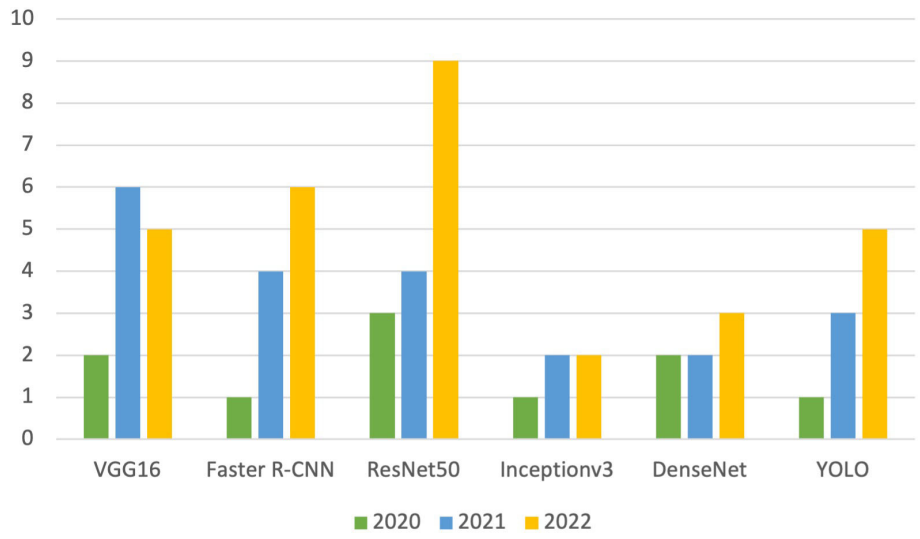


FIGURE 5
The graph of the evolution in the last three years of the most used neural networks.

datasets and then fine-tuning them for insect pest detection tasks. This approach helped in achieving better results even with limited labeled data for specific pests. For object detection and localization, object detection models like Faster R-CNN, YOLO and SSD were adapted for insect pest monitoring. These models not only classified pests but also provided bounding box coordinates, which is crucial for precise pest localization. As a new trend, researchers were experimenting with advanced data augmentation techniques to improve model robustness. Techniques like GANs were used to create synthetic pest images to augment the training dataset. Next, focusing on network architectures, capsule networks, which aim to address the limitations of traditional CNNs in handling hierarchical features, have been explored for insect pest recognition (Xu et al., 2022; Zhang S. et al., 2022). They can capture the spatial hierarchies of pest body parts for improved classification. Some researchers have proposed hybrid architectures that combine the strengths of CNNs for image processing and recurrent neural networks (RNNs) for sequential data processing. This is particularly useful when tracking pests' movements over time (Butera et al., 2021; Alsanea et al., 2022; Du et al., 2022). To make pest detection systems more transparent and interpretable, explainable AI in architectures techniques have been integrated into neural network architectures. This allows users to understand why a particular pest detection decision was made. Researchers often choose or design architectures based on the unique characteristics and challenges of the pests they are targeting and the monitoring environment. Advancements in neural network architectures for insect pest detection and monitoring are ongoing, so staying up to date with the latest research papers and developments in the field is essential for the most current insights.

2.4 Performance indicators

Looking at the area of impact and innovation, the new trends stand out with high-performance indices in relation to the area of pest identification. Attaching these was done to create a comparison area. Since the research was based on deep learning models, the indicators most used as evaluation methods of these models were highlighted as part of this study, being represented by accuracy, precision, sensitivity, specificity, F1 score, Jaccard index, mean average precision (mAP), and sometimes R2. Names and calculation formulas are attached in Table 4. The most used performance indicators were mAP, accuracy, and F1 score. Representative indices were also extracted from the creation of the confusion matrix (Ahmad et al., 2022) where the values for TP – True Positive, TN – True Negative, FP – False Positive, and FN – False Negative are indicated.

2.5 Software used

This study underlines the need of tracking the software used in NNs (Table 5). This is especially important given the fast developments in NNs and the advent of new software and approaches. Nevertheless, various software programs may yield somewhat different results due to differences in implementation and optimization strategies. Knowing what software was used allows others to replicate and validate the results. This is especially significant for improving the area and expanding on previous studies. Furthermore, knowing the software utilized helps enhance collaboration in the fields of NNs and PA. It allows academics to share code and data, enabling the flow of ideas and speeding up research and development.

As can be observed from Table 5, Tensorflow in combination with Keras is the most popular choice for software development in pest detection or identification systems using CNNs (Fang et al., 2020). The second popular way of software implementation, showing increasingly high and modern adoption, is represented by PyTorch with the attached torch and torch-vision libraries.

TensorFlow is an open-source software library developed by Google for building and training ML models (Abadi et al., 2016). It is a popular and powerful DL framework that provides a wide range of tools and APIs for building and training models (Wang et al., 2022). TensorFlow is a library for numerical computation that is particularly well-suited to the computation of large-scale linear algebra operations, which are a common component of many ML algorithms. It provides a wide range of tools for building and training DL models, including CNNs and recurrent NNs. It also includes support for distributed training and deployment on different hardware platforms. For the task of detection of harmful insects and pests in modern agriculture, it was a popular choice (Table 5).

Keras is an open-source software library written in Python that provides a high-level interface for building and training DL models (Chollet et al., 2015). It is built on top of other popular DL

TABLE 4 Performance indicators used in the review.

Indicator	Formula	Indicator	Formula
Accuracy (ACC)	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	Sensitivity (SEN)	$SEN = \frac{TP}{TP + FN}$
Precision (PRE)	$PRE = \frac{TP}{TP + FP}$	Specificity (SPE)	$SPE = \frac{TN}{TN + FP}$
F1 Score (F1)	$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$	Jaccard index (j)	$j = \frac{TP}{TP + FN + FP}$
Mean Average Precision (mAP)	$mAP = \frac{1}{N} * \sum_{i=1}^{N} AP_i$	R^2	$R^2(y, \hat{y}) = 1 - \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$ $R^2 = \frac{Explained\ variation}{Total\ Variation}$

TABLE 5 Software used.

Software	Description	Link	Papers
PyTorch	■ An open-source machine learning framework ■ Based on Python programming language and Torch library	https://pytorch.org/	(Cochero et al., 2022), (Du et al., 2022), (Dai et al., 2021), (Guo et al., 2021), (Huang et al., 2022), (Lv et al., 2022), (Wang et al., 2022), (Zhang Y. et al., 2022), (Zhang et al., 2023), (Shi et al., 2020)
TensorFlow	■ An end-to-end open-source machine learning platform	https://www.tensorflow.org/	(Ahmad et al., 2021), (Alsanea et al., 2022), (Bereciartua-Pérez et al., 2022), (De Cesaro Júnior et al., 2022), (Fang et al., 2020), (Guo et al., 2021), (Hossain et al., 2019), (Karam et al., 2022), (Knyshov et al., 2021), (Rajeena et al., 2022), (Rimal et al., 2022), (Sharma et al., 2020), (Takimoto et al., 2021), (Valan et al., 2019), (Wang et al., 2020), (Wang et al., 2022), (Wu et al., 2019)
Keras	■ High-level, modular, and flexible open-source neural network library and API based on Python programming language	https://keras.io/	(Ayan et al., 2020), (Fang et al., 2020), (Hossain et al., 2019), (Karam et al., 2022), (Knyshov et al., 2021), (Lu et al., 2019)
Imagga Cloud API	■ Image recognition API as a service	https://imagga.com /	(Bhoi et al., 2021)
Fastai	■ Deep learning library	https://www.fast.ai/	(Cochero et al., 2022)
MathWorks Matlab	■ Programming and numeric platform designed for engineers and scientists	https://www.mathworks.com/products/matlab.html	(Divyanth et al., 2022), (Nagar and Sharma, 2021)

frameworks, including TensorFlow, and provides a simple and intuitive API for defining and training models. Keras was designed with the goal of making DL accessible to a wider audience, including researchers, students, and developers with limited ML experience.

For insect monitoring tasks, another software used was PyTorch. Based on the analyzed papers, a strong adoption of the framework in pest detection tasks is observed in the last three years, especially in 2022. Facebook (actual Meta) team created PyTorch as an open-source machine learning framework (Paszke et al., 2019). It is a well-known and sophisticated DL framework that offers a variety of tools and APIs for developing and training ML models. Torch is a scientific computing framework that enables efficient tensor operations and automated differentiation. PyTorch is built on top of the Torch library and improves these capabilities by including a dynamic computational graph, allowing for more flexible and intuitive model creation, and debugging. PyTorch includes a variety of tools and APIs for developing and training DL models such as CNNs, recurrent NNs, and others.

The MATLAB programming and numerical computing platform do not have the same characteristics as the libraries and deep learning frameworks like Tensorflow + Keras or PyTorch, based on the performance and flexibility associated with the Python programming language in which they are implemented. MATLAB (matrix laboratory) is a programming environment and a programming language used primarily for numerical computing and scientific computing (MathWorks Matlab 22). MathWorks MATLAB provides a wide range of built-in functions and tools specifically designed for image processing and computer vision applications (Nagar and Sharma, 2021). MATLAB's Image Processing Toolbox provides a comprehensive set of tools for image analysis, filtering, segmentation, feature extraction, and object recognition (Divyanth et al., 2022). The toolbox includes functions for common image processing tasks such as image smoothing, noise reduction, edge detection, and morphological operations. MATLAB also provides support for deep learning and machine learning, which can be used for image classification and object recognition tasks.

In this sense, although there are various software solutions, the Python programming language remains a solid basis to build such deep-learning systems based on artificial neural networks in the detection, identification or even monitoring of insect pest. Cloud computing services capable of providing modules, APIs or even software platforms as a service in the development of deep-learning solutions for pest detection have also been noted. The main characteristic in their case is represented by the availability and flexibility in accessing these types of cloud resources, being therefore part of the new trends.

Another software used for insect monitoring in precision and modern agriculture was Fastai. It is a high-level open-source DL library built on top of PyTorch (Howard & Gugger, 2020). It is designed to make it easier to train state-of-the-art DL models with as little code as possible. The library provides a simple and consistent API for quickly training deep NNs on a wide range of tasks, such as image classification, object detection, text classification, and natural language processing. One of the unique features of Fastai is its approach to transfer learning, which involves leveraging pre-trained models and fine-tuning them for specific tasks (Cochero et al., 2022).

Another modern software that was used for insect monitoring in agriculture was Imagga Cloud API which is a cloud-based image recognition platform that provides a suite of APIs for developers to build image-related applications (Imagga, 2020). Imagga API was used for rice pest detection (Bhoi et al., 2021), integrating IoT and

UAV systems. The Imagga Cloud API provides a range of image analysis and recognition services, including image tagging, content-based image search, color extraction, cropping, and ML algorithms that can identify objects, scenes, colors, and other attributes within an image.

3 New trends in harmful insect and pest detection

Regarding insect monitoring for detection, classification, and even segmentation there are several modern approaches to train and validate a computer system for pest monitoring tasks using AI (Zhang, 2022). CNNs are frequently utilized in this procedure because they are particularly well-suited to image recognition tasks (Teng et al., 2022). Over time a well-trained system can be used to identify pests quickly and accurately in real-world scenarios and images, enabling farmers, growers, and other stakeholders to take action to address any issues quickly and effectively (Aota et al., 2021). The modification of networks in relation to specific detection or identification tasks has evolved over time and new ways of implementation and development have emerged to meet these needs.

Training and validating individual networks are the first starting point. Modifying existing architectures through various mathematical or structural methods is a common practice to increase the robustness of such a system. By increasing the number of training images and fine-tuning the network's parameters, the accuracy of pest identification using NNs may be enhanced (Xia et al., 2018). This procedure is done multiple times until the system achieves a satisfactory level of accuracy (Butera et al., 2021). On the other hand, approaches to modify the base structure and new optimization methods are addressed to satisfy the same final need, to increase the accuracy and precision of a system in relation to representative areas for pest detection and monitoring. Models with notable results starting from the basic structures of state-of-the-art networks by applying transfer learning techniques, increasing dimensions, and implementing custom optimizations were developed (Abbas et al., 2021). The research papers adopted transfer learning applied to several public databases or similar research datasets noted and described in previous chapters. Oftentimes, research has involved the creation of proprietary and private databases that are focused on the needs of each area under investigation.

Multinetwork-based systems are new trends for insect monitoring and detection. The most representative ones are based on custom ensemble models. The use of ensembles of NNs and innovative modified architectures can improve the accuracy of pest detection. A CNN ensemble is a mixture of several CNN models that results in a stronger, more accurate prediction model. The aim of an ensemble is to use the strengths of many models to compensate for the shortcomings of a single model (Xu et al., 2021). The final decision of an ensemble of CNNs is derived by fusion of the predictions of separate CNN models, often by majority voting or weighted averaging. An ensemble's diversity of models decreases the problem of overfitting, resulting in greater accuracy and precision. Once trained, the outputs of the individual CNN models are combined to form a final prediction. The idea is that by combining the predictions of multiple models, the overall accuracy and reliability of the system can be improved, and the risk of false positives or false negatives can be reduced. One of the main advantages of using a CNN ensemble for insect pest detection is that it can improve the ability of the system to generalize to new images or environments that may be different from the training dataset. By using multiple models with different strengths and weaknesses, the ensemble can be more robust to variations in lighting, background, or other factors that may affect the appearance of the insects in the images. The majority voting ensembles, weighted average ensembles, and multinetwork ensembles using a variety of CNNs backbones are the most popular and most adopted in the case of pest detection and identification. Some examples of ensemble models of NNs are presented in Figure 6.

Fusion by weighted sum rule and combinations based on different topologies and various Adam optimization were used (Nanni et al., 2022) for the detection of several insect pests attached to each database. The performance of the presented work was noted and compared for different datasets. CNN architectures are trained using various optimization functions, including some novel Adam variations, and then fused. The system is described in Figure 6A (inspired by Nanni et al., 2022). The paper compared some of the state-of-the-art architectures for pest classification: ResNet50, GoogleNet, DenseNet201, and EfficientNetB0. Some other models were added and used for their speed and efficiency on mobile devices: ShuffleNet and MobileNetV2. In terms of optimization, Adam variants like diffGrad was used to calculate a scaling factor in the learning rate.

Another strategy using transfer learning, fine-tuning, and model ensemble was proposed in (Ayan et al., 2020). D0, SMALL, and IP102 datasets were again selected and used to train, validate, and test the accuracy rates of the proposed models. The study involved modifying and re-training seven pre-trained CNN models using transfer learning and fine-tuning on a 40-class dataset.

The top three models (Inception-V3, Xception, and MobileNet) were ensembled using the sum of maximum probabilities and weighted voting with weights determined by a genetic algorithm to create two ensembled models: SMPEnsemble and GAEnsemble (Ayan et al., 2020). Pre-trained models on ImageNet were implemented and the proposed model of insect classification ensemble methodology can be seen in Figure 6B. The paper highlights that deep networks with different architectures can have varying generalization capabilities when trained on the same dataset. This is because different models can extract different features from the data based on their architecture. Therefore, it is important to consider the model architecture when selecting the best-performing model for a given task. Adopting the suitable CNN architecture for insect pest detection helps increase the detection system's accuracy and efficiency. It may be able to construct models that are more adapted to certain pest detection tasks by using the inherent capabilities of each architecture because various insect pests might have diverse physical characteristics that necessitate specific detection procedures. Certain pests, for example, may have

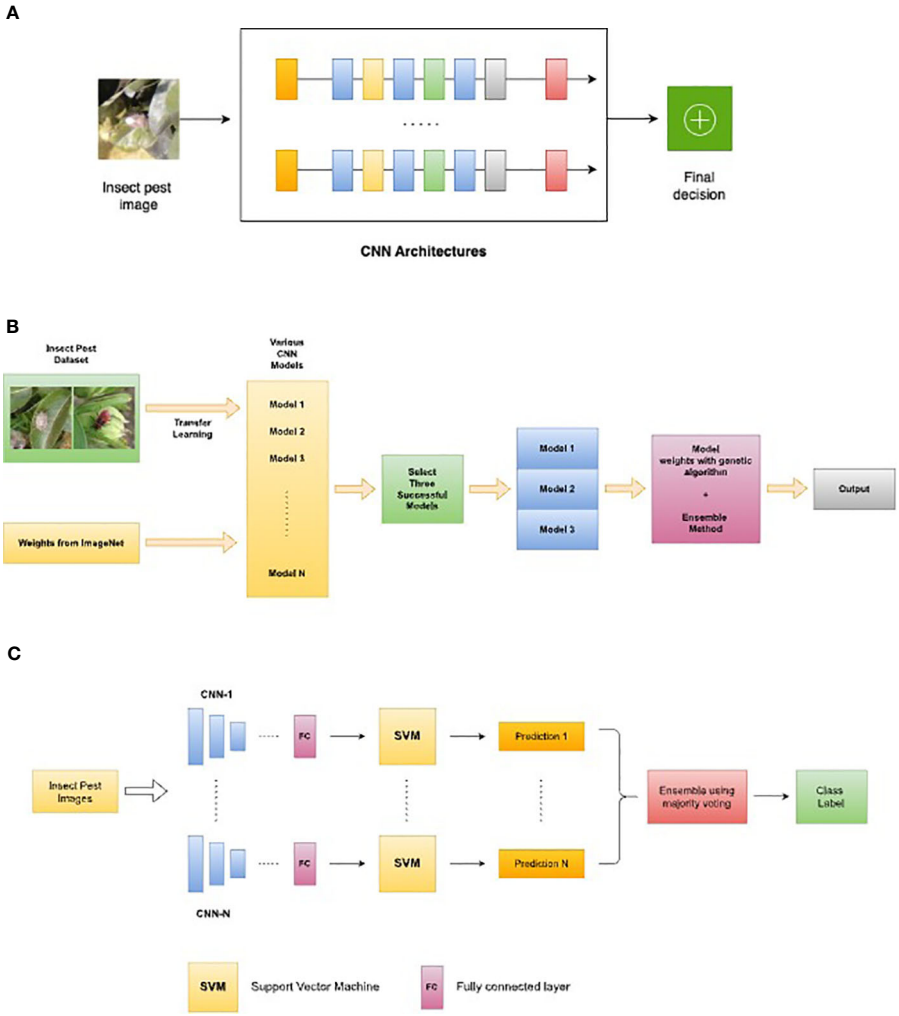


FIGURE 6
Examples of multi-network-based systems as a new trend: **(A)** Adapted system architecture for the ensemble model proposed in (Nanni et al., 2022) for insect pest detection, **(B)** Adapted insect classification ensemble methodology proposed in (Ayan et al., 2020), **(C)** Adapted majority voting ensemble model for pest classification proposed in (Turkoglu et al., 2022).

a distinguishing pattern of spots or stripes on their body, while others may have distinct antennae or wings. As a result, it is critical to carefully pick the CNN architecture to be employed for insect pest identification. Because of its capacity to extract data at different scales, an architecture like Inception-V3 may be more suited for pests with complex traits. MobileNet, on the other hand, can be more suited for simpler pests or resource-constrained applications because of its lightweight design.

Another ensemble model was proposed in (Turkoglu et al., 2022) using a majority voting method Figure 6C. Feature concatenation and SVM (Support Vector Machine) classifier was also implemented at the core of the proposed system which used six state-of-the-art networks for pest classification and plant disease classification.

The tendency of researchers to modify the NN backbone was also observed. Modifying the backbone of a pre-trained NN for a given task is a typical approach in deep learning (Li Z. et al., 2022). Many cutting-edge models are constructed on modified backbones of pre-trained NNs (Kuzuhara et al., 2020; Liu et al., 2022). Although this aspect does not define a new area, there are some directions that can be highlighted in relation to the idea of modifying the backbone of a neural network. In this sense, specific modifications of the backbone and the impact on the performance of the network can describe novel and innovative research (Table 6). Modifying the NN backbone can have a significant impact on its performance. For instance, changing the number of layers in the backbone can affect the depth of the network and its ability to learn more complex features. Adding or removing layers can also affect the number of parameters in the network, which can impact its overall computational efficiency. Additionally, changing the architecture of the backbone can impact the type of features extracted from the input data. A defining example of the area of innovation brought by modifying the backbone of a model can be researched in the study (Butera et al., 2021). The authors (Butera et al., 2021) used Faster R-CNN, SSD, and RetinaNet. Backbone used was based on several models such as VGG, ResNet, DenseNet, and MobileNet, adapted for the task of insect pest detection in real-world scenarios. Additionally, the

TABLE 6 CNN ensemble architectures and backbone modifications.

NN Used	Novelty	Combination/ Description	Function/ Application	Perfor-mances	Papers, year
AlexNet, GoogleNet, DenseNet201	CNN Ensemble	■ Majority voting fusion	Classification/Apple pest and disease classification in a real-time application	ACC: 96.1% -99.2%	(Turkoglu et al., 2020)
AlexNet, VGG16, ResNet-50, InceptionResNet V2	CNN Ensemble	■Fusion by correlation coefficient comparison ■Majority voting	Classification/Citrus pest	F1 Score: 0.935	(Khanramaki et al., 2021)
EfficientNetB0, GoogleNet, ResNet-50, MobileNetV2, ShuffleNet, DenseNet201	CNN Ensemble	■Fusion by weighted sum rule ■Combination based on different topologies ■ Various Adam optimization	Detection/ Insect pest	ACC: 95.52% (SMALL), 74.11% (IP102), 99.81% (D0)	(Nanni et al., 2022)
AlexNet, ResNet 18, 50 and 101, DenseNet201, GoogleNet	CNN Ensemble	■ Fusion by averaging ■ Majority voting ■Integrating SVM classifier	Detection and classification/ Plant disease and pest	ACC: 97,56%, 96.83%	(Turkoglu et al., 2022)
Inception-V3, ResNet-50, Xception, VGG-16, VGG-19, MobileNet	CNN Ensemble	■Fusion by majority voting ■Four-stage classification methodology	Classification/Insect	ACC: 98%	(Ayan et al., 2020)
FasterRCNN, MobileNetV3	Backbone modification	■FasterRCNN with MobileNetV3 backbone	Detection/Insect pest	mAP: 92.66%	(Butera et al., 2021)
YOLO-v4-tiny, CSPDarknet53	Backbone modification	■YOLO v4-tiny with CSPDarknet53-tiny backbone	Detection/Insect pest	F1: 0.838	(Genaev et al., 2022)
R-CNN ResNet50	Backbone modification	■Novel MSR-RCNN model with ResNet-50 backbone	Detection/ Multi-class pest	mAP: 67.4%	(Teng et al., 2022)
SSD, RetinaNet, FCOS, R-CNN, FPN, Cascade R-CNN	Backbone modification	■SSD with VGG16 as backbone ■ResNet 50 for object detetion	Detection/Insect pest		(Wang et al., 2022)
RetinaNet	Backbone modification	■RetinaNet with feature pyramid network backbone	Detection/multi-scale insect detector	mAP: 94.77%	(Li et al., 2020)
VGG, ZFNet, ResNet 50 - 101, Faster R-CNN	Backbone modification	■Deep CNN fused with CSA	Detection and classification/ Multi-class pests	mAP: 75.46%	(Liu et al., 2019)
YOLOv3, Xception	Two-stage detector	■Two-stage detection using YOLOv3 and Xception	Detection and classification/ Small insect pests	PRE: 77%	(Kuzuhara et al., 2020)
Inception, ResNet50	Two-stage detector and backbone modification	Two-stage CNN solution integrating GaFPN and GAM	Detection and classification/ Small insect pests	mAP: 71%	(Liu et al., 2022)
YOLOv5, ShuffleNetv2	Model combination	ShuffleNetv2-YOLOv5-Lite-E improved detection model for edge devices	Detection/ Tea culture pest	mAP: 97,43%	(Zhang et al., 2023)
GhostNet, YOLOv5	Model combination	YOLOv5-GhostNet combination for embedding devices	Detection/ Orchard pest	mAP: 99%	(Zhang Y. et al., 2022)

impact of the transfer learning technique on the models used for accuracy and inference time was also studied. The authors noted that a model based on Faster R-CNN with MobileNetv3 is a strong point for insect pest detection.

The YOLOv4-tiny architecture with CSPDarknet53-tiny as the backbone was used to train a pest fly detection model using a dataset of insects of interest (Genaev et al., 2022). The network consists of Backbone, Neck, and three recurring blocks including Convolution, CBL, and CSP blocks. The CSP block structure utilizes a feature pyramid network to divide the input feature map into two parts. This structure reduces computational complexity while maintaining accuracy in object detection. Using YOLOv4-tiny allowed for the development of a fly recognition method that can be implemented as a modern mobile application.

A network for robust pest detection, with emphasis on small-size, multi-scale, and high-similarity pests was proposed by the authors (Teng et al., 2022). The proposed pest detection network used two customized core designs: a multi-scale super-resolution (MSR) feature enhancement module and a Soft-IoU (SI) mechanism. The MSR module developed enhances feature expression ability for small-size, multi-scale, and high-similarity pests, while the SI mechanism emphasizes position-based detection

requirements. The MSR-RCNN is more suitable for pest detection tasks and includes a ResNet50 backbone and a feature full fusion mechanism to improve multi-scale pest detection. A feature full weighting mechanism was added and optimizes the detection performance of similar pests from two dimensions (depth and location). The implemented MSR module includes a super-resolution component used to obtain a six-layer feature map for recognizing small-sized objects. Additionally, the full feature fusion mechanism is used to integrate all features at once for recognizing multi-scale objects. On the other hand, in this study, a large-scale pest dataset of trap images was developed (LLDP-26). It can be observed that the changes made to the existing models and backbones bring considerable improvements in performance, enabling the solution of pest identification problems from digital images and outperforming existing state-of-the-art models and techniques.

A two-stage detection and identification method for small insect pests using CNN was proposed in (Kuzuhara et al., 2020). The authors used YOLOv3 as an object detection model, which is a popular deep learning model for object detection. A region proposal network (RPN) to help identify the regions of the image that contain the pest is used. After identifying the regions of interest, the proposed method performs pest classification using the Xception model (Chollet, 2017), which is a deep CNN that has been shown to achieve high accuracy in image classification tasks. The authors further improved the classification accuracy by using a data augmentation method based on image processing, which helped to generate more training examples by applying transformations to the original images. One of the strengths of this two-stage detection method is that it can handle the challenges posed by small insect pests, which are difficult to detect using traditional object detection methods due to their small size and low contrast. This method shows a good way in achieving high accuracy in detecting and identifying small insect pests, which can help improve pest management in agriculture.

Regarding the new trends, the authors in (Zhang et al., 2023) proposed an improved detection model based on ShuffleNetv2 and YOLOv5. This paper presents a target detection model based on the ShuffleNetv2-YOLOv5-Lite-E method, which substitutes the Focus layer with the ShuffleNetv2 algorithm. It also reduces the model size by pruning the YOLOv5 head at the neck layer. The suggested model is more robust and lightweight, and it may enhance detection efficiency while maintaining the recognition rate.

Combining YOLOv5 and GhostNet (Zhang Y. et al., 2022) and using a custom pest dataset allowed the method to achieve a higher mAP with the same number of epochs. In this case, the usage of GhostNet in YOLOv5 can be described as a new trend. GhostNet is a lightweight neural network architecture proposed in 2020 (Han et al., 2020) for usage in edge devices. The utilization of Ghost modules, which replace the usual convolutional layers in a NN, is the core characteristic of GhostNet. Ghost modules have a primary and secondary path. The primary path is a normal convolutional layer, but the secondary path has fewer channels and is used to simulate the behavior of the primary path. GhostNet may achieve equivalent precision to bigger networks by employing Ghost modules but with fewer parameters and lower processing cost. This makes it appropriate for deployment on low-power devices with limited processing resources. For the proposed model, authors noted 1.5% higher mAP than the original YOLOv5, with up to three times fewer parameters and the same or less detection time. With this architecture, the mAP obtained by the authors was about 99%.

Table 6 synthesizes the novelty and performances of CNN ensemble architectures.

4 Applications

In real applications, data classification and analyzing huge volumes of data are time-consuming. To increase efficiency, the final strategy is to create and optimize ML and DL models to estimate and create powerful systems for understanding features, patterns, and complex, big amounts of data (Csillik et al., 2018; Abayomi-Alli et al., 2021). The focus area is to train models to find optimal parameters, auto-adjust values, and adapt to a robust architecture generated and optimized step by step over several epochs of training with dataset capture (Nanni et al., 2022; Wang et al., 2022). For agricultural areas, ML is widely used to automate time-consuming, labor-intensive tasks and to collect essential information having at the core mathematical models, computational resources, and infrastructure with high performance and standards. As part of this study, we can note this as a new trend in precision agriculture. Proposed works show

TABLE 7 Applications.

Application	Papers
Harmful insect detection	(Albanese et al., 2021), (Alsanea et al., 2022), (Ayan et al., 2020), (Butera et al., 2021), (Cochero et al., 2022), (Genaev et al., 2022), (Guo et al., 2021), (Hansen et al., 2019), (Hong et al., 2021), (Hossain et al., 2019), (Iost Filho et al., 2022), (Espinoza et al., 2016), (Kasinathan et al., 2021), (Khanramaki et al., 2021), (Knyshov et al., 2021), (Li et al., 2019), (Li et al., 2020), (Li C. et al., 2022), (Liu et al., 2019), (Liu and Wang, 2020), (Lv et al., 2022), (Malathi and Gopinath, 2021), (Nagar and Sharma, 2021), (Nanni et al., 2022), (Rajeena et al., 2022), (Rimal et al., 2022), (Rustia et al., 2020), (Sanghavi et al., 2022), (Teng et al., 2022), (Valan et al., 2019), (Wang et al., 2020), (Wang et al., 2022), (Xia et al., 2018), (Zhang & Chen, 2020), (Shi et al., 2020)
Infected crops by insects	(Bereciartua-Pérez et al., 2022), (Bhoi et al., 2021), (Fang et al., 2020), (Espinoza et al., 2016), (Kusrini et al., 2021), (Nazri et al., 2018), (Sharma et al., 2020), (Singh et al., 2021), (Tian G. et al., 2020), (Turkoglu et al., 2020), (Turkoglu et al., 2022), (Wu et al., 2019), (Xing et al., 2019), (Xu et al., 2022), (Zhang S. et al., 2022), (Zhu et al., 2020)
Crop monitoring	(Ahmad et al., 2021), (Aota et al., 2021), (Bouroubi et al., 2018), (Brunelli et al., 2020), (De Cesaro Júnior et al., 2022), (Ding & Taylor, 2016), (Dai et al., 2021), (Partel et al., 2019), (Dos Santos et al., 2022), (Takimoto et al., 2021), (Zhong et al., 2018)

considerable results and note the popularity of AI in general. The applicability aspect of using these defined systems brings to the forefront a series of advantages and development areas. As can be seen from Table 7, most of the papers are focused on the following main applications: harmful insect detection, identification of infected crops, and crop monitoring.

4.1 Harmful insect detection

CNNs have become increasingly popular in image-processing applications for modern agriculture following their ability to identify insects and features in images. According to this study, one of the applications of CNNs in the field of modern and precision agriculture is harmful insect detection. The identification of harmful insects is crucial for the protection of crops and the prevention of plant diseases (Lv et al., 2022). CNNs can be an effective tool for harmful insect detection in images (Guo et al., 2021; Alsanea et al., 2022). By training the network on a large and diverse dataset, CNN can learn to identify a wide range of harmful insects. However, the issues of class imbalance and transferability need to be addressed to ensure that CNN performs well in real-world applications. For effective detection of harmful insects, the first step is to collect and label a dataset of digital images containing both harmful and non-harmful insects (Cochero et al., 2022; Wang et al., 2022). In this case, the dataset should be large and diverse to ensure the great performance of the CNN model and to ensure that the CNN can learn to recognize a wide range of harmful insects. Because this detection uses CNN models that learn different features of an image through convolutional operations, the second step is the preprocess the images in the dataset created to ensure that they are in a format that can be fed into the CNN. This may involve resizing the images, converting them to grayscale, or normalizing the pixel values (Alsanea et al., 2022; Zhang Y. et al., 2022). Following this scenario, the next step is to train the chosen CNN model using the dataset prepared (Malathi and Gopinath, 2021; Nagar and Sharma, 2021; Liu et al., 2022). This involves feeding the network the labeled images and adjusting the weights of the neurons through backpropagation to minimize the error between the predicted and actual labels. Transfer learning applied on a custom insect pest dataset can be used and hyperparameter tuning to speed up the process in this topic. Related to this aspect, most of the papers analyzed for this study include such methodology (Ayan et al., 2020). After the CNN was trained, it can be used to classify new images of insects as either harmful or non-harmful. To do this, the new image is fed into the CNN, and the output is a probability score indicating the likelihood that the insect in the image is harmful. A threshold value can be set, and if the probability score is above this value, the insect is classified as harmful.

One of the main challenges that were identified in applications for harmful insect detection using CNNs is the issue of class imbalance (Du et al., 2022). Harmful insects may be rare in the dataset, which can lead to the CNN being biased towards non-harmful insects. To overcome this, techniques such as over-sampling or under-sampling can be used to balance the dataset. Another challenge identified is the issue of transferability. CNNs trained on one dataset may not perform well on a different dataset due to differences in the types of insects or the background images. To address this, transfer learning can be used, which involves using a pre-trained CNN as a starting point and fine-tuning the network on the new dataset, as mentioned earlier (Butera et al., 2021; Li W. et al., 2022; Popkov et al., 2022).

4.2 Infected crops by insects

CNNs are a powerful tool for identifying insect-infected crops. They can be trained to learn patterns and features in images that are indicative of insect damage and provide predictions on whether the crops are healthy or infected (Turkoglu et al., 2022; Zhang S. et al., 2022). The use of CNNs in agriculture can improve crop yields and help farmers prevent and manage insect infestations more effectively (Espinoza et al., 2016; Sharma et al., 2020; Bereciartua-Pérez et al., 2022). Infected crops by insects can have a significant impact on the agricultural industry, leading to the loss of crops and revenue (Xu et al., 2022). With the increasing advancements in computer vision, for modern agriculture, our study highlights that the CNNs became an effective tool for identifying and detecting insect infestations in crops.

CNNs are commonly utilized in applications such as image classification, object identification, and segmentation. CNNs may be taught to recognize patterns and characteristics in images that are indicative of insect damage in the context of recognizing insect infestations in crops. Similarly, to the insect detection tasks discussed, a huge collection of images of healthy and infected crops must be developed for applications used to target diseased crops. The images are then annotated with whether the crops are healthy or sick, as well as the species of bug inflicting the harm. The CNN models are then trained by giving them tagged images, allowing them to understand the patterns and characteristics associated with insect-infested crops.

On the other hand, the CNN model can also provide information about the type of insect causing the damage, enabling farmers to take appropriate measures to prevent further damage. In addition to identifying insect-infected crops, CNNs can also be used for segmentation tasks (Zhang & Chen, 2020). Segmentation involves dividing an image into different regions or objects. In the context of identifying insect infestations, segmentation can be used to identify and evaluate the areas of the crop that are infected. This can provide more detailed information to farmers and enable them to target their treatment strategies more effectively.

4.3 Crop monitoring

The third area of applications using models based on DL, respectively on CNNs, is crop monitoring. This area of application has a major impact when considering pest insect populations and managing the effects of their presence. In this sense, there have been several studies that have included this direction of development. Crop monitoring refers to the process

of keeping track of the growth, development, and health of crops. Crop monitoring can be done using a variety of methods, including satellite imagery, drone imagery, ground-based sensors, and visual inspections. However, the traditional methods of crop monitoring can be time-consuming, expensive, and require a significant number of resources. With the advent of AI and ML, the use of CNNs for crop monitoring has become increasingly popular. In crop monitoring, CNNs can be used to analyze images of crops and provide insights into their growth, development, and health. The process of crop monitoring using CNNs typically involves several steps including data collection, data preprocessing, training NNs, and evaluating performances in a specific area of interest (Dai et al., 2021). Applications of crop monitoring using CNNs have a wide range of applications in modern agriculture, including disease and pest detection or even yield estimation (Zhong et al., 2018; Tian G. et al., 2020). CNNs can be used to detect the presence of diseases in crops by analyzing the images of the leaves and other parts of the plant. This can help farmers to take timely action to prevent the spread of diseases and minimize crop losses. On the other hand, this process can be automated by introducing real-time monitoring modules, based on hardware systems and software modules optimized for mobile platforms, used in the field. Important to note, crop monitoring using CNNs has the potential to revolutionize agriculture by providing farmers with real-time insights into the growth, development, and health of their crops. CNNs can analyze images of crops quickly, accurately, and at a fraction of the cost of traditional methods (Vanegas et al., 2018). By using CNNs for crop monitoring, farmers can make informed decisions about crop management, minimize losses due to meteorological conditions, diseases, and pests, and optimize their yields.

5 Discussion

This review paper points out several features in relation to the areas of massive pest detection, classification, and recognition in various crops. The research method plans to highlight the advantages and disadvantages as well as the new trends of CNNs and the application of image processing within these aspects of PA. On the other hand, this study highlights the use of innovative approaches and techniques, such as DL, transfer learning, active learning, ensembles of CNNs, and multi-scale feature fusion, for pest detection and classification from digital images. Overall, this study is focused on insect monitoring including real environment, NNs, and new trends.

Harmful insects and pest detection present a series of challenges that researchers tend to study more and more and solve the problems that arise. Analyzing the research extracted from established databases, we noticed the wide interest in recent years based on the topic of modern and precision agriculture. As it was presented in the previous chapters, the databases chosen for

TABLE 8 Recent review/survey papers on similar topics.

Paper/ year	Description	Period	Refe- rences	Our differences
(Abade et al., 2021)	■ Systematic plant disease review ■ CNN for crop disease recognition – trends and gaps. ■ State of the art through systematic review used – StArt Tool	2010-2019	121	■ Focused on insects including real environment. ■ Focused on new trends (including 2022). ■ New investigated methods for review papers (PRISMA). ■ More references.
(De Cesaro Júnior & Rieder, 2020)	■ Different approaches like CNN and other image classifiers for insect or diseased plants detection from images.	2015 - 2019	57	■ Focused on insects including real environment. ■ Focused on new trends (including 2022). ■ New investigated methods for review papers (PRISMA). ■ More references.
(Cardim Ferreira Lima et al., 2020)	■ Identification and monitoring of insect pests using automatic traps. ■ Using infrared sensors, audio sensors, and image-based classification	2007 - 2020	77	■ Focused on more insects including real environment ■ Focused on image processing ■ Focused on neural networks ■ Focused on new trends (including 2022). ■ More references. ■ New investigated methods for review papers (PRISMA)
(Iost Filho et al., 2019)	■ Using drones in pest management to obtain canopy reflectance data of arthropod infested plants.	1998 - 2018	319	■ Focused on insects including real environment

(Continued)

TABLE 8 Continued

Paper/ year	Description	Period	Refe- rences	Our differences
				■ Focused on new trends (including 2022). ■ Focused on neural networks ■ New investigated methods for review papers (PRISMA)
(Ghosh et al., 2021)	■ Strategies and future trends on molecular and automated pest identification (thrips) for rapid and high throughput diagnosis.	2001-2020	253	■ Focused on insects including real environment ■ Focused on new trends (including 2022). ■ New investigated methods for review papers (PRISMA)
(Kumar & Kukreja, 2022	■ Systematic review on wheat disease prediction models Kitchenham investigation method (Kitchenham et al., 2010)	1997-2021	102	■ Focused on insects including real environment ■ Focused on new trends (including 2022). More references. New investigated methods for review papers (PRISMA)
(Liu & Wang, 2021)	■ Plant disease and pest detection based on deep learning ■ Aspects of classification, detection and segmentation networks are discussed	2014-2020	108	■ Focused on new trends (including 2022). ■ Focused on insects including real environment ■ More references. ■ New investigated methods for review papers (PRISMA)
(Preti et al., 2021)	■ Insect pest management using camera-equipped traps and smart traps ■ Remote sensing and electronics for long-distance pest monitoring ■ Automatic detection and analysis for insect detection and counting ■ Automatic traps usage benefits	1980-2020	75	■ Focused on new trends (including 2022). ■ Focused on image processing ■ Focused on neural networks ■ More references. ■ New investigated methods for review papers (PRISMA)
(Toscano-Miranda et al., 2022)	■ Insect pests and disease detection in cotton cultures using ML and IoT ■ Focused on remote sensing and AI techniques ■ Trends for smart agriculture ■ Kitchenham investigation method [Kit 10]	2014-2021	100	■ Focused on new trends (including 2022). ■ Focused on insects including real environment ■ Focused on image processing ■ Focused on neural networks ■ More references ■ New investigated methods for review papers (PRISMA)

extracting the papers of this review study were Web of Science, IEEE, and Scopus. Most of the papers chosen for analysis were extracted from the Web of Science database one of the most widely used citation databases in the world. Research on new trends and impact information has been placed in the 2020-2022 range. For the review topic, similar articles were extracted and compared. Their analysis is presented in Table 8, where the differences compared to this presented review and the area of contributions were also noted. Based on the analysis, good quality information was highlighted, and it was observed that the interest in the detection of harmful insects and pests in modern agriculture using image processing and NNs is quite pronounced.

Training, validation, and testing modalities are important points in the research of architectures that automate processes in modern agriculture. In the initial steps, acquiring the data set and organizing it is extremely important. Most papers reviewed for this study highlighted the impact of a robust dataset, adding images taken from real contexts. It has been observed that for the modern area, techniques such as data augmentation and synthetic data generation play an important role to diversify the data set. These implications solve the problems where the training and validation data set is small and for multi-class pest detection tasks it can solve the class imbalance problem. A modern use case was noted by the authors in (Karam et al., 2022) developing a web app for synthetic data generation using DC-GANs, for agricultural pest detection (whiteflies). The study illustrates how employing GAN in the pipeline can improve the model's capacity to generalize and hence improve the accuracy of detected bounding boxes.

Image processing is another important step to note. Due to the acquisition of digital images from real contexts, the presence of

insects at the image level presents some aspects that have a negative impact on the training and evaluation of the model that receives this data as input. These aspects are represented by the relatively small size of the insects, artifacts at the image level, and the context in which they are illustrated: complex background, various types of occlusions (branches, leaves), the presence of insects in large numbers, and small object detection. Image processing aids in the preprocessing and enhancement of input pictures, hence boosting the accuracy and performance of CNN models. Images collected from various sources, such as digital cameras or drones, may have differences in lighting, background noise, and other artifacts that might affect the accuracy of insect detection. As a result, image processing techniques like filtering, segmentation, and normalization can aid in the removal of noise and artifacts, the improvement of contrast, and the highlighting of areas of interest in pictures. Image processing may also aid in the extraction and selection of useful aspects from digital images, such as color, texture, and shape, that are significant to insect pest identification. The CNN models can learn to discriminate between various insect species and effectively categorize them by finding and extracting these traits, even in complex situations.

To synthesize the findings, the present review paper highlighted the fact that the combination of CNN architectures, as well as the modification of existing architectures through various techniques, bring to the fore notable performances in terms of accuracy. According to the previously mentioned characteristics related to the novelty in the combination of convolutional neural networks and the problems in the detection of harmful insects of interest, a series of studies of interest were identified with various presented methods and integrating databases illustrating real contexts.

Starting in 2019, the authors (Liu et al., 2019) presented a DL approach named PestNet. It was highlighted that multi-class pest detection is a crucial step for effective pest management in modern agriculture. In this work, PestNet includes a novel channel-spatial attention module, a region proposal network, and a position-sensitive score map (PSSM). A newly collected large-scale pest image dataset named MPD2018 was proposed to evaluate the PestNet model achieving 75.46% mAP on 16 pest classes, outperforming other state-of-the-art methods.

Following Pest24 paper and database, to evaluate multi-pest detection performance, the dataset described is divided into training, validation, and test sets, with four state-of-the-art object detection methods employed. YOLOv3 achieves the highest mAP of 63.54% and an impressive AP of 98.6% for individual pests under optimal parameters. A 3-fold cross-validation experiment confirms similar results. The paper examines various factors affecting detection performance, highlighting the significant impact of relative scale on AP while indicating that color discrepancy has negligible influence.

Authors (Wang et al., 2022) also proposed a DL model, this time for the recognition and counting of apple pests. The MPest-RCNN named model achieved mAP and F1-Score values of 99.11% and 99.50%, evaluated using an original dataset of three typical pests in apple orchards. The paper presents a new Faster R-CNN structure based on the ResNet101 feature extractor and a novel CNN structure with small anchors to extract features, therefore boosting recognition accuracy for small pests.

Hunger Games search-based deep convolutional neural network (HGS-DCNN) model for crop pest image classification was proposed (Sanghavi et al., 2022), adding a new convolutional layer to decrease parameter redundancy. Pre-processing and augmentation, followed by pest categorization, are the two steps of the model proposed. Pre-processing makes use of a novel adaptive cascaded filter (ACF) in conjunction with decision-based median filtering (DMF) and guided image filtering techniques. The proposed model outperformed existing pre-trained architectures such as ResNet50, EfficientNet, Dense Net, Inceptionv3, and VGG-16 in terms of accuracy, precision, F1-score, sensitivity, and specificity, with values of 99.1%, 97.80%, 97.80%, 97.82%, and 99.43%, respectively.

In the area of precision agriculture, the advent of new-generation AI technology has ushered in a promising era of real-time pest population monitoring. CNNs have exhibited amazing performance in insect pest identification and categorization as part of deep learning approaches. Their capacity to learn detailed characteristics from large-scale visual data permits accurate recognition, even when inter-class variances are small. Factors like as dataset size, model design, and data quality can all have an impact on CNN performance. It is still difficult to provide robustness against intra-class volatility and data imbalance. Ongoing research in pest identification and monitoring enhances CNNs' capabilities. Collaboration among agricultural, entomology, computer vision, and machine learning professionals enables transdisciplinary solutions.

6 Conclusions

Following this study, the use of new trends in deep learning has the potential to revolutionize the field of pest monitoring and significantly improve pest management in agricultural sector. Algorithms such CNNss have shown great promise in accurately identifying and classifying pests in digital images with high precision and accuracy rates. Currently, CNNs have become a potent tool in identifying crops that are infected with insects. Researchers have developed ensemble techniques where multiple CNN models are combined to achieve better performance. This technique is becoming increasingly popular in the field of pest identification due to its effectiveness in handling complex datasets and the ability to capture diverse features of insects. Optimizing existing models for identifying harmful insects by modifying their architectures specifically for this topic represents another approach with a strong innovative impact. For modern and precision agriculture or integrated pest management, farmers can enhance their treatment approaches by utilizing applications like insect detection for harmful insects, identifying crop infections caused by insects, or monitoring crop growth, which can offer them comprehensive insights and allow them to precisely target their treatments.

Author contributions

DP: Writing – original draft, Writing – review & editing, Conceptualization. AD: Writing – original draft, Writing – review & editing, Methodology. LI: Writing – original draft, Writing – review & editing, Formal Analysis, Investigation. NA: Formal Analysis, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by HALY.ID project. HALY.ID is part of ERA-NET Co-fund ICT-AGRI-FOOD, with funding provided by national sources [Funding agency UEFISCDI, project number 202/2020, within PNCDI III] and co-funding by the European Union's Horizon 2020 research and innovation program, Grant Agreement number 862665 ERA-NET ICT-AGRI-FOOD (HALY-ID 862671).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abade, A., Ferreira, P. A., and Vidal, F. B. (2021). Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* 185, 106125. doi: 10.1016/j.compag.2021.106125

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI*. 265–283.

Abayomi-Alli, O. O., Damaševičius, R., Misra, S., and Maskeliūnas, R. (2021). Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Syst.* 38 (7), e12746. doi: 10.1111/exsy.12746

Abbas, A., Jain, S., Gour, M., and Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* 187, 106279. doi: 10.1016/j.compag.2021.106279

Ahmad, M. N., Mohamed Shariff, A. R., and Moslim, R. (2018). Monitoring insect pest infestation *via* different spectroscopic techniques. *Appl. Spectrosc. Rev.* 53 (10), 836–853. doi: 10.1080/05704928.2018.1445094

Ahmad, M. N., Shariff, A. R. M., Aris, I., and Abdul Halin, I. (2021). A four stage image processing algorithm for detecting and counting of bagworm, metisa plana walker (Lepidoptera: psychidae). *Agriculture* 11, 1265. doi: 10.3390/agriculture11121265

Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., et al. (2022). Deep learning based detector YOLOv5 for identifying insect pests. *Appl. Sci.* 12, 10167. doi: 10.3390/app121910167

Albanese, A., Nardello, M., and Brunelli, D. (2021). *Automated pest detection with DNN on the edge for precision agriculture* (IEEE Journal on Emerging and Selected Topics in Circuits and Systems).

Alsanea, M., Habib, S., Khan, N., Alsharekh, M. F., Islam, M., and Khan, S. (2022). A deep-learning model for real-time red palm weevil detection and localization. *J. Imaging* 8, 170.

Amorim, W. P., Tetila, E. C., Pistori, H., and Papa, J. P. (2019). Semi-supervised learning with convolutional neural networks for UAV images automatic recognition. *Comput. Electron. Agric.* 164, 104932. doi: 10.1016/j.compag.2019.104932

Ampatzidis, Y., Partel, V., and Costa, L. (2020). Agroview: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence. *Comput. Electron. Agric.* 174, 105457. doi: 10.1016/j.compag.2020.105457

Aota, T., Ashizawa, K., Mori, H., Toda, M., and Chiba, S. (2021). Detection of Anolis carolinensis using drone images and a deep neural network: an effective tool for controlling invasive species. *Biol. Invasions* 23, 1321–1327. doi: 10.1007/s10530-020-02434-y

Apolo-Apolo, O. E., Pérez-Ruiz, M., Martínez-Guanter, J., and Valente, João. (2020). A cloud-based environment for generating yield estimation maps from apple orchards using UAV imagery and a deep learning technique. *Front. Plant Sci.* 11, 16–31. doi: 10.3389/fpls.2020.01086

(2022) (Natick, Massachusetts).

Ayan, E., Erbay, H., and Varçın, F. (2020). Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Comput. Electron. Agric.* 179, 105809. doi: 10.1016/j.compag.2020.105809

Bereciartua-Pérez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., and Eggers, T. (2022). Insect counting through deep learning-based density maps estimation. *Comput. Electron. Agric.* 197, 106933. doi: 10.1016/j.compag.2022.106933

Bhoi, S. K., Jena, K. K., Panda, S. K., Long, H. V., Kumar, R., Subbulakshmi, P., et al. (2021). An Internet of Things assisted Unmanned Aerial Vehicle based artificial intelligence model for rice pest detection. *Microprocessors Microsystems* 80, 103607. doi: 10.1016/j.micpro.2020.103607

Bochkovskiy, A., Wang, C., and Liao, H. M. (2020). *YOLOv4: optimal speed and accuracy of object detection* (ArXiv).

Bouroubi, Y., Bugnet, P., Nguyen-Xuan, T., Gosselin, C., Bélec, C., Longchamps, L., et al. (2018). "Pest detection on UAV imagery using a deep convolutional neural network," in *14th International Conference on Precision Agriculture*.

Bouroubi, Y., Magarey, R., and Jess, L. (2022). Integrated pest management data for regulation, research, and education: crop profiles and pest management strategic plans. *J. Integrated Pest Manage.* 13 (1), 13.

Brunelli, D., Polonelli, T., and Benini, L. (2020). "Ultra-low energy pest detection for smart agriculture," in *2020 IEEE Sensors*. 1–4.

Butera, L., Ferrante, A., Jermini, M., Prevostini, M., and Alippi, C. (2021). Precise agriculture: effective deep learning strategies to detect pest insects. *IEEE-CAA J. Automatica Sin.* 9 (2), 246–258. doi: 10.1109/JAS.2021.1004317

Cardim Ferreira Lima, M., Damascena de Almeida Leandro, M. E., Valero, C., Pereira Coronel, L. C., and Gonçalves Bazzo, C. O. (2020). Automatic detection and monitoring of insect pests—A review. *Agriculture* 10, 161. doi: 10.3390/agriculture10050161

Chodey, M., and Shariff, C. (2021). Neural network-based pest detection with K-means segmentation: impact of improved dragonfly algorithm. *J. Inf. Knowledge Manage.* 20, 2150040. doi: 10.1142/S0219649221500404

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1251–1258.

Chollet, F., et al. (2015).

Cochero, J., Pattori, L., Balsalobre, A., Ceccarelli, S., and Marti, G. (2022). A convolutional neural network to recognize Chagas disease vectors using mobile phone images. *Ecol. Inf.* 68, 101587. doi: 10.1016/j.ecoinf.2022.101587

Csillik, O., Cherbini, J., Johnson, R., Lyons, A., and Kelly, M. (2018). dentification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones* 2 (4), 39. doi: 10.3390/drones2040039

Dai, F., Wang, F., Yang, D., Lin, S., Chen, X., Lan, Y., et al. (2021). Detection method of citrus psyllids with field high-definition camera based on improved cascade region-based convolution neural networks. *Front. Plant Sci.* 12.

Damos, P. (2015). Modular structure of web-based decision support systems for integrated pest management. *A review. Agron. Sustain. Dev.* 35, 1347–1372. doi: 10.1007/s13593-015-0319-9

De Cesaro Júnior, T., and Rieder, R. (2020). Automatic identification of insects from digital images: A survey. *Comput. Electron. Agric.* 178, 105784. doi: 10.1016/j.compag.2020.105784

De Cesaro Júnior, T., Rieder, R., Di Domênico, J. R., and Lau, D. (2022). InsectCV: A system for insect detection in the lab from trap images. *Ecol. Inf.* 67, 101516. doi: 10.1016/j.ecoinf.2021.101516

Deguine, J. P., Aubertot, J. N., Flor, R. J., Lescourret, F., Wyckhuys, K. A., and Ratnadass, A. (2021). Integrated pest management: good intentions, hard realities. A review. *Agron. Sustain. Dev.* 41, 38. doi: 10.1007/s13593-021-00689-w

Deng, L., Wang, Y., Han, Z., and Yu, R. (2018). Research on insect pest image detection and recognition based on bio-inspired methods. *Biosyst. Eng.* 169, 139–148. doi: 10.1016/j.biosystemseng.2018.02.008

Ding, W., and Taylor, G. (2016). Automatic moth detection from trap images for pest management. *Comput. Electron. Agric.* 123, 17–28. doi: 10.1016/j.compag.2016.02.003

Divyanth, L. G., Guru, D. S., Soni, P., Machavaram, R., Nadimi, M., and Paliwal, J. (2022). Image-to-image translation-based data augmentation for improving crop/weed classification models for precision agriculture applications. *Algorithms* 15, 401. doi: 10.3390/a15110401

Dong, W., Roy, P., and Isler, V. (2018). Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. *J. Field Robotics* 37, 97–121. doi: 10.1002/rob.21876

Dong, X., Zhang, Z., Yu, R., Tian, Q., and Zhu, X. (2020). Extraction of information about individual trees from high-spatial-resolution UAV-acquired images of an orchard. *Remote Sens.* 12, 133. doi: 10.3390/rs12010133

Dos Santos, A., Biesseck, B. J. G., Latte, N., de Lima Santos, I. C., dos Santos, W. P., Zanetti, R., et al. (2022). Remote detection and measurement of leaf-cutting ant nests using deep learning and an unmanned aerial vehicle. *Comput. Electron. Agric.* 198, 107071. doi: 10.1016/j.compag.2022.107071

Du, L., Sun, Y., Chen, S., Feng, J., Zhao, Y., Yan, Z., et al. (2022). A novel object detection model based on faster R-CNN for spodoptera frugiperda according to feeding trace of corn leaves. *Agriculture* 12, 248. doi: 10.3390/agriculture12020248

Espinoza, K., Valera, D. L., Torres, J. A., López, A., and Molina-Aiz, F. D. (2016). Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture. *Comput. Electron. Agric.* 127, 495–505. doi: 10.1016/j.compag.2016.07.008

Fang, W., Yue, L., and Dandan, C. (2020). "Classification system study of soybean leaf disease based on deep learning," in *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*. (IEEE) 1–5.

Genaev, M. A., Komyshev, E. G., Shishkina, O. D., Adonyeva, N. V., Karpova, E. K., Gruntenko, N. E., et al. (2022). Classification of fruit flies by gender in images using smartphones and the YOLOv4-tiny neural network. *Mathematics* 10, 295. doi: 10.3390/math10030295

Ghosh, A., Jangra, S., Dietzgen, R. G., and Yeh, W.-B. (2021). Frontiers approaches to the diagnosis of thrips (Thysanoptera): how effective are the molecular and electronic detection platforms? *Insects* 12, 920. doi: 10.3390/insects12100920

Guo, Q., Wang, C., Xiao, D., and Huang, Q. (2021). An enhanced insect pest counter based on saliency map and improved non-maximum suppression. *Insects* 12 (8), 705. doi: 10.3390/insects12080705

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., and Xu, C. (2020). "GhostNet: more features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 143–144. doi: 10.48550/arXiv.1911.11907

Hansen, L. P., Svenning, J. C., Olsen, K., Dupont, S., Garner, B., Iosifidis, A., et al. (2019). Species-level image classification with convolutional neural network enables insect identification from habitus images. *Ecol. Evol.* 10, 737–747. doi: 10.1002/ece3.5921

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778. doi: 10.1109/CVPR.2016.90

Hong, S. J., Nam, I., Kim, S. Y., Kim, E., Lee, C. H., Ahn, S., et al. (2021). Automatic pest counting from pheromone trap images using deep learning object detectors for matsucoccus thunbergianae, monitoring. *Insects* 12, 342. doi: 10.3390/insects12040342

Hossain, M. I., Paul, B., Sattar, A., and Islam, M. M. (2019). "A convolutional neural network approach to recognize the insect: A perspective in Bangladesh," in *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*. 384–389. doi: 10.1109/SMART46866.2019.9117442

Howard, J., and Gugger, S. (2020). fastai: A layered API for deep learning. *Information* 11 (2), 108. doi: 10.3390/info11020108

Huang, W., Feng, J., Wang, H., and Sun, L. (2020). A new architecture of densely connected convolutional networks for pan-sharpening. *ISPRS Int. J. Geo-Information* 9, 242. doi: 10.3390/ijgi9040242

Huang, Y., Li, R., Wei, X., Wang, Z., Ge, T., and Qiao, X. (2022). Evaluating data augmentation effects on the recognition of sugarcane leaf spot. *Agriculture* 12, 1997. doi: 10.3390/agriculture12121997

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708. doi: 10.1109/CVPR.2017.243

Imagga (2020) *Build the next generation of image recognition applications with imagga's API*. Available at: https://imagga.com/.

Iost Filho, F. H., de Bastos Pazini, J., de Medeiros, A. D., Rosalen, D. L., and Yamamoto, P. T. (2022). Assessment of injury by four major pests in soybean plants using hyperspectral proximal imaging. *Agronomy* 12, 1516. doi: 10.3390/agronomy12071516

Iost Filho, F. H., Heldens, W. B., Kong, Z., and de Lange, E. S. (2019). Drones: innovative technology for use in precision pest management. *J. Economic Entomology* 113, 1–25. doi: 10.1093/jee/toz268

Johansen, K., Raharjo, T., and McCabe, M. (2018).). Using multi-spectral UAV imagery to extract tree crop structural properties and assess pruning effects. *Remote Sens.* 10, 854. doi: 10.3390/agronomy12071516

Karam, C., Awad, M., Abou Jawdah, Y., Ezzeddine, N., and Fardoun, A. (2022). GAN-based semi-automated augmentation online tool for agricultural pest detection: A case study on whiteflies. *Front. Plant Sci.* 13, 813050.

Kasinathan, T., Singaraju, D., and Uyyala, S. R. (2021). Insect classification and detection in field crops using modern machine learning techniques. *Inf. Process. Agric.* 8 (3), 446–457. doi: 10.1016/j.inpa.2020.09.006

Khanramaki, M., Asli-Ardeh, E. A., and Kozegar, E. (2021). Citrus pests classification using an ensemble of deep learning models. *Comput. Electron. Agric.* 186, 106192. doi: 10.1016/j.compag.2021.106192

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O., Turner, M., Niazi, M., et al. (2010). Systematic literature reviews in software engineering-A tertiary study. *Inf. Software Technol.* 52, 792–805. doi: 10.1016/j.infsof.2010.03.006

Knyshov, A., Hoang, S., and Weirauch, C. (2021). Pretrained convolutional neural networks perform well in a challenging test case: identification of plant bugs (Hemiptera: miridae) using a small number of training images. *Insect Systematics Diversity* 5 (2), 3. doi: 10.1093/isd/ixab004

Kumar, D., and Kukreja, V. (2022). Deep learning in wheat diseases classification: A systematic review. *Multimedia Tools Appl.* 81, 10143–10187. doi: 10.1007/s11042-022-12160-3

Kusrini, K., Suputa, S., Setyanto, A., Agastya, I. M. A., Priantoro, H., Chandramouli, K., et al. (2021). Data augmentation for automated pest classification in Mango farms. *Comput. Electron. Agric.* IF5, 565. doi: 10.1016/j.compag.2020.105842

Kuzuhara, H., Takimoto, H., Sato, Y., and Kanagawa, A. (2020). "Insect pest detection and identification method based on deep learning for realizing a pest control system," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 709–714.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Li, W., Chen, P., Wang, B., and Xie, C. (2019). Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline. *Sci. Rep.* 9, 10.1038/s41598-019-43171-0. doi: 10.1038/s41598-019-43171-0

Li, Z., Song, J., Qiao, K., Li, C., Zhang, Y., and Li, Z. (2022). Research on efficient feature extraction: Improving YOLOv5 backbone for facial expression detection in live streaming scenes. *Front. Comput. Neurosci.* 16, 980063. doi: 10.3389/fncom.2022.980063

Li, C., Zhen, T., and Li, Z. (2022). Image classification of pests with residual neural network based on transfer learning. *Appl. Sci.* 12 (9), 4356. doi: 10.3390/app12094356

Li, J., Zhou, H., Wang, Z., and Jia, Q. (2020). Multi-scale detection of stored-grain insects for intelligent monitoring. *Comput. Electron. Agric.* 168. doi: 10.1016/j.compag.2019.105114

Li, W., Zhu, X., Yu, X., Li, M., Tang, X., Zhang, J., et al. (2022). Inversion of nitrogen concentration in apple canopy based on UAV hyperspectral images. *Sensors* 22, 3503. doi: 10.3390/s22093503

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved YOLO V3 convolutional neural network. *Front. Plant Sci.* 11, 898. doi: 10.3389/fpls.2020.00898

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17. doi: 10.1186/s13007-021-00722-9

Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., et al. (2019). PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access* 7, 45301–45312. doi: 10.1109/ACCESS.2019.2909522

Liu, L., Wang, R., Xie, C., Li, R., Wang, F., and Qi, L. (2022). A global activated feature pyramid network for tiny pest detection in the wild. *Mach. Vision Appl.* 33 (5). doi: 10.1007/s00138-022-01310-0

López-Granados, F., Torres-Sánchez, J., Jiménez-Brenes, F. M., Arquero, O., Lovera, M., and de Castro, A. I. (2019). An efficient RGB-UAV-based platform for field almond tree phenotyping: 3-D architecture and flowering traits. *Plant Methods* 15, 160. doi: 10.1186/s13007-019-0547-0

Lu, C. Y., Rustia, D. J. A., and Lin, T. T. (2019). Generative adversarial network based image augmentation for insect pest classification enhancement. *IFAC-PapersOnLine* 52 (30), 1–5. doi: 10.1016/j.ifacol.2019.12.406

Lv, J., Li, W., Fan, M., Zheng, T., Yang, Z., Chen, Y., et al. (2022). Detecting pests from light-trapping images based on improved YOLOv3 model and instance augmentation. *Front. Plant Sci.* 13, 939498. doi: 10.3389/fpls.2022.939498

Malathi, V., and Gopinath, M. P. (2021). Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica Section B — Soil Plant Sci.* 71, 552–559. doi: 10.1080/09064710.2021.1874045

Maryland Biodiversity Database (2022). Available at: https://www.marylandbiodiversity.com.

Mavridou, E., Vrochidou, E., Papakostas, G. A., Pachidis, T., and Kaburlasos, V. G. (2019). Machine vision systems in precision agriculture for crop farming. *J. Imaging* 5 (12), 89. doi: 10.3390/jimaging5120089

Misango, V. G., Nzuma, J. M., Irungu, P., and Kassie, M. (2022). Intensity of adoption of integrated pest management practices in Rwanda: A fractional logit approach. *Heliyon* 8 (1), e08735. doi: 10.1016/j.heliyon.2022.e08735

Mu, Y., Fujii, Y., Takata, D., Zheng, B., Noshita, K., Kiyoshi, H., et al. (2018). Characterization of peach tree crown by using high-resolution images from an unmanned aerial vehicle. *Horticulture Res.* 5, 74. doi: 10.1038/s41438-018-0097-z

Nagar, H., and Sharma, R. S. (2021) in *2021 International Conference on Computer Communication and Informatics (ICCCI)*. 1–5.

Nanni, L., Manfè, A., Maguolo, G., Lumini, A., and Brahnam, S. (2022). High performing ensemble of convolutional neural networks for insect pest image detection. *Ecol. Inf.* 67, 101515. doi: 10.1016/j.ecoinf.2021.101515

Nazri, A., Mazlan, N., and Muharam, F. (2018). PENYEK: Automated brown planthopper detection from imperfect sticky pad images using deep convolutional neural network. *PloS One* 13 (12), e0208501. doi: 10.1371/journal.pone.0208501

Padmanabhuni, S. S., and Gera, P. (2022). Synthetic data augmentation of tomato plant leaf using meta intelligent generative adversarial network: milgan. *Int. J. Advanced Comput. Sci. Appl. (IJACSA)* 13 (6). doi: 10.14569/IJACSA.2022.0130628

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* 372, n71. doi: 10.1136/bmj.n71

Partel, V., Nunes, L. M., Stansly, P., and Ampatzidis, Y. (2019). Automated vision-based system for monitoring Asian citrus psyllid in orchards utilizing artificial intelligence. *Comput. Electron. Agric.* 162, 328–336. doi: 10.1016/j.compag.2019.04.022

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *In Adv. Neural Inf. Process. Syst.* pp, 8024–8035. doi: 10.48550/arXiv.1912.01703

Popescu, D., El-Khatib, M., El-Khatib, H., and Ichim, L. (2022). New trends in melanoma detection using neural networks: A systematic review. *Sensors* 22, 496. doi: 10.3390/s22020496

Popescu, D., Stoican, F., Stamatescu, G., Ichim, L., and Dragana, C. (2020). Advanced UAV–WSN system for intelligent monitoring in precision agriculture. *Sensors* 20, 817. doi: 10.3390/s20030817

Popkov, A., Konstantinov, F., Neimorovets, V., and Solodovnikov, A. (2022). Machine learning for expert-level image-based identification of very similar species in the hyperdiverse plant bug family Miridae (Hemiptera: Heteroptera). *Systematic Entomology* 47, 487–503. doi: 10.1111/syen.12543

Preti, M., Verheggen, F., and Angeli, S. (2021). Insect pest monitoring with camera-equipped traps: strengths and limitations. *J. Pest Sci.* 94, 203–217. doi: 10.1007/s10340-020-01309-4

Rajeena, P. P., Orban, F., Vadivel, K. S., Subramanian, M., Muthusamy, S., Elminaam, D. S. A., et al. (2022). A novel method for the classification of butterfly species using pre-trained CNN models. *Electronics* 11. doi: 10.3390/electronics11132016

Ramadhan, A. A., and Baykara, M. (2022). A novel approach to detect COVID-19: enhanced deep learning models with convolutional neural networks. *Appl. Sci.* 12 (18), 9325. doi: 10.3390/app12189325

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.

Redmon, J., and Farhadi, A. (2017). "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7263–7271.

Redmon, J., and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv*, 1804.02767. doi: 10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of the IEEE International Conference on Computer Vision*. (Las Vegas, NV, USA) 91–99.

Rimal, K., Shah, K. B., and Jha, A. K. (2022). Advanced multi-class deep learning convolution neural network approach for insect pest classification using TensorFlow. *Int. J. Environ. Sci. Technol.* 20, 4003–4016. doi: 10.1007/s13762-022-04277-7

Ronchetti, G., Mayer, A., Facchi, A., Ortuani, B., and Sona, G. (2020). Crop row detection through UAV surveys to optimize on-farm irrigation management. *Remote Sens.* 12, 1967. doi: 10.3390/rs12121967

Roosjen, P., Kellenberger, B., Kooistra, L., Green, D., and Fahrentrapp, J. (2020). Deep learning for automated detection of Drosophila suzukii–Potential for UAV-based monitoring. *Pest Manage. Sci.* 76, 2994–3002. doi: 10.1002/ps.5845

Rustia, D. J., Chao, J. J., Chiu, L. Y., Wu, Y. F., Chung, J. Y., Hsu, J. C., et al. (2020). Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. *J. Appl. Entomology* 145 (3), 206–222. doi: 10.1111/jen.12834

Sanghavi, V. B., Bhadka, H., and Dubey, V. (2022). Hunger games search based deep convolutional neural network for crop pest identification and classification with transfer learning. *Evolving Syst.* 14, 649–671. doi: 10.1007/s12530-022-09449-x

Segalla, A., Fiacco, G., Tramarin, L., Nardello, M., and Brunelli, D. (2020). "Neural networks for pest detection in precision agriculture," in *2020 5th IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor*. (Trento, Italy) 7–12.

Sekabira, H., Tepa-Yotto, G. T., Djouaka, R., Clottey, V., Gaitu, C., Tamò, M., et al. (2022). Determinants for deployment of climate-smart integrated pest management practices: A meta-analysis approach. *Agriculture* 12, 1052. doi: 10.3390/agriculture12071052

Sharma, P., Berwal, Y. P. S., and Ghai, W. (2020). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Inf. Process. Agric.* 7 (4), 566–574. doi: 10.1016/j.inpa.2019.11.001

Shi, Z., Hao, D., Liu, Z., and Zhou, X. (2020). Detection and identification of stored-grain insects using deep learning: A more effective neural network. *IEEE Access* 8, 163703–163714. doi: 10.1109/ACCESS.2020.3021830

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv* 1409, 1556. doi: 10.48550/arXiv.1409.1556

Singh, P., Verma, A., and Alex, J. S. R. (2021). Disease and pest infection detection in coconut tree through deep learning techniques. *Comput. Electron. Agric.* 182, 105986. doi: 10.1016/j.compag.2021.105986

Stefas, N., Bayram, H., and Isler, V. (2016). Vision-based UAV navigation in orchards. *PapersOnLine* 49 (16), 10–15. doi: 10.1016/j.ifacol.2016.10.003

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Boston, MA, USA) 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the Inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (Las Vegas, NV, USA) 2818–2826.

Takimoto, H., Sato, Y., Nagano, A. J., Shimizu, K. K., and Kanagawa, A. (2021). Using a two-stage convolutional neural network to rapidly identify tiny herbivorous beetles in the field. *Ecol. Inf.* 66, 101466. doi: 10.1016/j.ecoinf.2021.101466

Teng, Y., Zhang, J., Dong, S., Zheng, S., and Liu, L. (2022). MSR-RCNN: A multi-class crop pest detection network based on a multi-scale super-resolution feature enhancement module. *Front. Plant Sci.* 13, 810546. doi: 10.3389/fpls.2022.810546

Terentev, A., Dolzhenko, V., Fedotov, A., and Eremenko, D. (2022). Current state of hyperspectral remote sensing for early plant disease detection: A review. *Sensors* 22, 757. doi: 10.3390/s22030757

Thakare, P., and Sankar, V. (2022). Advanced pest detection strategy using hybrid optimization tuned deep convolutional neural network. *J. Eng. Design Technol.* doi: 10.1108/JEDT-09-2021-0488

Thenmozhi, K., and Srinivasulu Reddy, U. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* 164, 104906. doi: 10.1016/j.compag.2019.104906

Tian, G., Liu, C., Liu, Y., Li, M., Zhang, J., and Duan, H. (2020). Research on plant diseases and insect pests identification based on CNN. *IOP Conf. Series: Earth Environ. Sci.* 594, 012009. doi: 10.1088/1755-1315/594/1/012009

Tian, H., Wang, T., Liu, Y., Qiao, X., and Li, Y. (2020). Computer vision technology in agricultural automation —A review. *Inf. Process. Agric.* 7 (1), 1–19. doi: 10.1016/j.inpa.2019.09.006

Tong, R., Wang, Y., Zhu, Y., and Wang, Y. (2022). Does the certification of agriculture products promote the adoption of integrated pest management among apple growers in China? *Environ. Sci. pollut. Res.* 29, 29808–29817. doi: 10.1007/s11356-022-18523-5

Toscano-Miranda, R., Toro, M., Aguilar, J., Caro, M., Marulanda, A., and Trebilcok, A. (2022). Artificial-intelligence and sensing techniques for the management of insect pests and diseases in cotton: A systematic literature review. *J. Agric. Sci.* 160 (1-2), 16–31. doi: 10.1017/S002185962200017X

Turkoglu, M., Hanbay, D., and Sengur, A. (2020). Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *J. Ambient Intell. Humanized Computing*. 13, 3335–3345. doi: 10.1007/s12652-019-01591-w

Turkoglu, M., Yanikoğlu, B., and Hanbay, D. (2022). PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *Signal Image Video Process.* 16, 301–309. doi: 10.1007/s11760-021-01909-2

Ultralytics (2020) *YOLOv5: A state-of-the-art real-time object detection system*. Available at: https://github.com/ultralytics/yolov5.

Valan, M., Makonyi, K., Maki, A., Vondráček, D., and Ronquist, F. (2019). Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Systematic Biol.* 68 (6), 876–895. doi: 10.1093/sysbio/syz014

Vanegas, F., Bratanov, D., Powell, K., Weiss, J., and Gonzalez, F. (2018). A novel methodology for improving plant pest surveillance in vineyards and crops using UAV-based hyperspectral and spatial data. *Sensors* 18, 260. doi: 10.3390/s18010260

Velusamy, P., Rajendran, S., Mahendran, R. K., Naseer, S., Shafiq, M., and Choi, J.-G. (2022). Unmanned aerial vehicles (UAV) in precision agriculture: applications and challenges. *Energies* 15, 217. doi: 10.3390/en15010217

Wang, J., Li, Y., Feng, H., Ren, L., Du, X., and Wu, J. (2020). Common pests image recognition based on deep convolutional neural network. *Comput. Electron. Agric.* 179, 105834. doi: 10.1016/j.compag.2020.105834

Wang, G., Sun, Y., and Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Comput. Intell. Neurosci.* 20178. doi: 10.1155/2017/2917536

Wang, Q.-J., Zhang, S.-Y., Dong, S.-F., Zhang, G.-C., Yang, J., Li, R., et al. (2020). Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. *Comput. Electron. Agric.* 175, 105585. doi: 10.1016/j.compag.2020.105585

Wang, T., Zhao, L., Li, B., Liu, X., Xu, W., and Li, J. (2022). Recognition and counting of typical apple pests based on deep learning. *Ecol. Inf.* 68, 101556. doi: 10.1016/j.ecoinf.2022.101556

Wen, C., and Guyer, D. E. (2012). Image-based orchard insect automated identification and classification method. *Comput. Electron. Agric.* 89, 110–115. doi: 10.1016/j.compag.2012.08.008

Wu, L., Liu, Z., Bera, T., Ding, H., Langley, D., Jenkins-Barnes, A., et al. (2019). A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Comput. Electron. Agric.* 166, 105002. doi: 10.1016/j.compag.2019.105002

Wu, X., Zhan, C., Lai, Y.-K., Cheng, M.-M., and Yang, J. (2019). "A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Long Beach, CA, USA) 8787–8796.

Xia, D., Chen, P., Wang, B., Zhang, J., and Xie, C. (2018). Insect detection and classification based on an improved convolutional neural network. *Sensors* 18, 4169. doi: 10.3390/s18124169

Xie, C., Wang, R., Zhang, J., Chen, P., Dong, W., Li, R., et al. (2018). Multi-level learning features for automatic classification of field crop pests. *Comput. Electron. Agric.* 152, 233–241. doi: 10.1016/j.compag.2018.07.014

Xie, C., Zhang, J., Li, R., Li, J., Hong, P., Xia, J., et al. (2015). Automatic classification for field crop insects *via* multiple-task sparse representation and multiple-kernel learning. *Comput. Electron. Agric.* 119, 123–132. doi: 10.1016/j.compag.2015.10.015

Xing, S., Lee, M., and Lee, K. (2019). Citrus pests and diseases recognition model using weakly dense connected convolution network. *Sensors* 19, 3195. doi: 10.3390/s19143195

Xu, R., Lin, H., Lu, K., Cao, L., and Liu, Y. (2021). A forest fire detection system based on ensemble learning. *Forests* 12, 217. doi: 10.3390/f12020217

Xu, C., Yu, C., Zhang, S., and Wang, X. (2022). Multi-scale convolution-capsule network for crop insect pest recognition. *Electronics* 11, 1630. doi: 10.3390/electronics11101630

Yuan, W., and Choi, D. (2021). UAV-based heating requirement determination for frost management in apple orchard. *Remote Sens.* 13, 273. doi: 10.3390/rs13020273

Zhang, Z. (2022). "Image recognition methods based on deep learning," in *3D imaging—Multidimensional signal processing and deep learning (Smart innovation, systems and technologies*, vol. 297 . Eds. L. C. Jain, R. Kountchev, Y. Tai and &R. Kountcheva (Springer), 1–15.

Zhang, Y., Cai, W., Fan, S., Song, R., and Jin, J. (2022). Object detection based on YOLOv5 and ghostNet for orchard pests. *Information* 13, 548. doi: 10.3390/info13110548

Zhang, X., and Chen, G. (2020). An automatic insect recognition algorithm in complex background based on convolution neural network. *Traitement du Signal* 37, 793–798. doi: 10.18280/ts.370511

Zhang, S., Jing, R., and Shi, X. (2022). Crop pest recognition based on a modified capsule network. *Syst. Sci. Control Eng.* 10 (1), 552–561. doi: 10.1080/21642583.2022.2074168

Zhang, N., Wang, Y., and Zhang, X. (2020). Extraction of tree crowns damaged by Dendrolimus tabulaeformisTsai et Liu *via* spectral-spatial classification using UAV-based hyperspectral images. *Plant Methods* 16, 135. doi: 10.1186/s13007-020-00678-2

Zhang, S., Yang, H., Yang, C., Yuan, W., Li, X., Wang, X., et al. (2023). Edge device detection of tea leaves with one bud and two leaves based on shuffleNetv2-YOLOv5-lite-E. *Agronomy* 13, 577. doi: 10.3390/agronomy13020577

Zhong, Y., Gao, J., Lei, Q., and Zhou, Y. (2018). A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors* 18, 1489. doi: 10.3390/s18051489

Zhu, J., Wu, A., Wang, X., and Zhang, H. (2020). Identification of grape diseases using image analysis and BP neural networks. *Multimedia Tools Appl.* 79, 14539–14551. doi: 10.1007/s11042-018-7092-0

frontiers | Frontiers in Plant Science

TYPE Review
PUBLISHED 27 November 2023
DOI 10.3389/fpls.2023.1237695

OPEN ACCESS

EDITED BY
Liangliang Yang,
Kitami Institute of Technology, Japan

REVIEWED BY
Asli Ozdarici-Ok,
Ankara Haci Bayram Veli University, Türkiye
Corneliu Lazar,
Gheorghe Asachi Technical University of Iaşi, Romania

*CORRESPONDENCE
Dan Popescu
dan.popescu@upb.ro

RECEIVED 09 June 2023
ACCEPTED 30 October 2023
PUBLISHED 27 November 2023

CITATION
Popescu D, Ichim L and Stoican F (2023) Orchard monitoring based on unmanned aerial vehicles and image processing by artificial neural networks: a systematic review.
Front. Plant Sci. 14:1237695.
doi: 10.3389/fpls.2023.1237695

Orchard monitoring based on unmanned aerial vehicles and image processing by artificial neural networks: a systematic review

Dan Popescu*, Loretta Ichim and Florin Stoican

Faculty of Automatic Control and Computers, National University of Science and Technology Politehnica Bucharest, Bucharest, Romania

Orchard monitoring is a vital direction of scientific research and practical application for increasing fruit production in ecological conditions. Recently, due to the development of technology and the decrease in equipment cost, the use of unmanned aerial vehicles and artificial intelligence algorithms for image acquisition and processing has achieved tremendous progress in orchards monitoring. This paper highlights the new research trends in orchard monitoring, emphasizing neural networks, unmanned aerial vehicles (UAVs), and various concrete applications. For this purpose, papers on complex topics obtained by combining keywords from the field addressed were selected and analyzed. In particular, the review considered papers on the interval 2017-2022 on the use of neural networks (as an important exponent of artificial intelligence in image processing and understanding) and UAVs in orchard monitoring and production evaluation applications. Due to their complexity, the characteristics of UAV trajectories and flights in the orchard area were highlighted. The structure and implementations of the latest neural network systems used in such applications, the databases, the software, and the obtained performances are systematically analyzed. To recommend some suggestions for researchers and end users, the use of the new concepts and their implementations were surveyed in concrete applications, such as a) identification and segmentation of orchards, trees, and crowns; b) detection of tree diseases, harmful insects, and pests; c) evaluation of fruit production, and d) evaluation of development conditions. To show the necessity of this review, in the end, a comparison is made with review articles with a related theme.

KEYWORDS

orchard monitoring, unmanned aerial vehicle, dataset, image processing, neural network, object detection, object segmentation, object classification

1 Introduction

The monitoring of modern orchards based on the acquisition and continuous processing of data has become a necessity for obtaining the highest possible production of healthy fruits. Within the data processing field, image processing is of particular interest for orchard monitoring because it efficiently solves several essential aspects like orchard mapping, tree segmentation, production (fruit) evaluation, disease detection, the need for water or special solutions, pest detection, etc. Both RGB (red-green-blue) and multispectral images are used to evaluate the parameters characterizing the orchard problems. They provide a significant volume of information used for efficient monitoring. The correct acquisition of images is necessary so that the regions of interest are of good quality. Various vectors have been used for image acquisition, such as human operators with cameras or smartphones, fixed cameras, cameras on land vehicles, aerial vehicles (autonomous or not), and satellites (Lin et al., 2021). Collecting image data in a complex 3D space, such as an orchard, is a relatively recent challenge made possible by the recent development of new technologies. Consequently, due to both the technological improvements and the economic aspects promoted by large-scale production, many agriculture-related problems have been augmented with the integration of artificial intelligence techniques and remote sensing systems. Although satellites and UAVs (Unmanned Aerial Vehicles) complement each other in the task of inspecting different terrestrial areas, in the case of orchard monitoring, UAVs offer clear advantages such as ultra-resolution, cost-effective operation, increased flexibility for individual tree inspection, and resilience against weather patterns such as cloudy (Alvarez-Vanhard et al., 2021). Not least, for the monitoring of crops in precision agriculture, collaboration with wireless ground sensor networks is of particular importance (Popescu et al., 2020). On the other hand, in complex applications related to orchard monitoring, UAVs have the advantage to take images from either a medium distance (10 m -100 m) through an appropriate design of the trajectories - such as in the case of orchard or tree segmentation (Adamo et al., 2021; Akca and Polat, 2022) or to determine the water stress index (Zhang C. et al., 2021) or from a smaller distance (tens of cm) - such as the case of detecting harmful insects (Aota et al., 2021; Ichim et al., 2022) or fruits (Wang S. et al., 2022). The UAVs compared to terrestrial robots is also a more flexible and less expensive solution. The automatic picking of fruits is an exception. In the future, the use of complex multirobot systems that combine the actions of UAVs, ground robots, and manipulators (Sulistijono et al., 2020; Ju et al., 2022) can lead to an increased degree of automation in modern orchards. However, research papers related to the application of artificial intelligence and the use of drones (UAVs) in the monitoring of orchards are relatively few compared to the monitoring of flat, field crops. This is a consequence of considering the 3D space in orchard applications.

It should not be forgotten that an essential condition for the effective use of UAVs is flights performed beyond the visual range of the operators. Due to the strong increase in the number of operational UAVs, it has become necessary to analyze the conditions for making safe flights in shared airspace. In this sense, working meetings are increasingly taking place at the level of the European Union to update the relevant flight regulations. For the safe operation of many drones, the "U-space" concept was introduced into European legislation (Barrado et al., 2020) to manage UAS (unmanned aerial systems) traffic. It refers to the framework of regulations, technologies, and procedures required to enable safe and efficient drone operations in low-altitude airspace. With the integration of drones into the airspace system, U-space provides a framework for ensuring safety, security, and efficiency in their operation. The continued development and implementation of U-space regulations and technologies are essential to realizing the full potential of drones and their applications in the future. The term refers to a collection of digitized and automated functions and processes aimed at ensuring safe, efficient, and equitable access to airspace for the growing number of civilian drone operators. This is essential for enabling the many benefits of drone technology, such as improved delivery services, monitoring and inspection of agricultural crops, and support for emergency services. Not least, by requiring pilots to obtain a license and submit a flight plan, U-space regulations help to mitigate the risks associated with drone operations and promote the responsible and safe use of this technology.

Efficient monitoring in precision agriculture requires precise mapping of agricultural crops and, implicitly, orchards. That is why the detection and location of orchards and trees in the orchard with the help of aerial robots and neural networks have undergone a spectacular evolution in recent years (Osco et al., 2020; Zhang et al., 2018; Osco et al., 2021). In precision agriculture, terrestrial robots and UAVs were used for instance segmentation and fine detection of crops, trees, and weed plants (Champ et al., 2020; Chen et al., 2019; Khan et al., 2020a). It can be stated that drones and neural networks are essential ingredients in precise and intelligent agriculture. As per (Jensen et al., 2021), pesticide usage is 30% of the total cost in citrus and 42% in olive orchards. The pesticide reduction is discussed in (Özyurt et al., 2022) where UAVs are used to assess areas in need of spraying in a hazelnut. The actual application of pesticides is not straightforward: multi-rotor UAVs are severely restricted in the maximum payload weight. Time is also a factor. (Zortea et al., 2018) show that a month of manual labeling in the field is replaced by a week of manually labeling images obtained from a UAV flight (which may be further reduced to less than a day when automatizing the labeling procedure). Noteworthy, no single algorithm works for any type of orchard/forest (Larsen et al., 2011).

Monitoring of orchards through automated methods based on image processing and artificial intelligence leads to increased productivity while reducing expenses. Application of deep learning for the delineation of visible cadastral boundaries of parcels in rural scenes from UAV imagery can be used with smaller effort for delineation compared to manual delineation (Crommelinck et al., 2019). This means adjusting data processing systems to various conditions, types, or sizes of orchards. Thus, recently, machine learning methods, intelligent classifiers, and, especially, convolutional neural networks (CNN) have been used for the detection, classification, and segmentation of regions of interest (RoI) from images acquired in the orchard for various applications. As a trend, Deep Convolutional Neural Networks (DCNNs) are increasingly used in object detection (Xiao et al., 2020), a particularly important aspect in orchard management (e.g., detection of fruits, diseases, insects, etc.). Deep neural networks and transfer learning were used for food crop

identification from UAV images (Chew et al., 2020). In the review paper (Alzubaidi et al., 2021), the main components of DCNN used for object detection are detailed, emphasizing the advantage offered by these networks to automatically detect the main features used without human intervention. Specifically, in fruit detection problems, several recent works have been making use of Deep Learning (DL) methods applied to images acquired at different height levels (Biffi et al., 2021).

The measurement of size, growth, and mortality of individual trees is of utmost importance for orchard or forest monitoring. To this end, the authors (Hu and Li, 2020) proposed a point cloud segmentation method for single trees. They used UAV tilt photography and a simple neural network (NN) for data processing feature extraction and classification tasks with an accuracy of about 90%. A method to detect, geolocate, and identify tree species by UAV flight and NN processing of acquired hyperspectral images is presented by (Miyoshi et al., 2020). UAVs are also used as a cheap and reliable solution for measuring the height of crops (Xie et al., 2021), including orchard trees. In this case, additional spatial information such as the digital terrain model and the ground truth of the height is required. In such cases, it becomes especially important to correct the positioning errors of global navigation satellite systems (GNSS) by different methods. To this end, UAVs are often equipped with a real-time kinematic positioning (RTK) module.

The early detection of tree disease in orchards can significantly improve the control of these diseases and avoid the spread of insects, viruses, or fungi. For example, vine disease detection by automatic methods leads to increase efficiency and productivity of vineyard crops in smart farming, simultaneously with the reduction of pesticides. Therefore, the detection of vine diseases in UAV images using neural networks has been widely addressed recently (Kerkech et al., 2018; Kerkech et al., 2020).

A difficulty that can be encountered in orchard monitoring is the dense tree crowns. This can often cause GPS (Global Positioning System) signal attenuation when the UAV or a terrestrial robot is traveling in an orchard. A method to overcome this drawback is proposed by (Kim et al., 2020) using a CNN to classify patches in the front image in path, tree, or background. For this purpose, the image is traversed successively with sliding investigation windows, and a path score map is generated through the CNN classification results.

Broadly speaking, an orchard monitoring system based on the use of UAVs and NNs has the structure of Figure 1. It has two main paths, system learning and system operating. In the first phase, the UAV acquires the images for the dataset (DS) needed for the learning and validation phases to establish the parameters and weights NN(L). Sometimes the dataset can be a public one. The images need a preprocessing set of operations by IPp (Image Preprocessing module). After learning, validation, and final configuration of structure NN(C), it is implemented in the operating configuration NN(O) on a terrestrial operating station or even on the UAV. The system output is a decision or/and a new image (D/I). In orchard monitoring, the respective applications and images are very different and therefore the choice of UAV trajectories to obtain the most relevant data (images) and especially the choice of NNs used for the analysis of the regions of interest constitute real challenges. Still, newer is the integration of the monitoring of agricultural crops, including orchards, into IoT (Internet of Things). Thus, if real-time processing of monitoring data is required, as in the case of pest detection, a solution presented by (Bhoi et al., 2021) is a UAV assisted by IoT, where images of pests are sent for processing to the Imagga cloud (https://imaga.com), to retrieve the pest information.

The current paper focuses on the importance of UAVs and image processing through artificial intelligence techniques (in particular, CNN) for orchard monitoring from various points of view such as flowering, evolution, diseases, harmful insects, fruit ripening, and picking. Thus, the paper focuses on the new trends in the use of UAVs and image processing based on NNs for efficient monitoring of orchards in precision agriculture with ecological considerations. Apart from the Introduction, the paper contains five sections. Section 2, entitled Survey Methodology, presents the methodology for investigating papers in the field from 2017-2022. Section 3, named UAVs and Cameras Used for Image Acquisition in Orchard Monitoring, presents the UAVs and video/photo cameras used in the analyzed papers, the characteristics of UAV trajectories in orchard monitoring, and develops the aspects related to the design and tracking of UAV trajectories in the orchard. Section 4, Neural Networks Used for Orchard Monitoring, refers to the presentation of the neural networks used, datasets, software, performances, and the new implementation trends based on the fusion of decisions or the combination of several neural networks. Section 5, Applications, is dedicated to the most frequent orchard monitoring applications through the prism of new technologies. In Section 6, Discussions, some observations are made regarding the global aspects of research in the field from the last three years and comparisons with review papers based on the same keywords. The last section is the Conclusions which highlights the important aspects of the paper. All development chapters are accompanied by graphs or synthetic tables. Since there are many notions and definitions that are repeated or are put in tables, in order not to fill unnecessary space and to make it easier to understand, a list with abbreviations is provided as Annex 1.

2 Survey methodology

For the systematic review paper, 872 papers were analyzed from different databases such as the Web of Science (311), Scopus (292), and IEEE Xplore (269). Finally, we selected 197 papers (173 research papers and 24 review-type papers) for this review. The eligibility criteria for paper selection were recent publications, new trends in orchard monitoring on different aspects, the impact of contributions, the involvement of UAVs, and the use of NNs in the processing of images acquired in orchards. As the impact, the citations can be a relative criterion because, in general for older papers, the citations are higher than for newer ones. The high rank of publications refers to Category Quartile Q1, Q2, and the Journal Impact Factor in Web of Science 2021. More than 68% of the total references meet this criterion. Most of the papers included in this study are from journals with an impact factor greater than 2. Among the analyzed articles, 167 are from journals and 30 are from conferences. Focusing on a relatively recent period (2017 – 2022), the most representative papers covering the ROI detection, segmentation, and classification in orchard images, using state-of-the-art NNs and UAVs, were investigated. Thus, 184 references

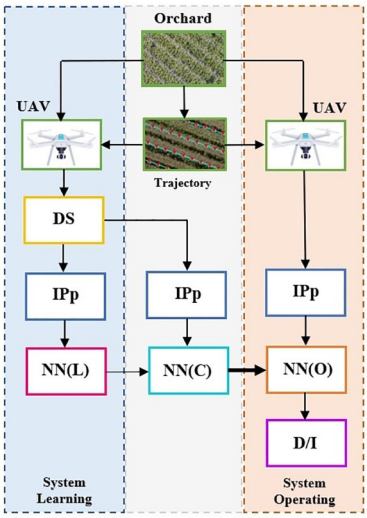


FIGURE 1
Structure of the orchard monitoring system composed of UAVs and neural networks. UAV – unmanned aerial vehicle (drone), DS – data set, IPp – image preprocessing module, NN(L) neural network learning (parameters and weights), NN(C) – final NN configuration (after validation), NN(O) – neural network implemented for operating phase.

between 2017 - 2022, representing 93.40% of the total, were selected, and focusing on 2019 – 2022, as a recent period, 84.69% of references were analyzed. In terms of new trends in using NNs for UAV image analysis, the following directions can be mentioned: a) improvement of a CNN with other networks included in its structure, most often adapted for orchard images, b) systems made up of several CNNs (that can be considered as elements of collective intelligence), and c) systems using CNN combined with other classifiers. This important aspect is detailed in Section 4.

For the systematic review and meta-analysis, we used a PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) (http://www.prisma-statement.org/) flow diagram (Figure 2). As can be seen from the diagram, from the total of 892 identified papers, we selected 197 papers according to the criteria mentioned in Figure 2. For the paper search strategy, we investigated similar papers in the field. The comparisons and the highlighting of the degree of novelty towards them are underlined in Section 6, Discussions. Most of the analyzed articles were selected from journals (Figure 2) such as Remote Sensing (RS), Computers and Electronics in Agriculture (CEA), Frontiers in Plant Science (FPS), Sensors (S), and IEEE Access (Access).

Although concerns about the orchard, UAVs or NNs used separately are older and the respective fields of study are well-established, the combination of these topics in orchard monitoring is relatively recent. As we considered the new trends in this direction, Figure 3 is presented our search in Web of Science (blue), Scopus (red), and IEEE Xplore (green) databases (DBs) between 2017 - 2022 considering the following topics: UAV control, UAV trajectory, U-space, agriculture, orchard, NNs, image processing, diseases, insects, and fruit production. It should be noted that to save space in Figure 3, the notation "uav" means UAV, UAS, or drone. The search was split between combinations of keywords using the "AND" connector: (A) neural networks AND image processing, (B) agriculture AND image processing, (C) orchard AND image processing, (D) orchard AND neural networks, (E) orchard AND uav, (F) orchard AND neural networks AND uav, (G) uav AND control AND neural networks, (H) uav AND trajectory AND neural networks, (I) uav AND U-space, (J) agriculture AND uav AND image processing, (K) orchard AND uav AND image processing, (L) agriculture AND uav AND neural networks, (M) orchard AND diseases, (N) orchard AND insects, and (O) orchard AND fruit production. The year is labeled on the x-axis and the number of publications identified according to the search in the database is labeled on the y-axis. It can be observed that the increase in research is higher in most of the cases involving NNs and/or UAVs, with an exception in 2022 because of the indexing latency. Furthermore, it should be noted that while we have strived for a fair comparison between Web of Science, Scopus, and IEEE Xplore, they do have different ways to handle queries, such as those we constructed, for obtaining the results from Figure 3. Because IEEE Xplore is not a paper database focused on agriculture the number of papers is much smaller compared to Scopus and Web of Science when the topic of agriculture or orchard appears in searches so that they can be neglected. Also, there is a big difference between the number of papers related to the use of NN and/or UAV in orchards compared to agriculture in general. This can be attributed to the difficulties of flying inside the orchards, the consideration of images in depth (tree crowns), and partially covered objects. In general, we see a rapid increase in papers

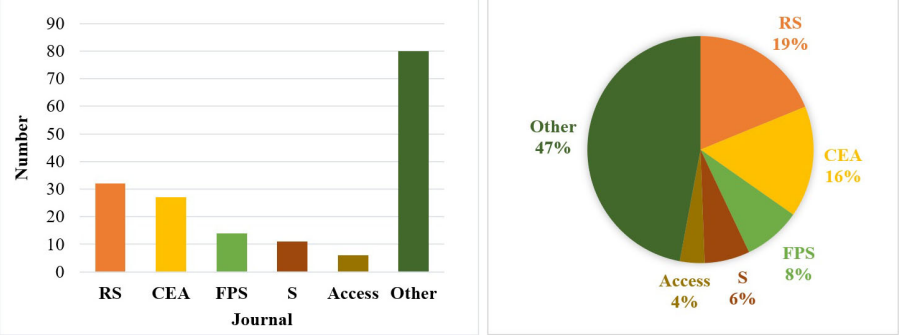


FIGURE 2
The number (left) and the percentage (right) of papers that are analyzed from journals: Remote Sensing (RS), Computers and Electronics in Agriculture (CEA), Frontiers in Plant Science (FPS), Sensors (S) and IEEE Access (Access).

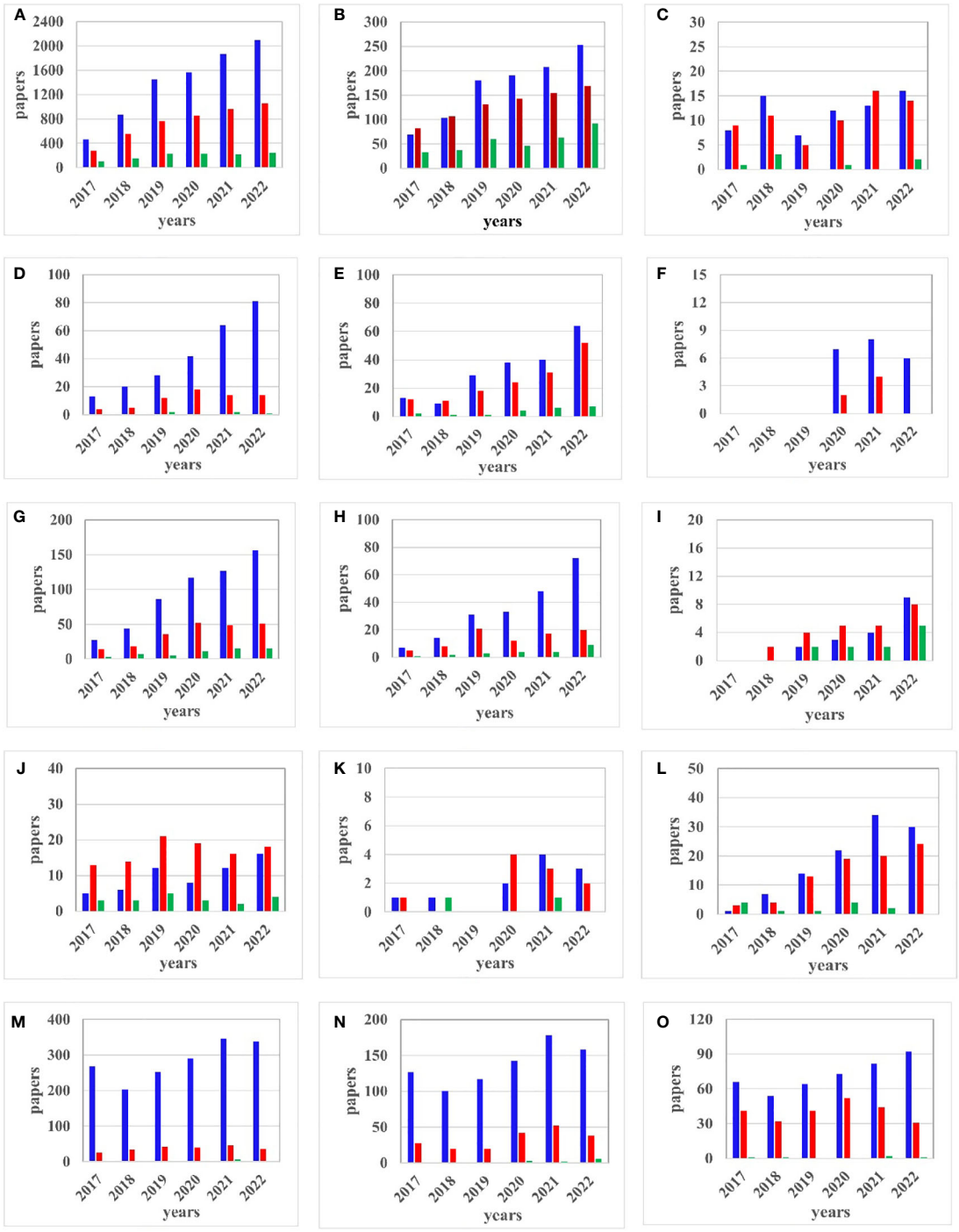


FIGURE 3
Web of Science-blue, Scopus – red, and IEEE Xplore – green; **(A)** neural networks AND image processing, **(B)** agriculture AND image processing, **(C)** orchard AND image processing, **(D)** orchard AND networks, **(E)** orchard AND uav, **(F)** orchard AND neural networks AND uav, **(G)** auv AND control AND neural networks, **(H)** uav AND trajectory AND neural networks, **(I)** uav AND U-space, **(J)** agriculture AND uav AND image processing, **(K)** orchard AND uav AND image processing, **(L)** agriculture AND uav AND neural networks, **(M)** orchard AND diseases, **(N)** orchard AND insects, and **(O)** orchard AND fruit production.

from 2017 to 2022, especially when it comes to NNs and UAVs in orchard monitoring. On the other hand, due to the appearance in 2018 of the legislation regarding U-space, no articles on this topic were published until that year. Likewise, papers considered by us to be important and containing the orchard-UAV-neural network triplet did not appear earlier than 2019.

3 UAVs and cameras used for image acquisition in orchard monitoring

UASs including UAVs tend to be the preferred platform for modern orchard monitoring (Zhang C. et al., 2019). UAV is a generic byword for unmanned fixed-wing devices or more usually

multi-rotor copters (multicopters). The latter are often quadcopters (with four motors, the minimum number to ensure simultaneous position and yaw angle tracking, hexacopters (six motors), and octocopters (eight motors with redundancy and increased stability). The drawback for the latter is that they are generally more expensive and require expert handling (due to their larger size and increased velocity any improper use may result in property damage and even accidents).

Each platform comes with its own list of, usually complementary, shortcomings. For example, fixed-wing UAVs have significantly more endurance (flight distance) and, sometimes, payload capacity but lack flexibility because they require a minimum speed to avoid a stall and operate at higher heights. They have traditionally been used for photogrammetry, monitoring, spraying, and data acquisition from large areas (Pederi and Cheporniuk, 2015). On the other hand, multicopters have limited battery life (often in the range of 20 - 30 minutes) but can hover in place and may get quite close to the objects of interest (tens of centimeters, at least when safety measures are deactivated). For these reasons, and due to their comparatively low cost, multicopters are the main tool in small and medium-precision agriculture. A comprehensive classification of multicopters cannot be carried out, but they are mostly divided by their number of motors and whether they are commercial (mainly DJI or Parrot variants) or custom-made for a particular research/application project. Currently, the drones most used for crop monitoring, in particular orchards, are medium or small-sized (adequate for image or sensor data acquisition applications). Larger drones are used for spraying, picking, or planting and are not as widespread yet. Lastly, electric multi-rotor drones are the most popular for orchard monitoring applications as the distances traveled are relatively small, and modern batteries have enough autonomy for this kind of application. For a brief enumeration: popular DJI quadcopter variants are Phantom 3 (Horton et al., 2017; Bouroubi et al., 2018; Apolo-Apolo et al., 2020b; Cheng et al., 2020; Fang et al., 2020; García-Murillo et al., 2020; Barbosa et al., 2021; Menshchikov et al., 2021), Mavic 2 Pro (Barmpoutis et al., 2019; Dong et al., 2020; Nguyen et al., 2021), and Inspire 2 (Häni, 2020) for Mavic Pro 3, and (Mu et al., 2018). The authors in (Zortea et al., 2018) use a GYRO-500X4 quadcopter, and (Torres-Sánchez et al., 2018) use a microdrone MD4-1000. Hexacopters such as the Tarot 960 are used by (Nevalainen et al., 2017). For larger payload capacity and increased stability, octocopters have been used in orchard applications (Abdulridha et al., 2019; Ampatzidis et al., 2019; Horstrand et al., 2019; Deng et al., 2020). Arguably, quadcopter models are the most used in orchard monitoring but hexacopters, even if larger and more expensive, are becoming increasingly popular due to propeller redundancy which leads to better stabilization in nominal functioning and increased reliability under hardware loss. A synthetic description of the kinematic and dynamic models of multicopters is given by (Ju et al., 2022).

Most commercial UAVs have GPS modules that they use as the go-to positioning system for localization in outdoor settings. The specific difficulties for GPS mainly manifest in cities or other areas with challenging vertical features (the "canyon effect", where not enough satellites are simultaneously visible for robust localization). In relatively smooth (i.e., of almost constant height) settings such as orchards, GPS in conjunction with sense and avoidance sensors exhibits acceptable performance, with position errors up to 1 m (Nevalainen et al., 2017). A straightforward improvement is the addition of an RTK module (for those drones which support it). This correction mechanism reduces the errors to 2 cm in planimetry and 3 cm in altimetry (Torres-Sánchez et al., 2018). Noteworthy, RTK modules have mostly deprecated the use of physical targets used for GPS correction (visible elements such as AeroPoints (Johansen et al., 2018), whose position is estimated accurately with a GPS module and is later used as a reference in the images taken by the drone. Examples of UAVs like Phantom 4 (quadcopter) with RTK flighting inside the orchard and fixed-wing UAV flighting over the orchard are given in Figure 4. The research papers that investigate orchard monitoring based on UAVs with different cameras are presented in the synthetic Table 1.

We observe a large variety of cameras and related applications. Although UAVs can be equipped with payloads containing various types of image or video sensors (RGB cameras, multispectral, hyperspectral, thermal, SAR), in orchard monitoring applications the most used are RGB and multispectral (Table 1). Many applications in crop monitoring use small UAVs with included video/photo cameras, without the possibility of attaching other cameras. In the case of larger UAVs, there is the possibility of using different cameras, depending on the requirements. Even if the number and type of UAVs are relatively limited, there is a great variation in the types and numbers of payloads with thermal (Mesas-Carrascosa et al., 2018; Pádua et al., 2020), multispectral (Horton et al., 2017), video (Torres-Sánchez et al., 2018) cameras, or even spectrometers (Ocean Optics (Nevalainen et al., 2017). Relatively recently, cameras with integrated machine learning features have started to appear in UAV applications due to reductions in cost, energy requirements, and weight.

3.1 Characteristics of UAV trajectories in orchard monitoring

For orchard monitoring, the UAV trajectory can be a challenge, because in many applications it can be a 3D trajectory, above and inside the orchard. For a programmed, automatic flight, the lateral distance from the crown of the trees correlated with the protection devices of the UAV creates difficulties in establishing and following the trajectory. Regardless of the trajectory specifics, some parameters are important. Among the most popular are total trajectory time, ground velocity, and flight altitude. As mentioned in (Torres-Sánchez et al., 2018) run times may be significant for terrestrial platforms with respect to UAV limitations. They give the example of an almond orchard where 6.2 km was covered in 1.5 hours (with multiple passes). In general, the UAV velocity is higher compared to ground-based vehicles. (Cheng et al., 2020) gives 5 m/sec for the UAV flight whereas (Dong et al., 2020) runs the UAV at 3 m/sec, and (Mu et al., 2018) consider a speed of 2.5 m/sec. Altitude is also a factor and it may vary significantly, depending on mission specifics: (Dong et al., 2020) mentions 50 m, (Mesas-Carrascosa et al., 2018) 120 m, and (Mu et al., 2018) 30 m. These

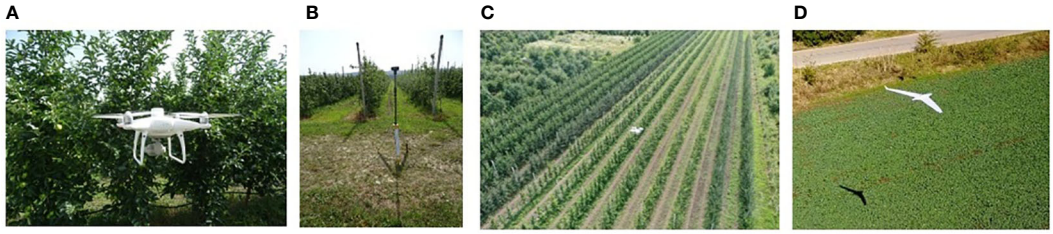


FIGURE 4
(A) Phantom 4 (quadcopter) RTK-flight inside the orchard, **(B)** Fixing the RTK module to the ground, **(C)** Phantom 4 RTK-flight over the orchard, and **(D)** Fixed-wing.

TABLE 1 UAVs with cameras used.

UAV/Type	Camera/Type	References
▪DJI Mavic 2 Pro/ quadcopter (DJI Corporation)	▪ Included: Hasselblad L1D-20c, 20MP/RGB	(Barmpoutis et al., 2019; Dong et al., 2020; Nguyen et al., 2021)
▪DJI Mavic 3 (DJI Corporation)	▪ Included	(Häni, 2020)
▪Phantom 3 Professional/ quadcopter (DJI Corporation)	▪ Included: RGB, Multispectral 5 channels, 12 MP	(Horton et al., 2017; Bouroubi et al., 2018; Apolo-Apolo et al., 2020b; Cheng et al., 2020; Fang et al., 2020; García-Murillo et al., 2020; Barbosa et al., 2021; Menshchikov et al., 2021)
▪Phantom 4, 4 PRO, 4 RTK/ Quadcopter (DJI Corporation)	▪Included: RGB, Multispectral 5 channels, 12 MP	(Lobo Torres et al., 2020; Fuentes-Pacheco et al., 2019; Ampatzidis et al., 2020; Apolo-Apolo et al., 2020a; Gallardo-Salazar and Pompa-García, 2020; Kalantar et al., 2020; Schiefer et al., 2020; Yang, M.-D. et al., 2020; Nguyen et al., 2021)
▪DJI Matrice 100/ quadcopter (DJI Corporation)	▪Different: Logitech C310 webcam, MicaSense RedEdge-M/multispectral	(Hulens et al., 2017; La Rosa et al., 2020; Sarabia et al., 2020)
▪DJI Matrice 210/ quadcopter/possible RTK (DJI Corporation)	▪ Different: Two cameras/RGB -48 MP (Sony Alpha 7) and multispectral 4 channels (Parrot Sequoia)	(Ampatzidis et al., 2020; Jurado et al., 2020)
▪4HSE-EVO/ quadcopter (ITALDRON)	▪ MicaSense RedEdge-M/multispectral	(Adamo et al., 2021)
▪DJI Inspire 1/ Quadcopter (DJI Corporation)	▪Included: RGB	(Hu and Li, 2020)
▪DJI Inspire 2/ Quadcopter (DJI Corporation)	▪Included: RGB	(Mu et al., 2018)
▪DJI Matrice 600/ hexacopter/possible RTK (DJI Corporation)	▪ Different: Zenmuse, Specim FX10, added/ Multispectral 5 channels, Resonon Pika L 2.4 hyperspectral, MicaSense RedEdge-M/multispectral	(Abdulridha et al., 2019; Ampatzidis et al., 2019; Horstrand et al., 2019; Deng et al., 2020)
▪OktoXL 6S12/ octocopter (Mikrokopter)	▪Alpha 7R, Sony/RGB	(Schiefer et al., 2020)
▪eBee Sense Fly/fixed wing (MikroKopter GmbH)	▪Different: Parrot SEQUOIA, Multispectral 4 channels, senseFly S.O.D.A.	(Duarte et al., 2020; Schoofs et al., 2020)
▪Trimble UX5 fixed wing (Trimble.Applanix)	▪Different: RGB and multiple bands	(Adhikari et al., 2021)

altitude values are for top-down observations (photogrammetry missions or disease/humidity monitoring). Flying close to the treetops or even in between tree rows obviously reduces the flight height to 1 m - 5 m. In this context, noteworthy elements which characterize an orchard are row inter and intra-distance. These depend on the type of tree and even on the country. (Cheng et al., 2020) reports 4 m between trees and 5 m between rows in the case of cherry trees and 3 m and 4 m respectively for apple trees. (Dong et al., 2020) mentions spacings of 4 m and 1.5 m (apples) and 4.5 m and 2 m (pears).

Beyond economic or availability factors, various mission specifics may force a particular choice of UAVs. Small/convoluted domains may require aggressive maneuvering which, for fixed-wing UAVs, is very difficult. On the other hand, large fields may lead to battery depletion. This is a major issue since typically a battery takes significantly more time to charge than to discharge. A typical solution is swapping the battery frequently for increased flight duration (a stop where the battery is quickly changed with a full one and the flight is then resumed). These considerations directly influence the choice of trajectory and mission parameters.

Another aspect is the flexibility of the trajectory. The more common approach is to pre-compute the trajectory, couple it with an autonomous sense-and-avoidance system, and then passively track the experiment (the supervisor intervenes only when urgency stop commands are required). Note that typical sense and avoidance mechanisms impose a hard limit of 1 m - 2 m between the drone and possible obstacles. A simple solution can be to adapt the avoidance mechanism and make sure at the supervision level that the drone trajectories accurately avoid the obstacles (tree branches) *via* embedded cameras or RTK-corrected GPS localization.

Not least, and especially for small and medium-sized drones, the presence of wind is a major factor. Thus, flights are often scheduled in periods when the wind is at a minimum. Less common, but still present is the case where flights are determined by the mission particularities. For example, some harmful insects (HH) have a daily cycle which means that they are active (and hence visible) only in the early morning and in periods of reproduction (Leskey and Nielsen, 2018).

While the more interesting missions are those closer to the ground, the most common are still the photogrammetry missions. While conceptually simple, the output of such as mission may be significantly affected by various flight parameters. Beyond those related to resolution (fly height, camera specifications) and mosaic/3D reconstruction (front and sideways overlapping for consecutive images), flight direction, solar irradiation, camera inclination, and whether the pictures are taken time or position-wise, are also relevant (Tu et al., 2020). Thus, most orchard applications reduce to a coverage problem. Beyond the technicalities imposed by the particularities of the problem (Mokrane et al., 2019) enumerate the generic properties that the resulted trajectory must verify: i) cover all points of interest; ii) avoid overlapping routes; iii) avoid obstacles; iv) as much as possible, use simple primitives to construct the trajectory (straight lines and/or arcs of circles).

Most users do not have the knowledge or the desire to design from zero a trajectory generator. There are various local or cloud-based applications that permit interaction with a drone. We may classify these apps depending on the level to which they interact/supersede the original architecture of the drone. Many of them reduce to providing an ergonomic interface that allows defining various simple missions like following waypoints, covering an area with straight parallel lines, etc. It is more challenging to intervene in the actual control scheme and provide direct control actions. For example, in (Horton et al., 2017) the cloud based DroneDeploy is used to construct a flight plan, by interfacing with both GoogleMaps and the drone. Extremely common is the Pixhawk+Ardupilot autopilot controller. This implements all low-level control actions leaving to operator only the task of providing the list of waypoints. Pix4dmapper was used in (Mesas-Carrascosa et al., 2018; Pádua et al., 2020) to triangulate and mosaic the images. (Jensen et al., 2021) uses MoveIt for 3D motion planning and the octomap_mapping package for 3D occupancy grid mapping. ODM (Open Drone Map - https://github.com/OpenDroneMap), in its multiple ports, is an open-source effort that aims to cover the entire workflow of image post-processing for photogrammetry applications.

As stated in the introductory section, due to the increase in the number of drones and flight areas, it is necessary to establish and update relevant flight regulations for UAVs. In Europe, the U-space concept has been formalized through the European Union's U-space Regulation, which was adopted in 2019 and came into effect in 2021. The regulation provides requirements for the design, implementation, and operation of U-space services, including registration and identification of drones, communication protocols, and geo-fencing. The unmanned aircraft system traffic management (UTM) concept is also being developed in other parts of the world (United States), with a range of different approaches being taken. It is safe to say that, in one form or another, a framework of rules and regulations has already taken shape and will govern human-UAV interactions in the future.

3.2 Trajectory design

For most orchard-related missions, the drone does a top-down analysis where the camera is oriented downwards to take pictures while the drone flies in a plane parallel with the horizontal one and at an altitude that is both safe and balances coverage and image resolution. (Ronchetti et al., 2020) provides a list of common altitude values. (Johansen et al., 2018) carries an interesting analysis of tree detection (center position and canopy delineation) in a lychee orchard by changing the height at which the pictures are taken. This is done to find a balance between coverage speed and precision of the estimates. Worth mentioning is that photogrammetry applications usually take photos at a constant sampling time (as a proxy for equal distances between coordinates). Thus, it is important to maintain a constant ground velocity along the flight path. This must be a design requirement at the trajectory generation step and must also be enforced by feedback laws due to the presence of various disturbances. The goal of such missions is often along the lines of photogrammetry in the sense that partially overlapped images are merged (offline, in a computationally intensive effort) into a large-scale map from which various

features of interest are extracted. For example, (Torres-Sánchez et al., 2018) estimate the shape of the tree. Crown volume estimation is carried out by (Torres-Sánchez et al., 2015). Noteworthy, in the latter, the authors mention a root mean square error of 0.39 m for tree height estimation. This may be interpreted as a safety factor for tree-level flights.

One of the few results which explicitly mentions flying at tree level is (Jensen et al., 2021) which implements a three-step run: first, a map of the orchard is created by flying over; second, rows and trees are identified from the acquired images; third, the drone tracks a trajectory between trees. The caveat is that the algorithm was only tested in simulation (within the ROS/Gazebo framework).

From papers that illustrate actual experiments various practical interactions among the UAV components also emerge. For example, (Mesas-Carrascosa et al., 2018) carries out a photogrammetry path planning (straight parallel lines) with emphasis on flight duration due to the need of calibrating the thermal sensors (there is drifting proportional to the duration of the flight). (Mesas-Carrascosa et al., 2018) also proposes to avoid the pre-calibration step by doing it post-flight over the images themselves and by carrying an in-flight drift correction for microbolometer thermal sensors.

Of course, the most important element for rotary drones is battery life. Their increased flexibility comes at the price of significantly less autonomy than in the case of fixed-wing UAVs. Hence, energy efficiency is paramount in trajectory design and influences mission planning at all stages. This may mean proposing very simple trajectories: straight lines as in (Mesas-Carrascosa et al., 2018) or a grid pattern as in (Mu et al., 2018). Usually, the UAV dynamics are ignored when assessing battery consumption (Furchì et al., 2022). Still, the drone behavior and type of trajectories employed can have a disproportionate effect on battery life. (Pradeep et al., 2018) provides a first principles approach to quantify consumption for the climb, cruise, and descent phases (with application to a DJI Phantom 4 quadcopter).

From a dynamics viewpoint "trajectory" means that both position and attitude must be specified at each moment of time during the flight. Except for laboratory/experimental setups, this is hidden by the embedded control software of the drone. Rather, the end-user simply gives a list of waypoints from which the drone's control mechanism designs a suitable trajectory. Choosing the waypoints that define a path is quite challenging, depending on the mission complexity. In such cases, often heuristic and graph-based methods are employed. For example, (Ochoa and Guo, 2019) combine a genetic algorithm (to determine way-point locations) with the Dijkstra algorithm (for path construction).

Many times, there are multiple flights carried during the same mission. Often, the first flight is for sensor calibration, an update of position information, and an update of the environment's map (new features of interest, changes in positioning, etc.). Only in the subsequent step, the actual flight (the one where data is gathered) is done. Thus, a typical workflow is as the one from (Horstrand et al., 2019):

i. initial flight to assess the environment,
ii. planning step on the flight management system (choose waypoints, area of interest, etc.),
iii. start the way-point tracking and supervise the UAV during its flight, with the possibility to update path/sending "turn to base" commands.

In the case of modern orchards, for UAV navigation inside the orchard, among the rows of trees, the orchard can be modeled as an aisle graph (Sorbelli et al., 2022) so that the images are collected as efficiently as possible. In this case, it is about collecting images to detect some harmful insects on trees. Most if not all graph-based methods are based on variations of the Traveling Salesman Problem (TSP). (Furchì et al., 2022) uses a Steiner TSP implementation where only a subset of the nodes must be visited. The paper is also noteworthy for considering battery usage and integrating it as a weight for the graph edges.

In general, formulating decision problems (graph-based or otherwise) for efficient orchard travel leads to a difficult optimization problem. Authors (Furchì et al., 2022) provided a mixed-integer formulation that makes use of binary variables to characterize decisions in the problem (which node is next, which path is followed from a given list, etc.). Such methods are prone to numerical issues and quickly become impractical for real-time implementations. The usual approach is then to simplify the problem and solve it to a sub-optimal solution. In this case, the computation time reduction is significant and the loss in performance is negligible. The heuristic methods employed are usually based on evolutionary procedures or greedy algorithms.

3.3 Trajectory tracking

Most agricultural UAV applications give the trajectory as a list of waypoints with associated actions. For example, the API (programming interface) of DJI drones allows by default to give a list of up to 100 waypoints and to associate up to 15 actions for each of them (camera focus, take an image, start/stop the video, etc.). The actual trajectory (path and input actions) is computed onboard the UAV by the autopilot. At this level, further restrictions may be considered (from the sensor and avoidance module, limitations on control actuation, etc.) which will affect the path's shape. Lower-level interactions are usually relegated to experimental drones used in research laboratories (Parrot Mambo or Crazyflie nano-drone, NXP HoverGames for mid-sized drones, etc.).

Any path-tracking algorithm is as good as the quality of information that it receives (Li, J.-M. et al., 2021). Usually, GPS (possibly corrected by an RTK module) information is employed. Albeit ubiquitous in recent years, GPS may be replaced or supplanted by other approaches. (Emmi et al., 2021) fusions information from 2D Lidar and RGB cameras to identify key locations and working areas which are next integrated into a semantic layer where the various features of interest have certain types (lane, alley, etc.) among which the UAV transitions. The authors in (Stefas et al., 2016) present a vision-based approach for UAV navigation within an orchard. Both the monocular and

binocular cases are analyzed. For the former, additional information about the structure of the orchard rows is used and for the latter, a depth-perception algorithm is implemented. In (Hulens et al., 2017) the vision-based approach also makes use of the orchard characteristics: the feasible path is determined by first detecting the center and end (the vanishing point) of the current corridor.

4 Neural networks used for orchard monitoring

The use of artificial intelligence and especially NNs for image processing in various fields of agriculture has led to a significant improvement in performance in tasks of detection, segmentation, and classification of regions or objects of interest. Thus, from the investigated researched papers, an improvement in orchard monitoring performances can be noted by NNs in the processing of orchard images. From Figure 3 it can see that the number of research papers that study the use of NN in orchard monitoring increased in the interval 2017-2022. Most of the NNs in the analyzed papers in this study used RGB images and few multispectral images as in (Kerkech et al., 2020).

4.1 Series of neural networks and their representants for image processing in orchard monitoring

Because orchard monitoring involves high-level image processing functions in various conditions, the NNs used in orchard monitoring for image processing were very diverse. Most often, the classification can be used for a special segmentation based on pixel classification named semantic segmentation. The name of the used NNs is explained in the list of abbreviations (Annex 1). The NNs for object detection, classification, and segmentation functions (including semantic segmentation) used in the investigated references are presented in Table 2. In some applications, the NNs from popular series, having small structural changes, got the names of respective applications like VddNet - Vine Disease Detection Network (Kerkech et al., 2020) and MangoYOLO (Koirala et al., 2019a)

The most used NNs were those from series R-CNN (Region-Based CNN) (Girshick et al., 2014), YOLO (You Only Look Once) (Redmon et al., 2016), U-Net (Ronneberger et al., 2015), ResNet (Residual Neural Network) (He et al., 2016), and SegNet (Semantic Segmentation Network) (Badrinarayanan et al., 2017). The basic structures of these important series are given in Figure 5. Among them, the YOLO-type NNs had the greatest growth trend. Details regarding the architectures and layers of the most used NNs in image processing for object detection, classification, and segmentation are given by (Alzubaidi et al., 2021; Bhatt et al., 2021). An interesting review (Nawaz et al., 2022) presents the detection of objects in multimedia using NNs, considering single-stage detection and two-stage detection algorithms. The advantages and disadvantages related to precision, complexity, and speed of operation, in various applications such as object detection, multi-object detection, and real-time object detection, were highlighted. The analyzed networks (proposed until 2020) were those from the YOLO, SSD, and RetinaNet series, for the single-stage algorithm, and R-CNN for the two-stage algorithm. Representatives from these series can also be found in the references analyzed in this paper.

R-CNN which is based on a two-stage algorithm for object detection has two important representatives: Faster R-CNN (Ronneberger et al., 2015) and Mask R-CNN (He et al., 2017) which share significant commonalities. Faster R-CNN provides two pieces of information for each candidate object, the classification (class label) and the bounding box (regression). Mask R-CNN extends Faster R-CNN by providing three pieces of information at the output: the class (C), the bounding box (B), and the segmentation mask (M) for each selected region of interest. For the latter, a branch (pixel-to-pixel alignment) is added in parallel in the Faster R-CNN structure. Since this branch has reduced additional computational effort, the network remains quite fast.

Both Faster R-CNN and YOLO are detection networks with object detection accuracy between 63.4% and 70% (Diwan et al., 2022). The YOLO series including several variants (like YOLO v1, v2, v3, v4, v5, v6, v7, v8, YOLOX, etc.) are networks in one stage, and for this reason, they are much faster than Fast R-CNN or Faster R-CNN which are detectors in two stages. Object detection in this case is seen as a regression problem and not a classification one. The areas of interest (objects) are identified, and their positioning is established by a bounding box associated with the probability of belonging to a class.

Faster R-CNN (Figure 5A) is a two-stage object detection algorithm providing the bounding box and classification. It can be successfully used for fruit detection in the natural environment in difficult conditions and positions (leaf occlusion, fruit occlusion, fruits in shadow, and different light exposure). A challenge in fruit detection is the great number of fruits (sometimes overlapping) in an input image. Also, it can be used for the detection of diseases and insect pests on fruits.

Mask R-CNN (Figure 5B) is like Faster R-CNN and adds to the output a binary mask for segmentation of the detected object. It gets the region where the fruit is located. It can detect and segment fruits in the natural environment (apples, pears, citrus, logan fruit bunches, etc.) in difficult conditions and positions. It was used for the identification and segmentation of trees in orchards from aerial imagery (orthophoto maps).

YOLO is a single-stage object detection algorithm providing the bounding box and classification. It is composed of four sections – input, backbone, neck, and prediction – which allow the detection and localization of objects of different sizes (including small objects) in orchards, like fruits, flower clusters, and insects. It can detect and identify fruits in the natural environment (apples, pears, citrus, logan fruit bunches, etc.) in difficult conditions and positions (covered by leaves, fruits in shadow, fruits at different distances from the camera, and fruit cluster) with precise box location and high accuracy. The various variants of YOLO networks consider a compromise between speed, accuracy, and simplicity. Many of them can be implemented directly on the UAV, for real-time applications simultaneously with video acquisition. The structure of the well-known YOLO v5 is presented in Figure 5C.

TABLE 2 NNs used in orchard monitoring (C, classification; D, Detection; S, segmentation or semantic segmentation).

NN series	Representatives/configuration	Function	References
▪**CNN**	▪CNN simple	▪C	(Kestur et al., 2019; (Kim et al., 2020; Li, Y. et al., 2020; (Csillik et al., 2018; Zortea et al., 2018; Lei et al., 2022)
	▪Multi-layer perceptron	▪D	(Nevalainen et al., 2017; Fernandez-Gallego et al., 2018)
	▪Sandglass bottleneck	▪C	(Chen, T et al., 2021)
	FCN	▪S	(Marmanis et al., 2016; Osco et al., 2021)
	▪CaffeNet	▪C	(Bouroubi et al., 2018)
▪**DaSNet**	▪DaSNet-A, DaSNet-B, DaSNet-C, DaSNet-v2	▪D, S	(Kang and Chen, 2019; Kang and Chen, 2020a)
▪**DeepLab**	▪DeepLab-ResNet	▪D, S	(Dias et al., 2018)
	▪Deep-LabV3 +	▪S	(Osco et al., 2021; Li, D. et al., 2022; Zhang X. et al., 2021)
▪**DensNet**	▪DensNet 121	▪D, C	(Nguyen et al., 2021; Peng et al., 2023)
▪**Encoder - Decoder**	▪CED-Net	▪D	(Kerkech et al., 2020)
	▪Spatial Pyramid- oriented Encoder-Decoder Cascade CNN	▪S	(Yuan and Choi, 2021)
	Staked Autoencoder	▪D	(Deng et al., 2020)
	▪VddNet with three autoencoders (Vine Disease Detection Network)	▪D	(Kerkech et al., 2020)
▪**FCRN**	▪FCRN	▪D	(La Rosa et al., 2020)
▪**GoogLeNet**	▪ Inception modules	▪C	(Breslla et al., 2020)
▪**HRNet**	▪HRNet	▪D, C, S	(Biffi et al., 2021)
▪**Inception**	▪Inception v3	▪C	(Fang et al., 2020; Hansen et al., 2020; Zhang, H. et al., 2019)
▪**LeNet**	▪LeNet5	▪C	(Kerkech et al., 2018; Kerkech et al., 2020)
▪**LedNet**	▪LedNet	▪S	(Kang and Chen, 2020b)
▪**RBF**	▪RBF/RBF+KNN	▪D	(Fernandez-Gallego et al., 2018; Abdulridha et al., 2019)
▪**R-CNN**	▪R-CNN	▪D	(Zhang et al., 2018; Biffi et al., 2021)
	▪Faster R-CNN	▪D	(Ren et al., 2017; Apolo-Apolo et al., 2020a; Apolo-Apolo et al., 2020b; Biffi et al., 2021; Barmpoutis et al., 2019; Cunha et al., 2021; Khan et al., 2021 Deng et al., 2022; Hu et al., 2022)
	▪Mask R-CNN	▪D, S	(He et al., 2017; Barmpoutis et al., 2019; Jia et al., 2020; Machefer et al., 2020; Santos et al., 2020; Iqbal et al., 2021; Zhang, W. et al., 2022)
	▪Libra R-CNN	▪D	(Biffi et al., 2021)
	▪Cascade R-CNN	▪D	(Biffi et al., 2021)
▪**ResNet**	▪ResNet 18	▪C	(Zhang et al., 2021; Zhang, X. et al., 2019)
	▪ResNet 50	▪C	(Fang et al., 2020; Park et al., 2020; Nguyen et al., 2021)
▪**RetinaNet**	▪RetinaNet	▪D	(Culman et al., 2020)
▪**SegNet**	▪SegNet	▪S	(Fuentes-Pacheco et al., 2019; Ochoa and Guo, 2019; Majeed et al., 2020; Menshchikov et al., 2021; Osco et al., 2021)
▪**SqeezeNet**	▪SqeezeNet	▪C	(Park et al., 2020; Nguyen et al., 2021)
▪**SSD**	▪SSD	▪D	(Aota et al., 2021)
	▪SSD with FSAF module	▪D	(Biffi et al., 2021)
▪**UNet**	▪Simple UNet	▪D, S	(Oliveira et al., 2019; Lin and Guo, 2020; Menshchikov et al., 2021; Osco et al., 2021)

(Continued)

TABLE 2 Continued

NN series	Representatives/configuration	Function	References
	▪UNet with SE-ResNeXt-50 as encoder	▪S	(Liu et al., 2021; Shang et al., 2021)/
	▪UNet with VGG-16 encoder	▪D, C, S	(Fawakherji et al., 2019; Kattenborn et al., 2019)
▪**VGG**	▪VGG16	▪C	(Park et al., 2020; Nguyen et al., 2021)
	▪VGG19	▪C	(Fang et al., 2020; Miyoshi et al., 2020)
▪**Xception**	▪Xception	▪C	(Fang et al., 2020)
▪**YOLO**	▪YOLOv2/improved	▪D	(Santos et al., 2020)
	▪YOLOv3/improved	▪D	(Ampatzidis et al., 2019; Li, J.M. et al., 2021; Liu and Wang, 2020; Santos et al., 2020; Chen, C.J. et al., 2021),
	▪YOLOv3/Tiny	▪D	(Chen, C.J. et al., 2021)
	▪YOLOv4	▪D	(He et al., 2020; Li D. et al., 2021; Lin et al., 2022; Popescu et al., 2022b)
	▪YOLOv5	▪D	(Li, D. et al., 2022; Lyu et al., 2022)
	▪YOLO BP	▪D	(Zheng et al., 2021)
	▪YOLOF-snake/ResNet101 as backbone	▪D, S	(Jia et al., 2022)
	▪YOLOX	▪D	(Zhang, Y. et al., 2022)
	▪YOLOP	▪D	(Sun et al., 2023)

U-Net (Ronneberger et al., 2015) series is especially important in image segmentation. Although U-Net networks have good segmentation accuracy, they can be trained with relatively few images. In a classic way, the network architecture is made up of two paths (subnets), the first one is contraction type (encoder) and the second one is expansion type (decoder). At each level of the two paths, there are concatenations (skip connections) between the up-sampling of the feature map and the corresponding down-sampling of the feature map. In the new improved versions of the network, various NNs are placed on the encoder as blocks instead of the original ones. Examples of such improved U-Net are given by (Bhatnagar et al., 2020), having ResNet 50 as a backbone, and (Liu et al., 2020), having SE-ResNeXt 50 as a backbone. The basic U-net architecture is presented in Figure 5D. Variants of U-Net were used in important applications like the segmentation of trees in the orchard and collecting orchard environment information from UAV images, segmentation of plantation cover area, segmentation of diseased plants and pests, and mapping of the tree species.

ResNet, the winner of the ILSVRC 2015 competition (He et al., 2016), introduced the elements of shortcut connections, within layers providing multi-layer connectivity. As a result, it has a lower computational complexity. Depending on the number of layers ResNet has more representatives: ResNet 18, ResNet 34, ResNet 50, ResNet 101, ResNet 110, ResNet 152, ResNet 164, and ResNet 1202. The most used type in the investigated papers was ResNet50 containing 49 convolutional layers and one FC layer (Alzubaidi et al., 2021). For example, the ResNet network from Figure 5E (Ichim and Popescu, 2020) was used to detect flooded zones in an area with vegetation (crops), the meaning of the notations (to save space) being the following: A and B— skip connections, repetitive modules, FC—fully connected layer, F—flood type patch, V—vegetation type patch, and n—number of module repetition). The image was partitioned into patches according to a specific algorithm and each patch (of small size) was classified/segmented as being flood or vegetation. This decomposition into patches can also be used to detect small objects (e.g., insects) compared to the whole image.

The SegNet network (Figure 5F) introduced in 2015 (Badrinarayanan et al., 2017) is like an encoder-decoder structure that, in the final stage, has a pixel-wise classification layer. Each layer in the encoder has a corresponding layer in the decoder. Finally, the multi-class soft-max classifier provides for each pixel a probability of belonging to a class, being thus possible a semantic segmentation of the regions of interest (RoIs). It was used in applications like tree localization and classification from aerial imagery, estimation of trees density (large-scale orchard monitoring), segmentation of trunks, branches, and trellis wires (orchard of trees on trellis wires).

As we mentioned, when the databases were unbalanced or the images collected from the orchards were insufficient, some authors used data augmentation techniques such as translations, rotations, transposition, rescaling, reflections, or changing the intensities on color channels. Usually, techniques for image preprocessing, size reduction, or cropping smaller windows were also used before entering the NNs.

In many applications, it has been proven that deep CNNs (DCNNs) can learn the invariant representations of images (as in the case of supervised learning) and can achieve performance at the level of human observers or even better (Khan et al., 2020b). They can also extract useful representations for unlabeled images (unsupervised learning). More recently, they can also be learned

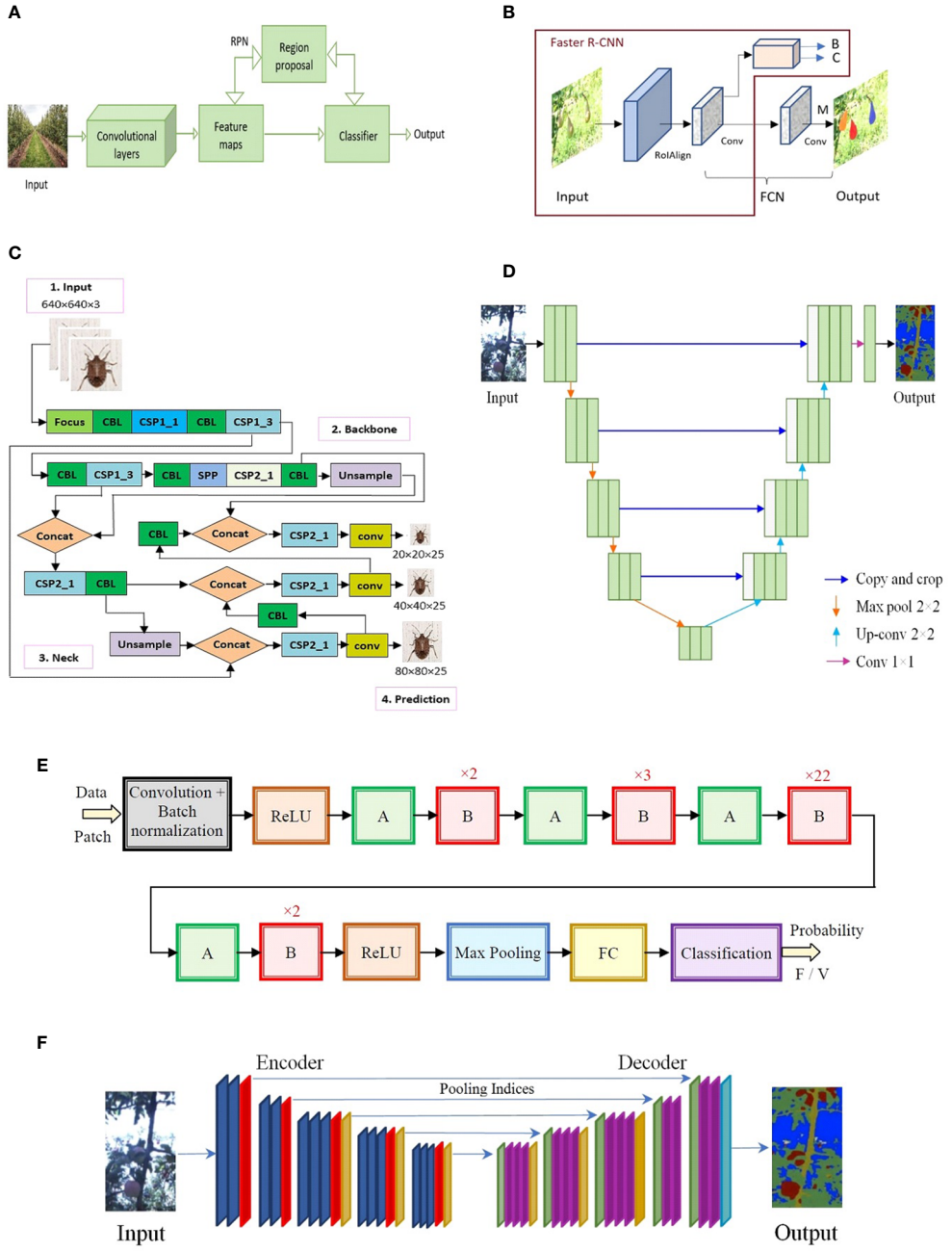


FIGURE 5
(A) Faster R-CNN, **(B)** Mask R-CNN (B-box bounding, C-class, M-mask), **(C)** YOLO v5 (Popescu et al., 2022), **(D)** U-Net architecture, **(E)** ResNet (Ichim and Popescu, 2020), **(F)** Segnet.

effectively through the reinforcement learning method (Arulkumaran et al., 2017) and federated learning (Deng et al., 2022). For example, in the review paper (Wang C. et al., 2022) the authors analyzed the CNN use throughout the fresh fruit production chain and evaluation: flowering, growth, and picking (using ground or aerial platforms). Another important aspect is the fact that modern NNs are pre-trained, for example on the ImageNet (Deng et al., 2009) and PASCAL VOC (Everingham et al., 2015) databases, making the transition to the desired concrete application much easier and faster, with fewer training images.

The use of NNs involves three distinct phases: training, validation, and testing. The images from the available data set (including those obtained by augmentation) must be randomly divided between these three phases. The proportion is 70% - training, 20% - validation, and 10% - testing. The validation phase is used in some works to establish network confidence levels for collective intelligence (Popescu et al., 2022a) or decision fusion systems (Ichim and Popescu, 2020). Sometimes the testing phase is abandoned and then the proportion is 80% - training and 20% - validation.

4.2 Software used

Different software libraries and modules (most of them free) are used for image processing in successive tasks like obtaining orthomosaic, georeferenced maps, 3D models, machine learning, image annotation, implementing deep neural networks, etc. To obtain useful information for tree canopy extraction and segmentation, the images acquired by UAVs must be processed with various software, for example, Agisoft Photoscan (https://www.agisoft.com/) to generate geo-referenced ortho-images (Kerkech et al., 2020; Adhikari et al., 2021). To implement the NN models the most used software and platforms were TensorFlow (https://www.tensorflow.org/), PyTorch (https://pytorch.org), and Keras (https://keras.io/). An important step in the learning and testing phases is image annotation. There is different software as image annotator like YOLOLabel for the YOLO series (Iqbal et al., 2021; Yuan and Choi, 2021) and VGG Image Annotator (Biffi et al., 2021).

4.3 Datasets

The databases used in the analyzed papers are divided into two groups: a) databases for learning/validating/testing NNs for the detection/classification/segmentation of objects of interest from the images acquired in the orchard and b) databases for configuring flights of photogrammetry or inside the orchards to collect data (images).

A pertinent presentation of public image databases for use in precision agriculture is made in (Lu and Young, 2020) which contains 34 such databases. Of these, 11 refer to orchards: DeepFruits, Orchard fruit, Date fruit, KFuji RGB-DS, MangoNet, MangoYOLO, WSU apple, Fuji-SfM, LFuji-air, MinneApple, and Apple Trees. They are created manually or by ground vehicles. Most are based on RGB images. Many times, augmentation, annotation, and sharing operations can be performed on the images from the databases when used in NNs. The augmentation operations, often necessary in the learning stage to establish the most correct parameters and weights, are not used in the validation or testing stages. To obtain correct training of NN sometimes the data set must carefully filter because it can contain errors. For example, the IP 102 dataset (Wu et al., 2019), with more than 75,000 images for pest detection, was filtered to obtain better results. The filtered dataset, HQIP102, containing 47,393 images of 102 pest classes on eight crops was used (Peng et al., 2023) to train and test NN for pest detection.

To be sure that the trained NNs will learn the main characteristics of the objects to be detected or classified and will be more robust in a natural environment such as the orchard, many researchers have performed data augmentation starting from the original data. For example, 15 different augmentation methods are mentioned in (Lei et al., 2022), such as Gaussian noise, impulse noise, out-of-focus blur, motion blur, zoom blur, elastic transformation, rotation transformation, random erase, random crop, random flip, fog, brighten, contrast, color dithering, and pixelated. To obtain good results on NN training, the classes in the dataset need to be balanced and annotated. In the case of data imbalance, the authors (Peng et al., 2023) proposed an efficient data augmentation based on a dynamic method that depends on the initial number of elements in each class. In addition to these classic augmentation operations, synthetic augmentation operations using NNs for generating new images such as GAN are also used lately (Lu and Young, 2020).

The applications studied through this manuscript often require large datasets for the training/validation of NNs. Unfortunately, these resources are not always well-defined or are restricted. There are also some exceptions such as (Torres-Sánchez et al., 2018) which list several point cloud collections.

The advantages of automatic analysis and labeling from UAV images are particularly important (Zortea et al., 2018): one day for automatic image labeling compared to one month for manual labeling in the field with a GPS locator and one week for manual labeling of images obtained from a UAV flight. To label manually, efficient software assisting tools were developed like labelImg used for annotation in the MangoYOLO dataset and VIA (VGG Image Annotator) used for annotation in the MinneApple dataset. Most datasets are created for image processing, classification, and segmentation inside the orchard with machine learning tools, but there are also datasets for photogrammetry applications, for example, the ODMdata page (https://github.com/OpenDroneMap/ODMdata) which contains a large collection of various data sets with open access (orchards, forest areas, parks, etc.).

It is worth mentioning that most identified databases deal with photogrammetry applications or, at most, with production estimation (fruit counting). In other words, there are no UAV collections that provide close-up images (to identify visually small bugs or morphology changes at the leaf level). In most papers, own data sets, specific to the application, were used, but there are also papers that were limited to public databases (Table 3).

4.4 Statistic performance indicators

Considering the results obtained from the experiments, the analyzed papers used the following elements that make up the confusion matrix (error matrix): true positive cases (TP), true negative (TN), false positive (FP), and false-negative (FN). Based on them, a series of statistical quality indicators were calculated for the assessment of detection, classification, or segmentation operations: Specificity (SPE), Sensitivity (SEN), Precision (PRE), Accuracy (ACC), Dice coefficient (F1 score) (DSC or F1), and Jaccard index (Table 4). If the application refers to several classes, many authors prefer to provide average values for DSC and ACC in all classes.

In addition to these indicators, Intersection over Union or Jaccard index (IoU) was used to assess detection and segmentation. Mean Average Precision (mAP) is a statistical indicator used to evaluate the performance of NN for object detection. It is calculated as an average over the number of classes n of APi entities that represented the average detection accuracy for class i (Table 4). The mAP is calculated for different IoU thresholds. In the case of evaluating the correctness of the detection and counting of several objects in the image (for example, in the case of instance segmentation), some papers used Capturing Rate (CR),

TABLE 3 Public datasets used.

Dataset name	Characteristics	Year	Number of images	Link	References
COCO-Stuff	Contains pixel-level annotations of classes such as grass, leaves, tree, and flowers	2017	123,287 images, 886,284 instances	https://cocodataset.org/#download	(Caesar et al., 2018; Dias et al., 2018)
AppleA, AppleB,	Datasets containing apples, peaches, and pears	2018	207 images	https://data.nal.usda.gov/dataset	(Dias et al., 2018; Dias et al., 2018)
MinneApple	Benchmark dataset for apple detection, segmentation, and counting in the orchard	2019	1,000 images with 40,000 annotated objects	https://rsn.umn.edu/downloads	(Häni, 2020)
IP102	Contains 102 pest classes on eight crops.	2019	more than 75,000 images	https://www.kaggle.com/datasets/rtlmhjbn/ip02-dataset	(Wu et al., 2019), (Peng et al., 2023)
Mango YOLO	Image dataset acquired with a farm terrestrial vehicle for train, testing, and validation	2019	1730 images	https://figshare.com/articles/dataset/MangoYOLO_data_set/13450661/2	(Koirala et al., 2019a)
Mendeley Data (dataset added)	Image dataset acquired from a UAV over an experimental site; added to Mendeley	2020	314 images	https://data.mendeley.com	(Encinas-Lara et al., 2020)
Pistachio Dataset	Pistachio orchard with two different nadir angles	2021	248 images	https://doi.org/10.5281/zenodo.7271542	(Vélez et al., 2022)

Detection Rate (DR), and Statistical Rate (SR) calculated based on the actual number of objects, the number of objects in the image and the number of objects detected by the computing system in the same image. Another indicator worth mentioning is the Coefficient of determination (R- squared), calculated from the sum of squares of residuals (SSE) and the total sum of squares (SST).

Also, learning time and operating time are considered. These time indicators strongly depend on the networks, the hardware used (CPU, GPU, computer cluster, etc.), the resolution, and the number of images.

4.5 New trends in the implementation of neural networks for orchard monitoring

The novelties of the recent papers in the analyzed field refer to the combination of several networks into decision systems to obtain better performances than the component networks, including a CNN as the backbone in other CNN (network in a network), the improvement (adaptation) of some networks for the respective application - hence the name of the network, and the improvement of well-established high-performance networks. The new trends in the use of NNs in orchard monitoring follow the general line regarding either the improvement of existing networks by optimizing resources and improving performance or by combining several NNs in network ensemble models. In this case, it can be noted either the decision of the global system through the majority vote of the decisions of the individual networks or through the weighted summation of the detection (or classification) probabilities offered by each component network of the ensemble. The weight of a network is assigned proportionally to its performance. To select the best NNs relative to an application, some papers present comparisons regarding the values of the performance indicators of several top NNs. Thus in (Torres-Sanchez et al., 2020) SegNet, U-Net, FC-DenseNet, DeepLabv+ Xception, and DeepLabv3+ MobileNetV2 are compared regarding

TABLE 4 Statistic performance indicators used in the review.

Indicator	Formula	Indicator	Formula
▪Specificity	$SPE = \frac{TN}{TN + FP}$	▪Sensitivity (Recall)	$SEN = \frac{TP}{TP + FN}$
▪Precision	$PRE = \frac{TP}{TP + FP}$	▪Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$
▪Dice coefficient (*F*1-score or simple *F*)	$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$	▪Jaccard index (In confusion matrices)	$J = \frac{TP}{TP + FN + FP}$
▪Intersection over Union or Jaccard index	$J(A,B) = IoU = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$	▪Mean Average Precision	$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i$
▪Coefficient of determination (R- squared)	$R^2 = 1 - \frac{SSE}{SST}$	▪Capturing rate (*CR*)	$CR = \frac{captured\ \ objects}{real\ \ objects}$
▪Detection rate (*DR*)	$DR = \frac{detected\ \ objects}{captured\ \ objects}$	▪Statistical rate (*SR*)	$SR = \frac{detected\ \ objects}{real\ \ objects}$

tree segmentation from UAV images. The obtained performances by (Zhang and Zhang, 2023) were ACC: 88.9 – 96.7%, F1-score: 87 – 96.1%, and IoU: 77.1 – 92.5%. These networks can be combined into ensemble systems for better detection (Deng et al., 2021; Popescu et al., 2022a).

For areas with several orchards and different conditions, for unitary management regarding several diseases and insect pests, the authors (Deng et al., 2022) proposed a federated learning method of NNs from several sources (obviously, several UAVs). In this way, if an orchard has unbalanced or insufficient data for a disease/pest, then the data is compensated from the other orchards, resulting in better learning. For example, the improved Faster R-CNN model by (Deng et al., 2022) can recognize fruit diseases and insect pests under occlusion.

The popular networks were modified to improve their performances. In (Zhang and Zhang, 2023) an improved U-Net, namely MU-Net was implemented to segment the plant diseased leaf. A residual block (Resblock) and a residual path (Respath) were introduced into U-Net to overcome gradient problems and, respectively, to improve the feature information between the two paths of U-Net. For better performances on pest classification, DensNet 121 was improved (Peng et al., 2023) in three directions: input information feature, channel attention technique, and adaptive activation function. Each improvement creates a modified DensNet 121 model. The three models are combined into an ensemble and the final decision is based on the sum of the normalized confidence values for each pest category on these three NNs.

By simultaneously considering RGB and NIR images, more precise information can be obtained about the health of plants, including orchards or vineyards. For example, in (Kerkech et al., 2020) multimodal images (visible and infrared) are used for disease detection in grapevine crops. Patches of 360 × 480 pixels were cropped and analyzed from the original images (4608 × 3456 pixels). Two channels are selected green and NIR and the regions of interest are segmented on both channels. For the dataset, semi-automatic labeling was used in two steps: LeNet 5 and manual correction. Four classes are considered: shadow, ground, healthy, and symptomatic vine. Two SegNet models were evaluated and tested for segmentation in RGB and NIR channels. The symptomatic cases are interpreted considering the fusion by intersection and union of segmentations obtained by the two networks. The recommendation is to consider a system with more NNs.

Some common NNs were adapted for a specific application and got the name of the application: Vine Disease Detection Network (VddNet) (Kerkech et al., 2020), YOLO designed for mango fruit detection (MangoYOLO) (Koirala et al., 2021), network to detect the invasion degree of Solanum rostratum Dunal (DeepSolanum-Net) (Wang et al., 2021).

A synthesis of the new trends of UAVs and NNs in the orchard monitoring context between 2020 and 2022 is done in Table 5. The trend of most used NNs as number of appearances in research papers between 2019–2022 were represented in Figure 6A. The symbol * marks the fact that at the time of writing the article, the Web of Science indexing for the year 2022 has not finished. An average of the main performance indicators is represented by the graph in Figure 6B. It can see that both ACC and F1 have an increasing trend, which means obtaining better-performing solutions.

5 Applications

In recent years, more and more tasks related to the monitoring of orchards in large areas are solved by the intelligent processing of data, and especially of images, collected with the help of drones. Most applications related to the use of UAVs and NNs in orchard monitoring refer to orchard mapping, pest and harmful insect detection, fruit detection, yield estimation, and orchard condition. In an automatic inspection of the orchard, for the desired application, the appropriate trajectory of the UAV must be specified and designed, according to Section 3. A major element in orchard surveillance is identifying regions or objects of interest. This may be at the macro level (orchard, tree lines, boundaries), medium level (corona shape estimation, tree center, and height identification), or micro level (counting fruits, pest detection, or insect detection). As expected, there is a large variety of approaches and tools to solve such problems. For example, (Torres-Sánchez et al., 2018) discusses canopy area, tree height, and crown volume. Noteworthy, the crown shape may vary even for the same type of tree (as remarked by (Mu et al., 2018) for peach orchards). Common geometric shapes considered for the crown shape are the cone, hemisphere, and ovoid (Torres-Sánchez et al., 2018). The precision of the estimation varies and strongly depends on the flight characteristics and camera performance (Gallardo-Salazar and Pompa-García, 2020).

As was mentioned in Section 4, there are cases where the networks take the name of the specific application. For example, the authors (Kestur et al., 2019) proposed a deep convolutional neural network architecture for mango detection using semantic segmentation named MangoNet. Also, the authors (Koirala et al., 2021) call the network YOLO used MangoYOLO, and (Sun et al., 2023) named YOLOP the modified YOLO v5 for pear fruit detection. The authors (Kerkech et al., 2020) proposed a deep convolutional neural network architecture for vine disease detection named VddNet with a parallel architecture based on the VGG encoder.

In the case of orchard monitoring using UAVs and NNs, there are several essential applications such as the detection and segmentation of orchards and individual trees, the detection of tree diseases, the detection of harmful insects, the identification of fruits and the evaluation of production, or the development of the orchard.

5.1 Orchard and tree segmentation

The mapping and segmentation of the orchards as well as the trees inside was the subject of many research articles from the analyzed period. Crop tree detection, location, and counting are estimated by (Sarabia et al., 2020; Dyson et al. 2019; Lobo Torres et al., 2020;

TABLE 5 A summary of new trends for the orchard-UAV-NN triplet.

Model Novelty	Characteristics, Pros, and Cons	NN used and function	Performance indicators	References
▪Combining two different CNNs	▪Semantic segmentation of vegetation. ▪Pros: Good results in a wetland mapping application. ▪Cons: Slower training process.	▪SegNet with VGG16 ▪SegNet with ResNet50 ▪UNet with VGG16 ▪UNet with ResNet50	▪ACC = 91% for SegNet with ResNet50 ▪Time for NN training: 700 min	(Bhatnagar et al., 2020)
▪Fusing the outputs of two CNN, one for RGB and the other for NIR images	▪Two camera sensors for RGB and NIR. Disease detection in vine crops using segmentation ▪Pros: Fusion by intersection is better than classes detected in the visible or infrared range: ▪Cons: Reduced performances on segmentation due to the small training set and too few NNs in the system, long runtime	▪Two SegNet (RGB and NIR) ▪Two LeNet5 (RGB and NIR) for pre-labeling	▪Leaf-level average ACC: 82.20% - fusion AND; 90.23% - fusion OR; ▪Grapevine-level average ACC: 88.14% - fusion AND; 95.02% - fusion OR;	(Kerkech et al., 2020)
▪Net with a specific name for the application: DeepSolanum-	▪Segmentation of UAV images to detect the invasion degree of "Solanum rostratum Dunal" ▪Pros: Reduced training time and complexity ▪Cons: Performances must be improved	▪DeepSolanum-Net based on U-Net	▪Precision = 89.95% ▪Recall = 90.3% ▪IoU = 82.76% ▪F1-score = 89.85%	(Wang et al., 2021)
▪Different CNN combined in a system for orchard monitoring ▪Net with a specific name: MangoYOLO	▪Detect and count the fruits within images. Input: tree image. Output: total fruits per tree ▪Pros: Good performance for fruit counting in one season. ▪Cons: It is not a robust model in different seasons.	▪Multi Layered Perceptron (MLP), ▪MangoYOLO model, ▪Xception_count model with a regression block, ▪Xception_classification model	▪Best R^2 = 94%	(Koirala et al., 2021)
▪Including a CNN as a backbone in other CNN	▪Detection and semantic segmentation of coconut trees ▪Pros: Good ACC ▪Cons: Need to classify and locate different kinds of trees.	▪Mask R-CNN with ResNet 101 as a backbone	▪mAP = 91% ▪ACC (classification) = 97%	(Iqbal et al., 2021)
▪Dual network-based system to eliminate successively some FN and FP errors	▪Detecting and classifying harmful insects in orchards (HH) ▪ Pros: Good performance to detect insects in the foreground. ▪ Cons: Need to detect insects in a distant plane.	▪YOLOv.4 with DarkNet combined with EfficientNet B3	▪ACC = 95% ▪F1-score = 92%	(Popescu et al., 2022b)
▪Combining NN YOLOv5s, DeepLabv3+ MobileNetv2	▪Detecting and segmentation of the logan fruit branch for logan harvesting using RGB-D camera ▪Pros: Reduced operating time and good ACC semantic segmentation ▪Cons: Limitations of object detection and segmentation in environmental interference conditions	▪Improved YOLOv5s for detection and DeepLabv3+ MobileNetv2 for semantic segmentation	▪ACC = 85.50% (fruit branch detection) ▪ACC = 94.52% (fruit branch semantic segmentation)	(Li, D. et al., 2022)
▪Faster R-CNN improved with the Feature Pyramid Networks (FPN)	▪Count the number of pecans in an orchard ▪Pros: Good mAP to identify pecans ▪Cons: Influence of lighting on fruit recognition and detection.	▪Faster R-CNN and FPN	▪mAP = 95.932%	(Hu et al., 2022)
▪Federated learning (FL) and improved Faster R-CNN.	▪Multiple pest detection ▪Pros: Can detect multiple pests in a short time. ▪Cons: ACC must be improved	▪Faster RCNN with ResNet 101 and with FL	▪mAP = 89.34% ▪ACC = 90.27% ▪Detection time = 0.05 s	(Deng et al., 2022)
▪Combining three improved DensNet 121	▪Pest detection from an augmented big dataset ▪Pros: Detecting pests on various agricultural crops ▪Cons: Performances must be improved	▪Improved three DensNet 121 and combined them into a decision fusion system	▪ACC = 75.28%	(Peng et al., 2023)

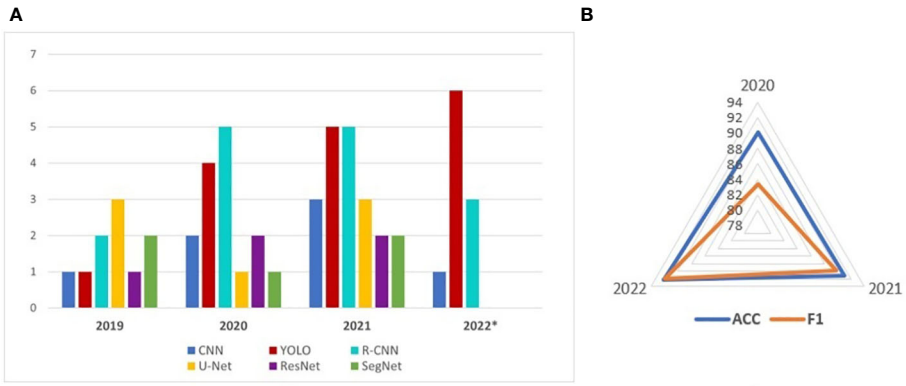


FIGURE 6
The most used NNs in orchards **(A)** and main performance indicators **(B)**.

Modica et al., 2020) based on UAV flight multispectral cameras, and morphological image processing techniques. Using U-Net and RGB images, the authors (Schiefer et al., 2020) perform tree species segmentation.

There are multiple ways to identify individual trees (canopy segmentation) in an orchard/forested area. These vary with the particularities of the specific trees and range in complexity from simple box partitioning like in (Horton et al., 2017) to handling irregular shapes and intermingled branches as in (Cheng et al., 2020) tested for cherry and apple trees orchards. Classically, the Hough transform for feature extraction has been often used but with relatively weak performance. Better performance was observed when using a Gaussian Mixture Model (Cheng et al., 2020). A similar approach is followed in (Dong et al., 2020), again for irregular crown shapes but this time applied to apple and pear trees. Crown segmentation is sometimes only an intermediary step for detecting the row lines and then, tree centers along each of these lines. (Zortea et al., 2018) implements such a mechanism for citrus orchards, a high-density case. Simply comparing the digital surface and terrain models (DSM and DTM) may also be used, as in (Gallardo-Salazar and Pompa-García, 2020) to geolocate trees and delineate their crowns.

The tree detection and classification procedure apply not only to curated environments (such as orchards) but also to natural growths which are more irregular in both tree size and placement like large boreal forest areas (Nevalainen et al., 2017). Another exception is (Tu et al., 2020) where high-resolution images were acquired from UAVs in a more complex context (areas with urban vegetation). The application is the semantic segmentation of trees of a specified species (Dipteryx alata - cumbaru class) using state-of-the-art networks. The NNs investigated were SegNet, U-Net, FC-DenseNet, and two DeepLabv3 + implementations (Xception and MobileNetV2) all with the same learning rates and optimizer for the learning phase. Moreover, a fully connected CRF (conditional random field) approach is proposed as a postprocessing step of the individual output NN decision. The results of using CRF were statistical performance improvement (ACC: 0.2% - 1.7%, F1-score: 0.2% - 1.9%, and IoU: 0.4% - 3%) and a decrease in computational efficiency (34.5 s for inference time). Regarding the performances of the studied networks, the best ACC, F1-score, and IoU (96.7%, 96.1%, and 92.5%) were obtained for FC-DenseNet and the lowest for DeepLabv3+Xception (88.9%, 87.1%, and 77.1%). Also, the best results for inference time were for FC-DenseNet (1.14 s) and the lowest for DeepLabv3+Xception (4.44 s).

It should be mentioned that some sources of error are systematic. For example, using a point cloud to estimate tree height naturally will provide less reliable height estimates if the tree shape narrows toward the top, which means that fewer points in the cloud are available for the 3D reconstruction (Gallardo-Salazar and Pompa-García, 2020). Even for simple photogrammetry applications, there are many features that may be considered. Beyond the standard segment length, segment intra-distance, and turn radius (the latter relevant only for fixed-wing UAVs) we may also consider height variation from segment to segment. E.g., in (Duarte et al., 2020) the segments follow the curvature of the terrain, leading to pictures taken along a surface that maintains a mostly constant height from the hilly ground beneath the camera. (Hulens et al., 2017) aims to detect through image processing the start and end points of an orchard row while traveling within it.

To obtain useful information for tree canopy extraction and segmentation, the images acquired by UAVs must be processed with various software (for example, Agisoft Photoscan) to generate geo-referenced ortho-images (Apolo-Apolo et al., 2020a; Adhikari et al., 2021). For example, in Figure 7 from a small, studied area the segmentation and elevation map is created using the photo capture points.

In most cases, the articles considered the detection and segmentation of some trees of a certain species, such as citrus (Csillik et al., 2018), palms (Culman et al., 2020), coconut (Iqbal et al., 2021), fig plant (Fuentes-Pacheco et al., 2019), etc., but the recommended solutions can also be applied to other types of orchards. In this case, the NNs system must be relearned with a new set of data (images) and the performances may be slightly different. Authors (García-Murillo et al., 2020) proposed the Cumulative Summation of Extended Maxima transform (SEMAX) methodology for the automatic individual detection of citrus and avocado trees.

A synthetic presentation of orchard and tree mapping and segmentation application is given in Table 6.

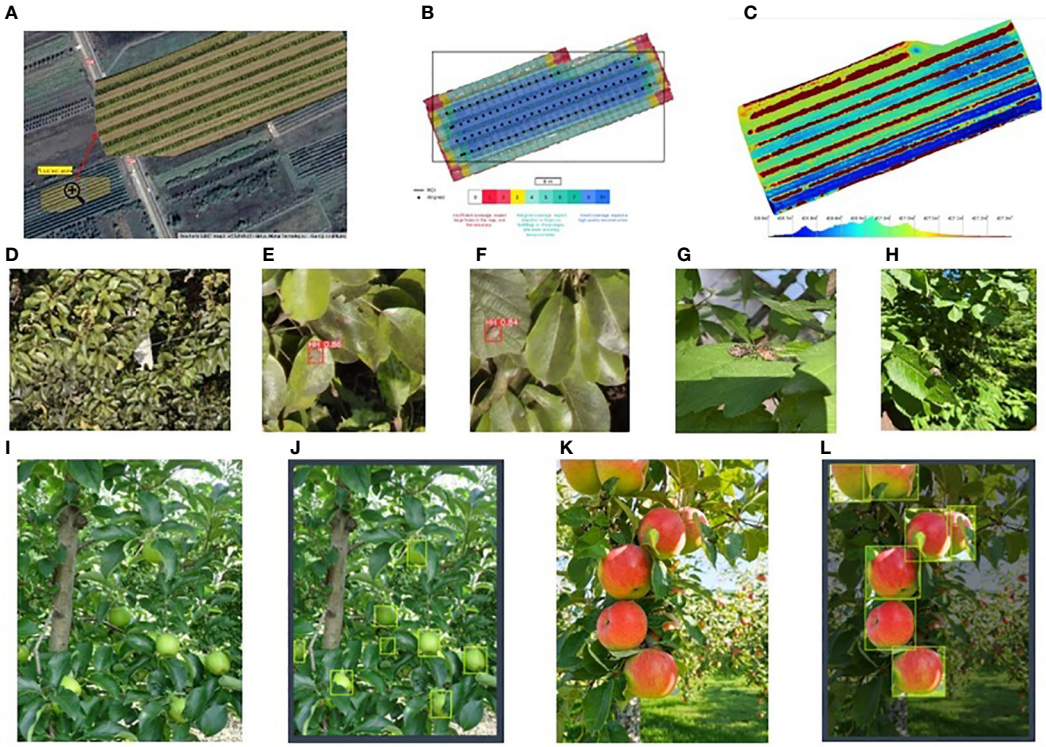


FIGURE 7
(A) Creating the elevation map: studied area, **(B)** Photo capture points, **(C)** Elevation map, **(D)** image from UAV containing HH in orchard, distant plan (4 m), **(E, F)** HH detected from D using image crops, **(G, H)** HH nymph at 0.6 m with manual acquisition, **(I)** Image with green apples in orchard, **(J)** Marked green apples, **(K)** Image with red apples in orchard, **(L)** Marked red apples.

5.2 Monitoring the evolution and condition of the orchard

Most of cases, the conditions and evolution of an orchard are evaluated from multispectral images, as can be seen in Table 6. But, since NNs are implemented for RGB images (three color channels), for multispectral images less of these networks were used. There are exceptions presented in Table 6. For example, in (Cunha et al., 2021) the vigor and health of peach trees are evaluated using vegetable indexes like NDVI (normalized difference vegetation index), GNDVI (green NDVI), NDRE (normalized difference red edge index), and REGNDVI (red-edge GNDVI) calculated from

TABLE 6 Orchard and tree segmentation. Monitoring the evolution and condition of the orchard.

Purpose (orchard task)	Resources	Performance	References
Orchard and tree segmentation			
▪Detection of Citrus Trees based on a UAV flight and image processing in two steps: detection and classification	▪UAV; multispectral camera; Simple CNN for detection; Simple Linear Iterative Clustering algorithm (SLIC) for classification.	▪ACC=96.24%,	(Csillik et al., 2018)
▪Individual palms detection from high-resolution remote sensing images	▪UAV; RGB camera; RetinaNet	▪mAP=86.1%	(Culman et al., 2020)
▪ Fig plant segmentation	▪UAV; RGB camera; encoder-decoder DCNN, inspired by SegNet architecture	▪ACC=93.85%	(Fuentes-Pacheco et al., 2019)
▪Tree detection and position	▪UAV; hyperspectral camera; different CNNs	▪F1 = 95.9%,	(Miyoshi et al., 2020)
▪Branch detection of apple trees	▪UAV; RGB camera; Pseudo-Color Images and Depth, R-CNN	▪REC=92%, ▪ACC=86%	(Zhang et al., 2018)
▪Detection and segmentation of trunk/branch, apples, and leaves	▪Terrestrial platform; RGB-D camera; ResNet-18	▪ACC= 94.5%-94.8%	(Zhang, X. et al., 2019)

(Continued)

TABLE 6 Continued

Purpose (orchard task)	Resources	Performance	References
▪Identify the tree trunks and branches for a harvesting system	▪RGB camera; Deeplab v3+ with backbone: ResNet-18, VGG-16, and VGG-19	▪Per-class accuracy (PcA) =97%	(Zhang X. et al., 2021)
▪Semantic segmentation of citrus trees in a dense orchard	▪UAV; multispectral camera; FCN, U-Net, SegNet, DDCN, DeepLabV3 +	▪ACC= 94.88%-95.96%	(Osco et al., 2021)
▪Detection and classification of individual tree	▪UAV; RGB camera; AlexNet, SqueezeNet, VGG 16; ResNet 50, DenseNet 121	▪ACC = 97.6% -99.5%	(Nguyen et al., 2021)
▪Dection and semantic segmentation of coconut trees	▪UAV; RGB camera; Mask R-CNN with ResNet101 as backbone	▪mAP=91%	(Iqbal et al., 2021)
▪Segmentation of planting rows of orange trees	▪UAV; RGB camera; Pipeline of two encoder-decoder networks (DetED – for detection and CorrED – for correction	▪ACC = 94% - 99.5%,	(Rosa et al., 2020)
Monitoring the evolution and condition of the orchard			
▪Evaluating the phenotypic characteristics of orange trees with influences on plant growth	▪UAV; multispectral camera; YOLO v3	▪PRE=99.9%	(Ampatzidis et al., 2020)
▪Evaluating the vigor and health of trees in a peach orchard using multispectral images	▪UAV; multispectral camera; Faster R-CNN	▪NA	(Cunha et al., 2021)
▪Recognition of spraying areas in the orchard.	▪UAV; RGB camera; improved Faster R-CNN	▪ACC=87.77% - 88.57%	(Khan et al., 2021)
▪Determination of the NDVI in a pomegranate orchard	▪UAV; Deep Stochastic Configuration Networks (DeepSCNs), regression model	▪R2 = 99.5%	(Niu et al., 2020)
▪Nitrogen concentration in an apple orchard	▪UAV; hyperspectral camera; backpropagation neural network (BPNN)	▪R2 = 77%	(Li, W. et al., 2022)
▪Nitrogen, Phosphorus, and Potassium foliar content retrieval in olive trees	▪UAV; multispectral camera; ANN	R2 = 63% - 95%	(Noguera et al., 2021)
▪Monitoring citrus orchards	▪UAV; RGB camera; FCRN-MTL	▪PRE=95%	(La Rosa et al., 2020)
▪Multispecies fruit flower (apple, peach, and pear) detection by semantic segmentation	▪Datasets publicly available; RGB camera; residual convolutional neural	▪F1 = 74.2%- 86%	(Dias et al., 2018)
▪ Estimating olive tree's biovolume	▪UAV; multispectral camera; Mask R-CNN based on ResNet50	▪F1 = 95%-98%	(Safonova et al., 2021)
▪Evaluating the temperature in an apple orchard for frost protection	▪UAV; RGB camera; thermal camera; YOLOv4	▪mAP= 66.08%-71.57%	(Yuan and Choi, 2021)

multispectral images. Other research is focused on the detection of spraying areas (Khan et al., 2021) and concentrations of various chemical substances like Nitrogen, Phosphorus, and Potassium (Noguera et al., 2021) in the leaves. The summary of the orchard evolution monitoring is in Table 6.

5.3 Detection of pests and tree diseases in orchards

Pest detection using UAV is an important application of orchard monitoring because pests cause significant loss of crop production (Castrignanò et al., 2021). A recent review of the impact of climate change (IPPC Secretariat et al., 2021) on plant pests showed that pests have expanded to new areas. FAO estimates that every year the losses caused by pests are up to 40% of global crop production. Therefore, pests and disease detection and their spread prediction in real-time are needed for efficient and non-polluting interventions. Detecting the pests and diseases of trees in orchards as early as possible can limit their spread. Manual observation is timely loss and inefficient (Roosjen et al., 2020). Using UAVs and artificial intelligence in pest detection and evaluation, important progress can be observed (Peng et al., 2023). The low-altitude flight of UAVs is more effective than the ground diagnosis which is time-consuming and laborious on large area monitoring (Lan et al., 2020).

In organic orchards, it is particularly important to detect and monitor insects, especially harmful ones. For this, there are several ways such as direct visual inspection of farmers, land platforms, or drones. The last option is the most efficient because it can cover a relatively important area in a short time. In (Sorbelli et al., 2022), a method of sweeping individual trees from an orchard for the detection and evaluation of harmful insects (Halyomorpha Halys (HH)) is described. Four NNs were compared (Ichim et al., 2022) to highlight the best-performing network in HH detection. For this experiment, the result was DenseNet201. Note that HH or other harmful insects are at least an order of magnitude smaller than fruits like apples or pears, hence the problem of accurately detecting and counting them is even more challenging. The partial occlusion

is challenging and the estimation of the abundance of these insects is a difficult problem. In Figure 7 some examples of HH at different stages of evolution and other insects in images taken on different conditions confirm the difficulty of real detection of insects in trees from UAV. As can be seen, the image from UAV at a safe distance (in automatic surveillance) contains insects hard to be distinguished and the recommended action is to split the images in crops and then detect the insects with NN. If the insects are in the first plan or in the public dataset the task detection is easier (Xing et al., 2019).

A synthetic presentation of tree disease and pest detection is given in Table 7.

5.4 Prediction and evaluation of orchard production

As specified by (Wang C. et al., 2022; Koirala et al., 2019b) the evaluation of fruit production is an important activity both from the social and economic points of view. The authors used a combined YOLO5 and FlowNet2 scheme to improve apple detection in an orchard for accurate yield estimation. They claim a good performance and a framerate of 20 frames/second even for partially occluded targets and under varying illumination conditions. This is in contrast with typical applications where the analysis is carried out offline.

TABLE 7 Detection of pests and tree diseases. Prediction and evaluation of orchard production .

Purpose (orchard task)	Resources and discussions	Performance	References
Detection of pests and tree diseases			
Infected or diseased trees detection	▪UAV; Faster R-CNN and Mask R-CNN approaches and fusing their outputs	▪SEN=81.67%	(Barmpoutis et al., 2019)
Detection of the citrus bacterial canker in disease development stages on Sugar Belle leaves and immature fruit	▪UAV; hyperspectral camera; the neural network Radial Basis Function (RBF) and the K-nearest neighbor (KNN)	▪ACC= 94%-100%	(Abdulridha et al., 2019)
Identification of fruit tree pests (Tessaratoma papillosa)	▪UAV; RGB camera; Tiny-YOLOv3	▪mAP= 38.12%-95.33%	(Chen, C.J. et al., 2021)
Detection of the degree of HLB (huanglongbing) infection on large-scale orchard citrus trees	▪UAV; multispectral camera; stacked autoencoder (SAE) neural network	▪ACC= 99.72%	(Deng et al., 2020)
	▪UAV; multispectral camera; autoencoder	▪ACC=97.28%,	(Lan et al., 2020)
Detection of diseases in vineyards	▪UAV; multispectral camera; LeNet-5, SegNet – single or combination	▪ACC=78.72%-95.02	(Kerkech et al., 2020)
	▪UAV; RGB camera; LeNet-5	▪ACC=95.8%	(Kerkech et al., 2018)
	▪UAV; RGB camera; CaffeNet	▪NA	(Bouroubi et al., 2018)
	▪UAV; multispectral camera; VddNet	▪ACC=93.72	(Kerkech et al., 2020)
Detection of the presence and behavior of the nematode pest in coffee crops	▪UAV; RGB camera; U-Net and PSPNet	▪F1 = 69%	(Oliveira et al., 2019)
Detection of black rot on grape leaves	▪UAV; RGB camera; YOLOv3 with SPP module	▪PRE=94.05%, SEN=93.26%	(Zhu et al., 2021)
Sick tree detection	▪UAV; RGB camera; different CNNs: Alexnet, Squeezenet, VGG 16; Resnet 50, Densenet 121	▪ACC=97.6% -99.5%	(Nguyen et al., 2021)
Bug detection (Halyomorpha Halys) in an orchard	▪UAV; RGB camera; processing (NN)	▪NA	(Sorbelli et al., 2022), (Ichim et al., 2022)
Insect detection, invasive species (Anolis carolinensis)	▪UAV, RGB camera; SSD-based model of DCNN	▪PRE=70%	(Aota et al., 2021)
Invasion degree of "Solanum rostratum Dunal" detection	▪UAV; RGB camera; DeepSolanum-Net based on U-Net	▪F1 = 89.85%	(Wang et al., 2021)
Prediction and evaluation of orchard production			
▪Method for semantic segmentation and instance segmentation of bayberry fruit.	▪Terrestrial platform; RGB camera; Multi-module convolutional neural network	▪AP = 75.5% -91.3%	(Lei et al., 2022)
▪Accurate monitoring of fruit quantity in apple orchards	▪UAV inside orchard; RGB camera; YOLO v5s	▪AP = 90.39%	(Wang S. et al., 2022)
▪Yield estimates in apple orchards. Detecting apples on individual trees.	▪UAV; RGB camera; R-CNN	▪R^2 = 80% - 86%	(Apolo-Apolo et al., 2020a)

(Continued)

TABLE 7 Continued

Purpose (orchard task)	Resources and discussions	Performance	References
▪Detection, counting, and estimation of the size of citrus fruits on individual trees	▪UAV; RGB camera; Faster R-CNN	▪F1 = 89%	(Apolo-Apolo et al., 2020b)
▪Detection and location of longan fruits	▪UAV; RGB camera; MobileNet backbone used to improve YOLOv4	▪mAP = 54.22 -89.73%	(Li D. et al., 2021)
▪Holly fruits detection and counting	▪UAV; RGB camera; YOLOX	▪DR >99%	(Zhang Y. et al., 2022)
▪Canopy extraction. Detect mango and predict the number on the tree	▪Terrestrial platform; RGB camera; Mango YOLO, Xception, Random Forest	▪R^2 = 98%	(Koirala et al., 2021)
▪Detect apple fruit in the orchard	▪Manual images; RGB camera; comparing RetinaNet, Libra-RCNN, Cascade-RCNN, Faster-RCNN, FSAF, HRNet, and ATSS	▪Maximum AP = 94.6%	(Biffi et al., 2021)
▪Longan harvesting UAVs. Branch detection and fruit branch semantic segmentation.	▪UAV; RGB-D camera; YOLOv5s – for detection, and improved DeepLabv3+ (MobileNet v2) for semantic segmentation	▪ACC = 85.50% – 94.52%	(Li D. et al., 2022)
▪Grape detection, instance segmentation	▪RGB camera; Mask R-CNN with ResNet 101 as the backbone	▪F1 = 91%	(Santos et al., 2020)
▪Pear (fruit) detection	▪RGB camera; YOLO-P	F1 = 96.1%	(Sun et al., 2023)

The standard, encountered in virtually all aerial systems older than 5-10 years, is to gather the raw data and, at most, do some preliminary preprocessing before sending it to a ground station for further analysis. This has the obvious benefit of minimizing the hardware complexity and energy requirements for the drone but makes impractical "live" implementations where the mission must be updated on-the-fly from the gathered information. Recent applications, due to significant hardware resources, have started to handle increasing parts of the workflow onto the drone. While the effort is by no means trivial, dedicated software such as Jetson Nano, Google Coral, and the like permit image processing directly onto the drone. This means that decisions may be taken in a fully local manner (without interaction with the ground). Even a supervisor (human or software agent) still must be in the loop (as is the case for most commercial applications), there still is the benefit of reduced bandwidth allocation (since more steps of the image processing are done on the platform, it means that only relevant information is exchanged with the ground).

On the other hand, for position correction, collision avoidance, and even target counting (Wang S. et al., 2022), optical flow methods which compare consecutive frames to detect changes are used. This has the advantage of improving performance but comes usually with a reduction in resolution (since video frames have, unavoidably, less resolution than static images).

The great majority of drone trajectories are out of a plane (images/videos are taken top-down while the drone is flying over the treetops). Still, there are some results such as in (Wang S. et al., 2022) where the drone travels mid-row, through the orchard's rows.

Using artificial intelligence methods to process the images acquired by autonomous terrestrial or aerial platforms, the conditions for picking fruits that have reached maturity in the optimal period can be improved. This approach leads to increased economic efficiency for orchards (Lei et al., 2022). Fruit estimation is challenging and the number of fruits on a tree cannot be measured exactly due to occlusions (Zhang X. et al., 2019).

Because of the similarity between the fruit and the leaf, the detection of green citrus fruits or green apples (Figure 7) is quite difficult. The authors (Zheng et al., 2021) proposed a modification of the YOLO neural network modules (starting from YOLO v4), called YOLO BP which detects the respective fruits with higher precision than YOLO v4. If the fruits are a color different from the leaves or are not obturated the detection task is easier (Figure 7). NIR is used especially for highlighting the leaves and the production of almonds in a tree. For example, in (Tang et al., 2023) aerial multi-spectral images (near-infrared, red edge, red, and green) are processed by a CNN to estimate the almond production in an orchard with a coefficient of determination, R2 = 96%. It is specified that the sun-shadow effect can decrease system performance.

A synthetic presentation of fruit production evaluation is given in Table 7.

6 Discussion

The use of UAVs and NNs for image processing in orchard monitoring is a relatively new method open to both research and end-user implementation. This was possible due to the development of new technologies in recent years and the decrease in the prices of the necessary equipment. Unfortunately, most of the current UAV applications are relatively simple from the viewpoint of trajectory generation (straight lines or successive set points to be reached). Still, continuous advances in hardware capabilities and the expected expansion of mission complexity mean that more complex scenarios will be defined and tackled. Continuous reduction in size, cost, and dimensions means that various sensor mechanisms (Lidar for example) may now be mounted onboard. Not least, improvements in embedded image processing (software and hardware modules such as Jetson Nano or Google Coral) mean that image-based positioning is now increasingly used. Henceforth, we expect that algorithms initially tailored for ground vehicles will

be adapted in the next few years to aerial systems. For example, a great many algorithms exist for in-lane orchard navigation for ground autonomous systems (small-sized tractors, (Emmi et al., 2021)) and it should be possible to adapt them with minimal modifications. Although it is preferable to other methods such as terrestrial platforms or human operators, automatic UAV flight and establishing the trajectory inside the orchard for the acquisition of images is sometimes a real challenge due to several aspects such as: a) keeping a safe distance from tree branches, b) obtaining a continuous 3D surface (similar to orthomosaic) from which to cut out the images to be analyzed, c) detecting, segmenting and classifying small (insects, some fruits, diseases) and/or partially covered objects, d) large differences in brightness, e) background difficulty, etc. All this, including the characteristics of public databases (if they are used) leads to different performances for the same type of application.

It can be noted that, in general, the performances obtained depend both on the networks used and on the quality of the acquired data set. Many times, the division of high-resolution acquired images into sub-images (patches) and their analysis by the proposed NNs give better results than the processing of large images through the resizing required by the networks. This solution can be useful when trying to detect small objects in trees (such as insects). The performance of networks or systems made of multiple networks leans either on meeting the needs of precision or on meeting the needs fast processing, or on the compromise between these two. Anyway, for a large-scale application, on various farms, a solution that saves resources or a remote processing solution *via* the Internet is preferable. Another recommendation is to use, in situations where NIR images provide relevant information, to combine NNs for RGB with NNs for NIR in a global decision system.

There are several review articles with the topic of some common parts with this article, but none that include the triplet orchard, UAV, and NNs. Their descriptions and the novelty introduced in our paper are presented in Table 8.

7 Conclusions

This review covers a critical gap in modern orchard monitoring considering the essential contribution of both UAV and NNs as

TABLE 8 Recent review/survey papers on similar topics.

Paper	Description	Period	Ref.	Our differences (improvement or novelty)
(Kamilaris and Prenafeta-Boldú, 2018)	▪Using CNNs in agriculture. ▪Comparing NN with other techniques in agricultural applications, high precision, and accuracy are obtained.	1995-2018	62	▪Focused on orchard monitoring from different points of view (applications). Focused on new trends in NN usage. Graphs on the evolution of UAV and NN use in the last period. Description of using UAVs for image acquisition. More references. New period.
(Koirala et al., 2019b)	▪Using DL for fruit detection and yield estimation. ▪Comparing the statistical performances of CNN methods.	1991-2019	83	▪Focused on orchard monitoring from different points of view. Focused on new trends in NN usage. Graphs on the evolution of UAV and NN use in the last period. Description of using UAVs for image acquisition. More references. New period.
(Barbedo, 2019)	▪Using UAVs and image acquisition and processing to monitor and assess the plant stresses.	2003-2018	169	▪Focused on orchard monitoring from different points of view (applications). Focused on new trends in NN usage. Graphs on the evolution of UAV and NN use in the last period. More references. New period.
(Ma et al., 2019)	▪Using deep NNs in general remote sensing applications.	1991-2018	148	▪Focused on orchard monitoring from different points of view. Focused on new trends in NN usage. Graphs on the evolution of UAV and NN use in the last period. Description of using UAVs for image acquisition. More references. New period.
(Iost Filho et al., 2020)	▪Using multi-copters in pest management to identify harmful areas and to accurately spray pesticides. Sensing and actuation UAVs are investigated in agricultural systems	1986-2019	320	▪Focused on orchard monitoring from different points of view (applications). Focused on detailed descriptions of NN used and new trends. Graphs on the evolution of UAV and NN use in the last period. New period.
(Lu and Young, 2020)	▪Analyzing and establishing the main characteristics of 34 public image DSs for computer vision tasks in precision agriculture: 15 on weed control, 10 on fruit detection, and 9 for other applications.	2009-2020	98	▪Focused on orchard monitoring from different points of view (applications). Focused on new trends in NN usage. Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use in the last period. More references. New period.
(Naranjo-Torres et al., 2020)	▪Using CNN for fruit recognition. Presentation of fundamentals, tools, and examples of CNNs for fruit sorting and quality control.	1998-2020	104	▪Focused on orchard monitoring from different points of view. Focused on new trends in NN usage. Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use in the last period. More references. New period.
(Zhang et al., 2020)	▪Using DL for dense scenes analysis in agriculture. Analyzing the challenges in dense agricultural scenes. Presentation of architectures of DL algorithms and CNNs used in dense agricultural scenes	1988-2019	122	▪Focused on orchard monitoring from different points of view (applications). Focused on new trends in NN usage. Graphs on the evolution of UAV and NN use in the last period. Description of using UAVs for image acquisition. More references. New period.

(Continued)

TABLE 8 Continued

Paper	Description	Period	Ref.	Our differences (improvement or novelty)
(Dhaka et al., 2021)	▪Using DCNN for prediction of plant diseases from leaf images.	1989-2021	124	▪Focused on orchard monitoring from different points of view (applications). Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use. More references.
(Li L. et al., 2021)	▪Using DL for plant leaf disease detection and classification	2006-2020	113	▪Focused on orchard monitoring from different points of view (applications). Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use. More references.
(Liu and Wang, 2021)	▪Using DL for plant diseases and pest detection, considering three functions of NN: classification, detection, and segmentation.	2006-2021	108	▪Focused on orchard monitoring from different points of view (applications). Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use. More references.
(Olson and Anderson, 2021)	▪Presentation of UAVs, image sensors, image acquisition, image processing, and their applications in agriculture	1973-2021	154	▪Focused on orchard monitoring from different points of view (applications). Focused on new trends in NN usage. Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use in the last period. More references.
(Zhang C. et al., 2021)	▪Presentation of orchard management with small UAVs	1978-2019	147	▪Focused on new trends in NN usage for image processing for orchard monitoring. Graphs on the evolution of NN use in the last period. More references. New period.
(de Castro et al., 2021)	▪Using UAVs for vegetation monitoring considering diverse agricultural and forestry scenarios such as vegetation indices, technological goals, and applications.	2004-2021	48	▪Focused on orchard monitoring from different points of view (applications). Focused on detailed descriptions of NN used and new trends. Graphs on the evolution of UAV and NN use. More references.
(Wang C. et al., 2022)	▪Detecting the phases of fruit evolution from flower, growth, ripening, picking, and classification, based on the analysis of images captured by terrestrial or aerial robots. NNs with one or two stages, built for object detection were considered.	1986-2022	201	▪Focused on orchard monitoring from different points of view (applications). More NNs. Focused on new trends in NN usage. Description of using UAVs for image acquisition. Graphs on the evolution of UAV and NN use in the last period. More applications

exponents of new technologies. As can be seen both from the analysis of research articles and review articles, only in recent years have these hardware/software resources been involved and analyzed in research in the field. Both the advantages offered by the two components (UAV and NN) of the analyzed orchard monitoring systems were highlighted as well as the challenges due to the difficulties encountered in real orchards, related to the UAV flight inside the orchards among the trees and the detection of small objects such as fruits or insects inside the crowns. The newest technologies used in modern orchards were analyzed in support of increasing production, increasing fruit quality, and eliminating pests and diseases through environmentally friendly means. Special emphasis was placed on the new trends in the development of the main analyzed vectors, namely NNs, and UAVs. The final discussion regarding the comparison with other review articles highlights the article's contributions regarding improvements and new approaches. We hope the paper will help the researchers and producers of modern systems for orchard monitoring in the context of Agriculture 4.0. As previously stated in the paper, a limitation of the approach is the relatively small number of existing research articles in the complex topic of orchard monitoring-UAV-neural networks (it is a new field, in full expansion). As a future direction, we will follow the ever-growing evolution in this field, based on the fusion of information from terrestrial and aerial robots, for the most efficient monitoring of orchards using artificial intelligence techniques.

Author contributions

Conception: DP, LI. Project administration: DP, LI, and FS. Writing – original draft: DP, LI, and FS. All authors contributed to the article and approved the submitted version.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article. This work was supported by HALY.ID project. HALY.ID is part of ERA-NET Co-fund ICT-AGRI-FOOD, with funding provided by national sources [Funding agency UEFISCDI, project number 202/2020, within PNCDI III] and co-funding by the European Union's Horizon 2020 research and innovation program, Grant Agreement number 862665 ERA-NET ICT-AGRI-FOOD (HALY-ID 862671).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abdulridha, J., Batuman, O., and Ampatzidis, Y. (2019). UAV-based remote sensing technique to detect citrus canker disease utilizing hyperspectral imaging and machine learning. *Remote Sens.* 11 (11), 1–22, 1373. doi: 10.3390/rs11111373

Adamo, F., Attivissimo, F., Di Nisio, A., Ragolia, M. A., and Scarpetta, M. (2021). A new processing method to segment olive trees and detect xylella fastidiosa in UAVs multispectral images. Proc. IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Glasgow, United Kingdom, 1–6. doi: 10.1109/I2MTC50364.2021.9459835

Adhikari, A., Kumar, M., Agrawal, S., and Raghavendra, S. (2021). An integrated object and machine learning approach for tree canopy extraction from UAV datasets. *J. Indian Soc. Remote Sens.* 49, 471–478. doi: 10.1007/s12524-020-01240-2

Akca, S., and Polat, N. (2022). Semantic segmentation and quantification of trees in an orchard using UAV orthophoto. *Earth Sci. Inform* 15 (1), 2265–2274. doi: 10.1007/s12145-022-00871-y

Alvarez-Vanhard, E., Corpetti, T., and Houet, T. (2021). UAV & satellite synergies for optical remote sensing applications: A literature review. *Sci. Remote Sens.* 3, 100019. doi: 10.1016/j.srs.2021.100019

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big. Data* 8, 53. doi: 10.1186/s40537-021-00444-8

Ampatzidis, Y., Partel, V., and Costa, L. (2020). Agroview: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence. *Comput. Electron. Agric.* 174, 1–12, 10545. doi: 10.1016/j.compag.2020.105457

Ampatzidis, Y., Partel, V., Meyering, B., and Albrecht, U. (2019). Citrus rootstock evaluation utilizing UAV-based remote sensing and artificial intelligence. *Comput. Electron. Agric.* 164 (C), 1–10, 104900. doi: 10.1016/j.compag.2019.104900

Aota, T., Ashizawa, K., Mori, H., Toda, M., and Chiba, S. (2021). Detection of Anolis carolinensis using drone images and a deep neural network: an effective tool for controlling invasive species. *Biol. Invasions.* 23, 1321–1327. doi: 10.1007/s10530-020-02434-y

Apolo-Apolo, O. E., Martínez-Guanter, J., Egea, G., Raja, P., and Pérez-Ruiz, M. (2020b). Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. *Eur. J. Agron.* 115, 1–11, 126030. doi: 10.1016/j.eja.2020.126030

Apolo-Apolo, O. E., Pérez-Ruiz, M., Martínez-Guanter, J., and Valente, J. A. (2020a). Cloud-based environment for generating yield estimation maps from apple orchards using UAV imagery and a deep learning technique. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.01086

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: a brief survey. *IEEE Signal Process. Magazine.* 34 (6), 26–38. doi: 10.1109/MSP.2017.2743240

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: a deep convolutional encoder decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (12), 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Barbedo, J. G. A. (2019). A review on the use of Unmanned Aerial Vehicles and imaging sensors for monitoring and assessing plant stresses. *Drones* 3 (2), 1–27, 40. doi: 10.3390/drones3020040

Barbosa, B. D. S., Araújo e Silva Ferraz, G., Mendes dos Santos, L., Santana, L. S., Bedin Marin, D., Rossi, G., et al. (2021). Application of RGB images obtained by UAV in coffee farming. *Remote Sens.* 13, 1–19, 2397. doi: 10.3390/rs13122397

Barmpoutis, P., Kamperidou, V., and Stathaki, T. (2019). Estimation of extent of trees and biomass infestation of the suburban forest of Thessaloniki (Seich Sou) using UAV imagery and combining R-CNNs and multichannel texture analysis. In *Proc. Twelfth. Int. Conf. Mach. Vision (ICMV)*, Amsterdam, 114333C. doi: 10.1117/12.2556378

Barrado, C., Boyero, M., Brucculeri, L., Ferrara, G., Hately, A., Hullah, P., et al. (2020). U-space concept of operations: a key enabler for opening airspace to emerging low-altitude operations. *Aerospace* 7 (3), 1–18, 24. doi: 10.3390/aerospace7030024

Bhatnagar, S., Gill, L., and Ghosh, B. (2020). Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities. *Remote Sens.* 12, 1–26, 2602. doi: 10.3390/rs12162602

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., et al. (2021). CNN variants for computer vision: history, architecture, application, challenges and future scope. *Electronics* 10 (20), 1–28, 2470. doi: 10.3390/electronics10202470

Bhoi, S. K., Jena, K. K., Panda, S. K., Long, H. V., Kumar, R., Subbulakshmi, P., et al. (2021). An Internet of Things assisted Unmanned Aerial Vehicle based artificial intelligence model for rice pest detection. *Microprocessors. Microsyst.* 80, 1–11, 103607. doi: 10.1016/j.micpro.2020.103607

Biffi, L. J., Mitishita, E., Liesenberg, V., Santos, A., Gonçalves, D. N., Estrabis, N. V., et al. (2021). ATSS deep learning-based approach to detect apple fruits. *Remote Sens.* . 13 (1), 1–22, 54. doi: 10.3390/rs13010054

Bouroubi, Y., Bugnet, P., Nguyen-Xuan, T., Gosselin, C., Bélec, C., Longchamps, L., et al. (2018). Pest detection on UAV imagery using a deep convolutional neural network. Proc. 14th International Conference on Precision Agriculture, Montreal, Quebec, Canada, 1–11. doi: 10.1145/3232651.3232661

Breslla, K., Bortolotti, G., Boini, A., Perulli, G., Morandi, B., Grappadelli, L. C., et al. (2020). "Sensor-fusion and deep neural networks for autonomous UAV navigation within orchards," in *Proc. IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. 230–235. doi: 10.1109/MetroAgriFor50201.2020.9277568

Caesar, H., Uijlings, J., and Ferrari, V. (2018). "COCO-Stuff: thing and stuff classes in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*. 1209–1218. doi: 10.48550/arXiv.1612.03716

Castrignanò, A., Belmonte, A., Antelmi, I., Quarto, R., Quarto, F., Shaddad, S., et al. (2021). Semi-automatic method for early detection of Xylella fastidiosa in olive trees using UAV multispectral imagery and geostatistical-discriminant analysis. *Remote Sensing* 13 (1), 1–23, 14. doi: 10.3390/rs13010014

Champ, J., Mora-Fallas, A., Goëau, H., Mata-Montero, E., Bonnet, P., and Joly, A. (2020). Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots. *Appl. Plant Sci.* 8 (7), 1–10, e11373. doi: 10.1002/aps3.11373

Chen, C.-J., Huang, Y.-Y., Li, Y.-S., Chen, Y.-C., Chang, C.-Y., and Huang, Y.-M. (2021). Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access* 9, 21986–21997. doi: 10.1109/ACCESS.2021.3056082

Chen, T., Zhang, R., Zhu, L., Zhang, S., and Li, X. (2021). A method of fast segmentation for banana stalk exploited lightweight multi-feature fusion deep neural network. *Machines* 9, 66. doi: 10.3390/machines9030066

Chen, Y., Hou, C., Tang, Y., Zhuang, J., Lin, J., He, Y., et al. (2019). Citrus tree segmentation from UAV images based on monocular machine vision in a natural orchard environment. *Sensors* 19, 5558. doi: 10.3390/s19245558

Cheng, Z., Qi, L., Cheng, Y., Wu, Y., and Zhang, H. (2020). Interlacing orchard canopy separation and assessment using UAV images. *Remote Sens.* 12, 767. doi: 10.3390/rs12050767

Chew, R., Rineer, J., Beach, R., O'Neil, M., Ujeneza, N., Lapidus, D., et al. (2020). Deep neural networks and transfer learning for food crop identification in UAV images. *Drones* 4, 7. doi: 10.3390/drones4010007

Crommelinck, S., Koeva, M., Yang, M. Y., and Vosselman, G. (2019). Application of deep learning for delineation of visible cadastral boundaries from remote sensing imagery. *Remote Sens.* 11, 2505. doi: 10.3390/rs11212505

Csillik, O., Cherbini, J., Johnson, R., Lyons, A., and Kelly, M. (2018). Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones* 2, 39. doi: 10.3390/drones2040039

Culman, M., Delalieux, S., and Van Tricht, K. (2020). Individual palm tree detection using deep learning on RGB imagery to support tree inventory. *Remote Sensing* 12 (21), 1-30, 3476. doi: 10.3390/rs12213476

Cunha, J., Gaspar, P. D., Assunção, E., and Mesquita, R. (2021). Prediction of the vigor and health of peach tree orchard. *Lecture. Notes Comput. Sci.* 12951, 541–551. doi: 10.1007/978-3-030-86970-0_38

de Castro, A. I., Shi, Y., Maja, J. M., and Peña, J. M. (2021). UAVs for vegetation monitoring: overview and recent scientific contributions. *Remote Sensing* 13 (11), 1–13, 1–13, 2139. doi: 10.3390/rs13112139

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). "ImageNet: a large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 248–255. doi: 10.1109/CVPR.2009.5206848

Deng, F., Mao, W., Zeng, Z., Zeng, H., and Wei, B. (2022). Multiple diseases and pests detection based on federated learning and improved faster R-CNN. *IEEE Trans. Instrumentation. Measurement.* 71, 3523811. doi: 10.1109/TIM.2022.3201937

Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., et al. (2021). Automatic diagnosis of rice diseases using deep learning. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.701038

Deng, X., Zhu, Z., Yang, J., Zheng, Z., Huang, Z., Yin, X., et al. (2020). Detection of citrus huanglongbing based on multi-input neural network model of UAV hyperspectral remote sensing. *Remote Sens.* 12, 2678. doi: 10.3390/rs12172678

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Kavita,, Ijaz, M. F., et al. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21 (14), 4749. doi: 10.3390/s21144749

Dias, P. A., Tabb, A., and Medeiros, H. (2018). Multispecies fruit flower detection using a refined semantic segmentation network. *IEEE Robotics. Automation. Lett.* 3 (4), 3003–3010. doi: 10.1109/LRA.2018.2849498

Diwan, T., Anirudh, G., and Tembhurne, J. V. (2022). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* 82 (6), 9243–9275. doi: 10.1007/s11042-022-13644-y

Dong, X., Zhang, Z., Yu, R., Tian, Q., and Zhu, X. (2020). Extraction of information about individual trees from high-spatial-resolution UAV-acquired images of an orchard. *Remote Sens.* 12 (1), 133. doi: 10.3390/rs12010133

Duarte, A., Acevedo-Muñoz, L., Gonçalves, C. I., Mota, L., Sarmento, A., Silva, M., et al. (2020). Detection of Longhorned Borer attack and assessment in eucalyptus plantations using UAV imagery. *Remote Sens.* 12 (19), 3153. doi: 10.3390/rs12193153

Dyson, J., Mancini, A., Frontoni, E., and Zingaretti, P. (2019). Deep learning for soil and crop segmentation from remotely sensed data. *Remote Sens.* 11 (16), 1859. doi: 10.3390/rs11161859

Emmi, L., Le Flécher, E., Cadenat, V., and Devy, M. (2021). A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture. *Precis. Agric.* 22, 524–549. doi: 10.1007/s11119-020-09773-9

Encinas-Lara, M. S., Méndez-Barroso, L. A., and Yépez, E. A. (2020). Image dataset acquired from an unmanned aerial vehicle over an experimental site within El Soldado estuary in Guaymas, Sonora, México. *Data Brief* 30, 105425. doi: 10.1016/j.dib.2020.105425

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). The PASCAL visual object classes challenge: a retrospective. *Int. J. Comput. Vision* 111 (1), 98–136. doi: 10.1007/s11263-014-0733-5

Fang, W., Yue, L., and Dandan, C. (2020). "Classification system study of soybean leaf disease based on deep learning," in *Proc. International Conference on Internet of Things and Intelligent Applications (ITIA)*. 1–5. doi: 10.1109/ITIA50152.2020.9312252

Fawakherji, M., Youssef, A., Bloisi, D., Pretto, A., and Nardi, D. (2019). "Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation," in *Proc. Third IEEE International Conference on Robotic Computing (IRC)*. 146–152. doi: 10.1109/IRC.2019.00029

Fernandez-Gallego, J. A., Kefauver, S. C., Gutiérrez, N. A., Nieto-Taladriz, M. T., and Araus, J. L. (2018). Wheat ear counting in-field conditions: high throughput and low-cost approach using RGB images. *Plant Methods* 14, 22. doi: 10.1186/s13007-018-0289-4

Fuentes-Pacheco, J., Torres-Olivares, J., Roman-Rangel, E., Cervantes, S., Juarez-Lopez, P., Hermosillo-Valadez, J., et al. (2019). Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sens.* 11 (10), 1157. doi: 10.3390/rs11101157

Furchì, A., Lippi, M., Carpio, R. F., and Gasparri, A. (2022). "Route optimization in precision agriculture settings: a multi-steiner TSP formulation," in *IEEE Transactions on Automation Science and Engineering*. 1–18. doi: 10.1109/TASE.2022.3204584

Gallardo-Salazar, J. L., and Pompa-García, M. (2020). Detecting individual tree attributes and multispectral indices using unmanned aerial vehicles: applications in a pine clonal orchard. *Remote Sens.* 12 (24), 4144. doi: 10.3390/rs12244144

García-Murillo, D. G., Caicedo-Acosta, J., and Castellanos-Dominguez, G. (2020). Individual detection of citrus and avocado trees using extended maxima transform summation on digital surface models. *Remote Sens.* 12 (10), 1633. doi: 10.3390/rs12101633

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 580–587. doi: 10.1109/CVPR.2014.81

Häni, N. (2020). MinneApple: a benchmark dataset for apple detection and segmentation. *IEEE Robotics. Automation. Lett.* 5 (2), 852–858. doi: 10.1002/ece3.5921

Hansen, O. L. P., Svenning, J. C., Olsen, K., Dupont, S., Garner, B. H., Iosifidis, A., et al. (2020). Species-level image classification with convolutional neural network enables insect identification from habitus images. *Ecol. Evol.* 10 (2), 737–747. doi: 10.1002/ece3.5921

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2980–2988. doi: 10.1109/ICCV.2017.322

He, M.-X., Hao, P., and Xin, Y.-Z. (2020). A robust method for wheatear detection using UAV in natural scenes. *IEEE Access* 8, 189043–189053. doi: 10.1109/ACCESS.2020.3031896

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. doi: 10.1109/CVPR.2016.90

Horstrand, P., Guerra, R., Rodríguez, A., Díaz, M., López, S., and López, J. F. (2019). A UAV platform based on a hyperspectral sensor for image capturing and on-board processing. *IEEE Access* 7, 66919–66938. doi: 10.1109/ACCESS.2019.2913957

Horton, R., Cano, E., Bulanon, D., and Fallahi, E. (2017). Peach flower monitoring using aerial multispectral imaging. *J. Imaging* 3 (2), 1–10. doi: 10.3390/jimaging3010002

Hu, X., and Li, D. (2020). Research on a single-tree point cloud segmentation method based on UAV tilt photography and deep learning algorithm. *IEEE J. Selected. Topics. Appl. Earth Observations. Remote Sens.* 13, 4111–4120. doi: 10.1109/JSTARS.2020.3008918

Hu, C. H., Shi, Z. F., Wei, H. L., Hu, X. D., Xie, Y. N., and Li, P. P. (2022). Automatic detection of pecan fruits based on Faster RCNN with FPN in orchard. *Int. J. Agric. Biol. Eng.* 15 (6), 189–196. doi: 10.25165/j.ijabe.20221506.7241

Hulens, D., Vandersteegen, M., and Goedemé, T. (2017). "Real-time vision-based UAV navigation in fruit orchards," in *Proc. 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*. 617–622. doi: 10.5220/0006242906170622

Ichim, L., Ciciu, R., and Popescu, D. (2022). "Using drones and deep neural networks to detect halyomorpha halys in ecological orchards," in *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 437–440. doi: 10.1109/IGARSS46834.2022.9883742

Ichim, L., and Popescu, D. (2020). Segmentation of vegetation and flood from aerial images based on decision fusion of neural networks. *Remote Sens.* 12 (15), 2490. doi: 10.3390/rs12152490

Iost Filho, F. H., Heldens, W., Kong, Z., and de Lange, E. S. (2020). Drones: innovative technology for use in precision pest management. *J. Econ. Entomol.* 113 (1), 1–25. doi: 10.1093/jee/toz268

IPPC Secretariat, Gullino, M. L., Albajes, R., Al-Jboory, I., Angelotti, F., Chakraborty, S., et al. (2021). *Scientific review of the impact of climate change on plant pests – A global challenge to prevent and mitigate plant pest risks in agriculture, forestry, and ecosystems* (Rome: FAO on behalf of the IPPC Secretariat). doi: 10.4060/cb4769en

Iqbal, M. S., Ali, H., Tran, S. N., and Iqbal, T. (2021). Coconut trees detection and segmentation in aerial imagery using mask region-based convolution neural network. *IET. Comput. Vis.* 15 (6), 428–439. doi: 10.1049/cvi2.12028

Jensen, K., Krogh, O. K., Jorgensen, M. W., Lehotsky, D., Andersen, A. B., Porqueras, E., et al. (2021). "Determining dendrometry using drone scouting, convolutional neuralnetworks and point clouds," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPRW)*. 2912–2920. doi: 10.1109/CVPRW53098.2021.00326

Jia, W., Liu, M., Luo, R., Wang, C., Pan, N., Yang, X., et al. (2022). YOLOF-Snake: an efficient segmentation model for green object fruit. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.765523

Jia, W., Tian, Y., Luo, R., Zhang, Z., Lian, J., and Zheng, Y. (2020). Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* 172, 105380. doi: 10.1016/j.compag.2020.105380

Johansen, K., Raharjo, T., and McCabe, M. F. (2018). Using multi-spectral UAV imagery to extract tree crop structural properties and assess pruning effects. *Remote Sens.* 10 (6), 854. doi: 10.3390/rs10060854

Ju, C., Kim, J., Seol, J., and Il Son, H. (2022). A review on multirobot systems in agriculture. *Comput. Electron. Agric.* 202, 107336. doi: 10.1016/j.compag.2022.107336

Jurado, J. M., Ortega, L., Cubillas, J. J., and Feito, F. R. (2020). Multispectral mapping on 3D models and multi-temporal monitoring for individual characterization of olive trees. *Remote Sens.* 12, 1–26, 1106. doi: 10.3390/rs12071106

Kalantar, A., Edan, Y., Gur, A., and Klapp, I. (2020). A deep learning system for single and overall weight estimation of melons using unmanned aerial vehicle images. *Comput. Electron. Agric.* 178, 1–11, 105748. doi: 10.1016/j.compag.2020.105748

Kamilaris, A., and Prenafeta-Boldú, F. (2018). A review of the use of convolutional neural networks in agriculture. *J. Agric. Sci.* 156 (3), 312–322. doi: 10.1017/S0021859618000436

Kang, H., and Chen, C. (2019). Fruit detection and segmentation for apple harvesting using visual sensor in orchards. *Sensors* 19 (20), 4599. doi: 10.3390/s19204599

Kang, H., and Chen, C. (2020a). Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* 171, 105302. doi: 10.1016/j.compag.2020.105302

Kang, H., and Chen, C. (2020b). Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* 168, 105108. doi: 10.1016/j.compag.2019.105108

Kattenborn, T., Eichel, J., and Fassnacht, F. (2019). Convolutional Neural Networks enable efficient, accurate and fine- grained segmentation of plant species and communities from high-resolution UAV imagery. *Sci. Rep.* 10, 17656. doi: 10.1038/s41598-019-53797-9

Kerkech, M., Hafiane, A., and Canals, R. (2018). Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Comput. Electron. Agric.* 155, 237–243. doi: 10.1016/j.compag.2018.10.006

Kerkech, M., Hafiane, A., and Canals, R. (2020). Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach. *Comput. Electron. Agric.* 174, 105446. doi: 10.1016/j.compag.2020.105446

Kestur, R., Meduri, A., and Narasipura, O. (2019). MangoNet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Eng. Appl. Artif. Intell.* 77, 59–69. doi: 10.1016/j.engappai.2018.09.011

Khan, A., Ilyas, T., Umraiz, M., Mannan, Z. I., and Kim, H. (2020a). CED-Net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. *Electronics* 9 (10), 1602. doi: 10.3390/electronics9101602

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020b). A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* 53, 5455–5516. doi: 10.1007/s10462-020-09825-6

Khan, S., Tufail, M., Khan, M., Ahmad, Z., and Anwar, S. (2021). Deep-learning-based spraying area recognition system for Unmanned-Aerial-Vehicle-based sprayers. *Turkish. J. Electrical. Eng. Comput. Sci.* 29, 241–256. doi: 10.3906/elk-2004-4

Kim, W.-S., Lee, D.-H., Kim, Y.-J., Kim, T., Hwang, R.-Y., and Lee, H.-J. (2020). Path detection for autonomous traveling in orchards using patch-based CNN. *Comput. Electron. Agric.* 175, 105620. doi: 10.1016/j.compag.2020.105620

Koirala, A., Walsh, K. B., and Wang, Z. (2021). Attempting to estimate the unseen—correction for occluded fruit in tree fruit load estimation by machine vision with deep learning. *Agronomy* 11 (2), 347. doi: 10.3390/agronomy11020347

Koirala, A., Walsh, K., Wang, Z., and McCarthy, C. (2019a). *MangoYOLO data set* (CQUniversity). Dataset. Available at: https://hdl.handle.net/10018/1261224.

Koirala, A., Walsh, K. B., Wang, Z., and McCarthy, C. (2019b). Deep learning – Method overview and review of use for fruit detection and yield estimation. *Comput. Electron. Agric.* 162, 219–234. doi: 10.1016/j.compag.2019.04.017

Lan, Y., Huang, Z., Deng, X., Zhu, Z., Huang, H., Zheng, Z., et al. (2020). Comparison of machine learning methods for citrus greening detection on UAV multispectral images. *Comput. Electron. Agric.* 171, 105234. doi: 10.1016/j.compag.2020.105234

La Rosa, L. E. C., Zortea, M., Gemignani, B. H., Oliveira, D. A. B., and Feitosa, R. Q. (2020). "FCRN-based multi-task learning for automatic citrus tree detection from UAV images," in *IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*. 403–408. doi: 10.1109/LAGIRS48042.2020.9165654

Larsen, M., Eriksson, M., Descombes, X., Perrin, G., Brandtberg, T., and Gougeon, F. A. (2011). Comparison of six individual tree crown detection algorithms evaluated under varying forest conditions. *Int. J. Remote Sens.* 32 (20), 5827–5852. doi: 10.1080/01431161.2010.507790

Lei, H., Huang, K., Jiao, Z., Tang, Y., Zhong, Z., and Cai, Y. (2022). Bayberry segmentation in a complex environment based on a multi-module convolutional neural network. *Appl. Soft. Computing.* 119, 108556. doi: 10.1016/j.asoc.2022.108556

Leskey, T. C., and Nielsen, A. L. (2018). Impact of the invasive brown marmorated stink bug in North America and Europe: history, biology, ecology, and management. *Annu. Rev. Entomol.* 63, 599–618. doi: 10.1146/annurev-ento-020117-043226

Li, J.-M., Chen, C.-W., and Cheng, T.-H. (2021). Motion prediction and robust tracking of a dynamic and temporarily - occluded target by an Unmanned Aerial Vehicle. *IEEE Transactions on Control Systems Technology* 29, 4, 1623–1163. doi: 10.1109/TCST.2020.3012619

Li, Y., Ren, J., and Huang, Y. (2020). An end-to-end system for Unmanned Aerial Vehicle high-resolution remote sensing image haze removal algorithm using convolution neural network. *IEEE Access* 8, 158787–158797. doi: 10.1109/ACCESS.2020.3020359

Li, D., Sun, X., Elkhouchlaa, H., Jia, Y., Yao, Z., Lin, P., et al. (2021). Fast detection and location of longan fruits using UAV images. *Comput. Electron. Agric.* 190, 106465. doi: 10.1016/j.compag.2021.106465

Li, D., Sun, X., Lv, S., Elkhouchlaa, H., Jia, Y., Yao, Z., et al. (2022). A novel approach for the 3D localization of branch picking points based on deep learning applied to longan harvesting UAVs. *Comput. Electron. Agric.* 199, 107191. doi: 10.1016/j.compag.2022.107191

Li, L., Zhang, S., and Wang, B. (2021). Plant disease detection and classification by deep learning—a review. *IEEE Access* 9, 56683–56698. doi: 10.1109/ACCESS.2021.3069646

Li, W., Zhu, X., Yu, X., Li, M., Tang, X., Zhang, J., et al. (2022). Inversion of nitrogen concentration in apple canopy based on UAV hyperspectral images. *Sensors. 22 (9)* 1–14, 3503. doi: 10.3390/s22093503

Lin, Z., and Guo, W. (2020). Sorghum panicle detection and counting Using Unmanned Aerial system images and deep learning. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.534853

Lin, C., Jin, Z., Mulla, D., Ghosh, R., Guan, K., Kumar, V., et al. (2021). Toward large-scale mapping of tree crops with high-resolution satellite imagery and deep learning algorithms: a case study of olive orchards in Morocco. *Remote Sens.* 13 (9), 1740. doi: 10.3390/rs13091740

Lin, P., Li, D., Jia, Y., Chen, Y., Huang, G., Elkhouchlaa, H., et al. (2022). A novel approach for estimating the flowering rate of litchi based on deep learning and UAV images. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.966639

Liu, C., Li, H., Su, A., Chen, S., and Li, W. (2021). Identification and grading of maize drought on RGB images of UAV based on improved U-Net. *IEEE Geosci. Remote Sens. Lett.* 18 (2), 198–202. doi: 10.1109/LGRS.2020.2972313

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00898

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 22. doi: 10.1186/s13007-021-00722-9

Lobo Torres, D., Queiroz Feitosa, R., Nigri Happ, P., Elena Cué La Rosa, L., Marcato Junior, J., Martins, J., et al. (2020). Applying fully convolutional architectures for semantic segmentation of a single tree species in urban environment on high resolution UAV optical imagery. *Sensors* 20, 2, 563. doi: 10.3390/s20020563

Lu, Y., and Young, S. (2020). A survey of public datasets for computer vision tasks in precision agriculture. *Comput. Electron. Agric.* 178, 105760. doi: 10.1016/j.compag.2020.105760

Lyu, S., Li, R., Zhao, Y., Li, Z., Fan, R., and Liu, S. (2022). Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system. *Sensors* 22 (2), 576. doi: 10.3390/s22020576

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., and Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS. J. Photogrammetry. Remote Sens.* 152, 166–177. doi: 10.1016/j.isprsjprs.2019.04.015

Machefer, M., Lemarchand, F., Bonnefond, V., Hitchins, A., and Sidiropoulos, P. (2020). Mask R-CNN refitting strategy for plant counting and sizing in UAV imagery. *Remote Sens.* 12, 3015. doi: 10.3390/rs12183015

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., et al. (2020). Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* 170, 105277. doi: 10.1016/j.compag.2020.105277

Marmanis, D., Wegner, J. D., Galliani, S., Schindler, K., Datcu, M., and Stilla, U. (2016). Semantic segmentation of aerial images with an ensemble of CNNs. *ISPRS. Ann. Photogramm. Remote Sens. Spatial. Inf. Sci.* III-3, 473–480. doi: 10.5194/isprs-annals-III-3-473-2016

Menshchikov, A., Shadrin, D., Prutyanov, V., Lopatkin, D., Sosnin, S., and Tsykunov, E. (2021). Real-time detection of hogweed: UAV platform empowered by deep learning. *IEEE Transactions on Computers* 70(8), 1175–1188. doi: 10.1109/TC.2021.3059819

Mesas-Carrascosa, F.-J., Pérez-Porras, F., Meroño de Larriva, J. E., Mena Frau, C., Agüera-Vega, F., Carvajal-Ramírez, F., et al. (2018). Drift correction of lightweight microbolometer thermal sensors on-board Unmanned Aerial Vehicles. *Remote Sens.* 10 (4), 615. doi: 10.3390/rs10040615

Miyoshi, G. T., Arruda, M., Osco, L. P., Marcato Junior, J., Gonçalves, D. N., Imai, N. N., et al. (2020). A novel deep learning method to identify single tree species in UAV-based hyperspectral images. *Remote Sens.* 12 (8), 1294. doi: 10.3390/rs12081294

Modica, G., Messina, G., De Luca, G., Fiozzo, V., and Praticò, S. (2020). Monitoring the vegetation vigor in heterogeneous citrus and olive orchards. A multiscale object-based approach to extract trees' crowns from UAV multispectral imagery. *Comput. Electron. Agric.* 175, 105500. doi: 10.1016/j.compag.2020.105500

Mokrane, A., Braham, A. C., and Cherki, B. (2019). "UAV coverage path planning for supporting autonomous fruit counting systems," in *Proc. International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*. 1–5. doi: 10.1109/ICAAID.2019.8934989

Mu, Y., Fujii, Y., Takata, D., Zheng, B., Noshita, K., Hond, K., et al. (2018). Characterization of peach tree crown by using high-resolution images from an Unmanned Aerial Vehicle. *Hortic. Res.* 5, 74. doi: 10.1038/s41438-018-0097-z

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., and Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* 10 (10), 3443. doi: 10.3390/app10103443

Nawaz, S. A., Li, J., Bhatti, U. A., Shoukat, M. U., and Ahmad, R. M. (2022). AI-based object detection latest trends in remote sensing, multimedia and agriculture applications. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1041514

Nevalainen, O., Honkavaara, E., Tuominen, S., Viljanen, N., Hakala, T., Yu, X., et al. (2017). Individual tree detection and classification with UAV-based photogrammetric point clouds and hyperspectral imaging. *Remote Sens.* 9 (3), 185. doi: 10.3390/rs9030185

Nguyen, H. T., Lopez Caceres, M. L., Moritake, K., Kentsch, S., Shu, H., and Diez, Y. (2021). Individual sick fir tree (Abies mariesii) identification in insect infested forests by means of UAV images and deep learning. *Remote Sens.* 13, 260. doi: 10.3390/rs13020260

Niu, H., Wang, D., and Chen, Y. (2020). Estimating crop coefficients using linear and deep stochastic configuration networks models and UAV-based normalized difference vegetation index (NDVI). *Proc. Int. Conf. Unmanned. Aircraft. Syst. (ICUAS).*, 1485–1490. doi: 10.1109/ICUAS48674.2020.9213888

Noguera, M., Aquino, A., Ponce, J. M., Cordeiro, A., Silvestre, J., Arias-Calderón, R., et al. (2021). Nutritional status assessment of olive crops by means of the analysis and modelling of multispectral images taken with UAVs. *Biosyst. Eng.* 211, 1–18. doi: 10.1016/j.biosystemseng.2021.08.035

Ochoa, K. S., and Guo, Z. (2019). A framework for the management of agricultural resources with automated aerial imagery detection. *Comput. Electron. Agric.* 162, 53–69. doi: 10.1016/j.compag.2019.03.028

Oliveira, A. J., Assis, G. A., Faria, E. R., Souza, J. R., Vivaldini, K. C. T., Guizilini, V., et al. (2019). "Analysis of nematodes in coffee crops at different altitudes using aerial images," in *Proc. 27th European Signal Processing Conference (EUSIPCO)*. 1–5. doi: 10.23919/EUSIPCO.2019.8902734

Olson, D., and Anderson, J. (2021). Review on unmanned aerial vehicles, remote sensors, imagery processing, and their applications in agriculture. *Agron. J.* 113, 1–22. doi: 10.1002/agj2.20595

Osco, L., dos Santos de Arruda, M., Junior, J. M., da Silva, N. B., Ramos, A. P., Moryia, É.A.S., et al. (2020). A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS. J. Photogrammetry. Remote Sens.* 160, 97–106. doi: 10.1016/j.isprsjprs.2019.12.010

Osco, L. P., Nogueira, K., Marques Ramos, A. P., Pinheiro, M. M. F., Furuya, D. E. G., Gonçalves, W., et al. (2021). Semantic segmentation of citrus-orchard using deep neural networks and multispectral UAV-based imagery. *Precis. Agric.* 22, 1171–1188. doi: 10.1007/s11119-020-09777-5

Özyurt, H. B., Duran, H., and Çelen, İ.H. (2022). Determination of the application parameters of spraying drones for crop production in hazelnut orchards. *J. Tekirdag. Agric. Faculty.* 19 (4), 819–828. doi: 10.33462/jotaf.1105420

Pádua, L., Adão, T., Hruška, J., Guimarães, N., Marques, P., Peres, E., et al. (2020). "Vineyard classification using machine learning techniques applied to RGB-UAV imagery," in *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 6309–6312. doi: 10.1109/IGARSS39084.2020.9324380

Park, J., Kim, D. I., Choi, B., Kang, W., and Kwon, H. W. (2020). Classification and morphological analysis of vector mosquitoes using deep convolutional neural networks. *Sci. Rep.* 10, 1012. doi: 10.1038/s41598-020-57875-1

Pederi, Y. A., and Cheporniuk, H. S. (2015). "Unmanned Aerial Vehicles and new technological methods of monitoring and crop protection in precision agriculture," in *Proc. IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. 298–301. doi: 10.1109/APUAVD.2015.7346625

Peng, H., Xu, H., Gao, Z., Zhou, Z., Tian, X., Deng, Q., et al. (2023). Crop pest image classification based on improved densely connected convolutional network. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1133060

Popescu, D., El-Khatib, M., and Ichim, L. (2022a). Skin lesion classification using collective intelligence of multiple neural networks. *Sensors* 22 (12), 4399. doi: 10.3390/s22124399

Popescu, D., Serghei, T.-L., and Ichim, L. (2022b). "Dual networks based system for detecting and classifying harmful insects in orchards," in *Proc. International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. 1–6. doi: 10.1109/ICECCME55909.2022.9988360

Popescu, D., Stoican, F., Stamatescu, G., Ichim, L., and Dragana, C. (2020). Advanced UAV–WSN system for intelligent monitoring in precision agriculture. *Sensors* 20 (3), 817. doi: 10.3390/s20030817

Pradeep, P., Park, S. G., and Wei, P. (2018). "Trajectory optimization of multirotor agricultural UAVs," in *IEEE Aerospace Conference*. 1–7. doi: 10.1109/AERO.2018.8396617

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Ronchetti, G., Mayer, A., Facchi, A., Ortuani, B., and Sona, G. (2020). Crop row detection through UAV surveys to optimize on-farm irrigation management. *Remote Sens.* 12, 1967. doi: 10.3390/rs12121967

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: convolutional networks for biomedical image segmentation. *Lecture. Notes Comput. Sci.* 9351, 234–241. doi: 10.1007/978-3-319-24574-4_28

Roosjen, P. P., Kellenberger, B., Kooistra, L., Green, D. R., and Fahrentrapp, J. (2020). Deep learning for automated detection of Drosophila Suzukii: potential for UAV-based monitoring. *Pest Manag. Sci.* 76 (9), 2994–3002. doi: 10.1002/ps.5845

Rosa, L. E. C. L., Oliveira, D. A. B., Zortea, M., Gemignani, B. H., and Feitosa, R. Q. (2020). Learning geometric features for improving the automatic detection of citrus plantation rows in UAV images. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2020.3024641

Safonova, A., Guirado, E., Maglinets, Y., Alcaraz-Segura, D., and Tabik, S. (2021). Olive tree biovolume from UAV multi-resolution image segmentation with Mask R-CNN. *Sensors* 21 (5), 1617. doi: 10.3390/s21051617

Santos, T. T., de Souza, L. L., dos Santos, A. A., and Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Electron. Agric.* 170, 105247. doi: 10.1016/j.compag.2020.105247

Sarabia, R., Aquino, A., Ponce, J. M., López, G., and Andújar, J. M. (2020). Automated identification of crop tree crowns from UAV multispectral imagery by means of morphological image analysis. *Remote Sens.* 12 (5), 748. doi: 10.3390/rs12050748

Schiefer, F., Kattenborn, T., Frick, A., Frey, J., Schall, P., Koch, B., et al. (2020). Mapping forest tree species in high resolution UAV-based RGB-imagery by means of convolutional neural networks. *ISPRS. J. Photogrammetry. Remote Sens.* 170, 205–215. doi: 10.1016/j.isprsjprs.2020.10.015

Schoofs, H., Delalieux, S., Deckers, T., and Bylemans, D. (2020). Fire blight monitoring in pear orchards by Unmanned Airborne Vehicles (UAV) systems carrying spectral sensors. *Agronomy* 10 (5), 615. doi: 10.3390/agronomy10050615

Shang, G., Liu, G., Zhu, P., Han, J., Xia, C., and Jiang, K. (2021). A deep residual U-Type network for semantic segmentation of orchard environments. *Applied Sciences* 11 (1), 1-13, 322. doi: 10.3390/app11010322

Sorbelli, F. B., Corò, F., Das, S. K., Di Bella, E., Maistrello, L., Palazzetti, L., et al. (2022). "**A** drone-based application for scouting Halyomorpha halys bugs in orchards with multifunctional nets," in *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 127–129. doi: 10.1109/PerComWorkshops53856.2022.9767309

Stefas, N., Bayram, H., and Isler, V. (2016). Vision-based UAV navigation in orchards. *IFAC-PapersOnLine* 49 (16), 10–15. doi: 10.1016/j.ifacol.2016.10.003

Sulistijono, I. A., Ramadhani, M. R., and Risnumawan, A. (2020). "Aerial drone mapping and trajectories generator for agricultural ground robots," in *Proc. International Symposium on Community-centric Systems (CcS)*. 1–6. doi: 10.1109/CcS49175.2020.9231397

Sun, H., Wang, B., and Xue, J. (2023). YOLO-P: An efficient method for pear fast detection in complex orchard picking environment. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1089454

Tang, M., Sadowski, D. L., Peng, C., Vougioukas, S. G., Klever, B., Khalsa, S. D. S., et al. (2023). Tree-level almond yield estimation from high resolution aerial imagery with convolutional neural network. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1070699

Torres-Sánchez, J., de Castro, A. I., Peña, J. M., Jiménez-Brenes, F. M., Arquero, O., Lovera, M., et al. (2018). Mapping the 3D structure of almond trees using UAV acquired photogrammetric point clouds and object-based image analysis. *Biosyst. Eng.* 176, 172–184. doi: 10.1016/j.biosystemseng.2018.10.018

Torres-Sánchez, J., López-Granados, F., Serrano, N., Arquero, O., and Peña, J. M. (2015). High-throughput 3-d monitoring of agricultural-tree plantations with unmanned aerial vehicle (UAV) technology. *PloS One* 10 (6), e0130479. doi: 10.1371/journal.pone.0130479

Tu, Y.-H., Phinn, S., Johansen, K., Robson, A., and Wu, D. (2020). Optimising drone flight planning for measuring horticultural tree crop structure. *ISPRS. J. Photogrammetry. Remote Sens.* 160, 83–96. doi: 10.1016/j.isprsjprs.2019.12.006

Vélez, S., Vacas, R., Martín, H., Ruano-Rosa, D., and Álvarez, S. (2022). High-resolution UAV RGB imagery dataset for precision agriculture and 3D photogrammetric reconstruction captured over a pistachio orchard (Pistacia vera L.) in Spain. *Data* 7 (11), 157. doi: 10.3390/data7110157

Wang, Q., Cheng, M., Xiao, X., Yuan, H., Zhu, J., Fan, C., et al. (2021). An image segmentation method based on deep learning for damage assessment of the invasive weed Solanum Rostratum Dunal. *Comput. Electron. Agric.* 188, 106320. doi: 10.1016/j.compag.2021.106320

Wang, C., Liu, S., Wang, Y., Xiong, J., Zhang, Z., Zhao, B., et al. (2022). Application of convolutional neural network-based detection methods in fresh fruit production: a comprehensive review. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.868745

Wang, S., Zhang, X., Shen, H., Tian, M., and Li, M. (2022). "Research on UAV online visual tracking algorithm based on YOLOv5 and FlowNet2 for apple yield inspection," in *Proc. WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. 280–285. doi: 10.1109/WRCSARA57040.2022.9903925

Wu, X., Zhan, C., Lai, Y.-K., Cheng, M.-M., and Yang, J. (2019). "IP102: A largescale benchmark dataset for insect pest recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8779–8788. doi: 10.1109/cvpr.2019.00899

Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., et al. (2020). A review of object detection based on deep learning. *Multimed. Tools Appl.* 79 (33), 23729–23791. doi: 10.1007/s11042-020-08976-6

Xie, T., Li, J., Yang, C., Jiang, Z., Chen, Y., Guo, L., et al. (2021). Crop height estimation based on UAV images: methods, errors, and strategies. *Comput. Electron. Agric.* 185, 106155. doi: 10.1016/j.compag.2021.106155

Xing, S., Lee, M., and Lee, K.-k. (2019). Citrus pests and diseases recognition model using weakly dense connected convolution network. *Sensors* 19 (14), 3195. doi: 10.3390/s19143195

Yang, M.-D., Tseng, H.-H., Hsu, Y.-C., and Tsai, H. P. (2020). Semantic segmentation using deep learning with vegetation indices for rice lodging identification in multi-date UAV visible images. *Remote Sensing* 12 (4), 1–20, 630. doi: 10.3390/rs12040633

Yuan, W., and Choi, D. (2021). UAV-based heating requirement determination for frost management in apple orchard. *Remote Sens.* 13 (2), 273. doi: 10.3390/rs13020273

Zhang, W., Chen, X., Qi, J., and Yang, S. (2022). Automatic instance segmentation of orchard canopy in unmanned aerial vehicle imagery using deep learning. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1041791

Zhang, X., Fu, L., Karkee, M., Whiting, M. D., and Zhang, Q. (2019). Canopy segmentation using ResNet for mechanical harvesting of apples. *IFAC-PapersOnLine* 52 (30), 300–305. doi: 10.1016/j.ifacol.2019.12.550

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., and Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). *Comput. Electron. Agric.* 155, 386–393. doi: 10.1016/j.compag.2018.10.029

Zhang, X., Karkee, M., Zhang, Q., and Whiting, M. D. (2021). Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. *J. Field Robotics.* 38, 476–493. doi: 10.1002/rob.21998

Zhang, Q., Liu, Y., Gong, C., Chen, Y., and Yu, H. (2020). Applications of deep learning for dense scenes analysis in agriculture: a review. *Sensors* 20 (5), 1520. doi: 10.3390/s20051520

Zhang, C., Valente, J., Kooistra, L., Guo, L., and Wang, W. (2019). Opportunities of UAVs in orchard management. *Int. Arch. Photogramm. Remote Sens. Spatial. Inf. Sci.* XLII-2/W13, 673–680. doi: 10.5194/isprs-archives-XLII-2-W13-673-2019

Zhang, C., Valente, J., Kooistra, L., Guo, L., and Wang, W. (2021). Orchard management with small unmanned aerial vehicles: a survey of sensing and analysis approaches. *Precis. Agric.* 22, 2007–2052. doi: 10.1007/s11119-021-09813-y

Zhang, H., Wang, X., Chen, Y., Jiang, G., and Lin, S. (2019). Research on vision-based navigation for plant protection UAV under the near color background. *Symmetry* 11 (4), 533. doi: 10.3390/sym11040533

Zhang, S., and Zhang, C. (2023). Modified U-Net for plant diseased leaf image segmentation. *Comput. Electron. Agric.* 204, 107511. doi: 10.1016/j.compag.2022.107511

Zhang, Y., Zhang, W., Yu, J., He, L., Chen, J., and He, Y. (2022). Complete and accurate holly fruits counting using YOLOX object detection. *Comput. Electron. Agric.* 198, 107062. doi: 10.1016/j.compag.2022.107062

Zheng, Z., Xiong, J., Lin, H., Han, Y., Sun, B., Xie, Z., et al. (2021). A method of green citrus detection in natural environments using a deep convolutional neural network. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.705737

Zhu, J., Cheng, M., Wang, Q., Yuan, H., and Cai, Z. (2021). Grape leaf black rot detection based on super-resolution image enhancement and deep learning. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.695749

Zortea, M., Macedo, M., Mattos, A. B., Bernardo, R., and Gemignani, B. H. (2018). "Automatic citrus tree detection from UAV images based on convolutional neural networks," in *Proc. 31st Sibgrap/WIA - Conference on Graphics, Patterns and Images*. 1–8.

Glossary

AKAZE	Accelerated-KAZE
ACC	Accuracy
AI	Artificial Intelligence
AP	Average Precision
ATSS	Adaptive Training Sample Selection
CRF	Conditional Random Field
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CR	Capturing rate
DASNet	Dual Attentive fully convolutional Siamese Network
DB	Database
DCNN	Deep Convolutional Neural Network
DDCN	Dynamic Dilated Convolution Network
DeepSCN	Deep Stochastic Configuration Network
DL	Deep Learning
DR	Detection Rate
DS	Dataset
DSC	Dice Coefficient
DSM	Digital Surface Model
DTM	Digital Terrain Model
F1	Dice Coefficient (F1 Measure)
FCN	Fully Convolutional Network
FCRN	Fully Convolutional Regression Network
FCRN-MTL	Fully Convolutional Regression Network Multi-Task Learning
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Networks
FSAF	Feature Selective Anchor-Free
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HRNet	High Resolution Network
IoT	Internet of Things
IoU	Intersection-Over-Union
KNN	K-Nearest Neighbor
mAP	Mean Average Precision
ML	Machine Learning
NDVI	Normalized Difference Vegetation Index
NIR	Near-infrared
NN	Artificial Neural Network
ODM	Open Drone Map
PRE	Precision
PSPNet	Pyramid Scene Parsing Network
RBF	Radial Basis Function
R-CNN	Region-Based CNN
ResNet	Residual Neural Network
RGB	Red-Green-Blue (images)
RTK	Real-Time Kinematic Positioning
RoI	Region of Interest
ROS	Robot Operating System
SAE	System Architecture Evolution
SAR	Synthetic-aperture radar
SegNet	Semantic Segmentation Network
SEN	Sensitivity
SPE	Specificity
SPP	Spatial Pyramid Pooling
SR	Statistical Rate
SSD	Single Shot MultiBox Detector
TN	True Negative
TP	True Positive
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
VGG	Visual Geometry Group
WOS	Web of Science
YOLO	You Only Look Once

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 28 November 2023
DOI 10.3389/fpls.2023.1241921

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Alejandro Román Vázquez,
Spanish National Research Council (CSIC), Spain
Giuseppe Modica,
University of Messina, Italy

*CORRESPONDENCE
Tejasri Nampally
ai19resch11002@iith.ac.in

RECEIVED 17 June 2023
ACCEPTED 08 November 2023
PUBLISHED 28 November 2023

CITATION
Nampally T, Kumar K, Chatterjee S, Pachamuthu R, Naik B and Desai UB (2023) StressNet: a spatial-spectral-temporal deformable attention-based framework for water stress classification in maize. *Front. Plant Sci.* 14:1241921. doi: 10.3389/fpls.2023.1241921

StressNet: a spatial-spectral-temporal deformable attention-based framework for water stress classification in maize

Tejasri Nampally[1]*, Kshitiz Kumar[1], Soumyajit Chatterjee[1], Rajalakshmi Pachamuthu[2], Balaji Naik[3] and Uday B. Desai[2]

[1]Department of Artificial Intelligence, Indian Institute of Technology (IIT) Hyderabad, Hyderabad, India, [2]Department of Electrical Engineering, Indian Institute of Technology (IIT) Hyderabad, Hyderabad, India, [3]Department of Agronomy, Professor Jayashankar Telangana State Agricultural University (PJTSAU), Hyderabad, India

In recent years, monitoring the health of crops has been greatly aided by deploying highthroughput crop monitoring techniques that integrate remotely captured imagery and deep learning techniques. Most methods rely mainly on the visible spectrum for analyzing the abiotic stress, such as water deficiency in crops. In this study, we carry out experiments on maize crop in a controlled environment of different water treatments. We make use of a multispectral camera mounted on an Unmanned Aerial Vehicle for collecting the data from the tillering stage to the heading stage of the crop. A pre-processing pipeline, followed by the extraction of the Region of Interest from orthomosaic is explained. We propose a model based on a Convolution Neural Network, added with a deformable convolutional layer in order to learn and extract rich spatial and spectral features. These features are further fed to a weighted Attention-based Bi-Directional Long Short-Term Memory network to process the sequential dependency between temporal features. Finally, the water stress category is predicted using the aggregated Spatial-Spectral-Temporal Characteristics. The addition of multispectral, multi-temporal imagery significantly improved accuracy when compared with mono-temporal classification. By incorporating a deformable convolutional layer and Bi-Directional Long Short-Term Memory network with weighted attention, our proposed model achieved best accuracy of 91.30% with a precision of 0.8888 and a recall of 0.8857. The results indicate that multispectral, multi-temporal imagery is a valuable tool for extracting and aggregating discriminative spatial-spectral-temporal characteristics for water stress classification.

KEYWORDS

multispectral, multitemporal, UAV, stress classification, maize, BiLSTM, attention-based network

1 Introduction

The growth and health of the crop depend on several essential agronomic inputs Boyer (1982) such as water and soil nutrients like nitrogen and phosphorous. These factors play a pivotal role in determining both the quantity and quality of production. Water aids in the transportation of nutrients Gonzalez-Dugo et al. (2010) from the soil to different regions of the plant. Inadequate water supply leads to the development of abiotic stress in plants, disrupting their capacity Wang et al. (2016); Vicente et al. (2018) to carry out vital processes such as photosynthesis, affecting the crop's yield. In the recent past, the phenomenon of global warming Mueller et al. (2012); Food and of the United Nations (2019) resulted in irregular rainfall patterns leading to water scarcity. Water shortage leads to diverse physiological changes, including loss of greenness and reduced leaf surface and biomass. Maize is a staple food around the globe and accounts for 36% of the world's grain production, constituting nearly 9% of the Indian food basket Dataset IIMR (2020). Since there are about one to two kernels per plant, drought stress impacts Zhou et al. (2020); Liu et al. (2020) the quality, harvesting ability, and crop yield. As per the recent study by Laborde et al. (2020), the pandemic in 2019 (COVID) resulted in uncertainties in global food security. Owing to the potential that maize occupies a significant amount towards ensuring the food supply, especially in developing nations like India, it is necessary to advance crop monitoring methods through comprehensive geographical evaluation. Accurate determination of optimal timing and quantity of water will facilitate enhanced irrigation.

Over the last decade, remote sensing methods have been extensively used by Semmens et al. (2016); Thorp et al. (2018); Tian et al. (2020) for characterizing water stress in crops. Aerial-based remote sensing emerged as a non-invasive technique to gather data from crop, soil, and environmental factors. It made a significant impact by obtaining "farm" level to "leaf" level information through image data. Further, this data helped Berni et al. (2009); Al-Tamimi et al. (2022) in quantifying various traits of water stress responses. Of the current aerial remote sensing techniques, Unmanned Aerial Vehicles (UAVs) have surfaced as efficient platforms for high-throughput phenotyping to monitor crop fields due to their high spatial and temporal resolution, further resulting in the improvement of the management of water stress in agriculture. UAVs can be accommodated with different types of camera sensors. They can fly at lower altitudes, cost-effective, enabling increased monitoring frequencies Berni et al. (2009); Araus and Cairns (2014); Gago et al. (2015).

Over the recent years in the field of computer vision, from conventional image processing techniques to present novel methods, automated learning-based feature extraction techniques have made substantial progress Li et al. (2020). These popular techniques include Support Vector Machine, K-Means clustering, and Random Forest. Moreover, Deep Learning (DL), a method that leverages LeCun et al. (2015) hierarchical feature extraction from images, has opened up new possibilities for interpreting vast amounts of data and permeated the field of data analytics in the field of agriculture. The plant science community is increasingly embracing these DL methods to extract meaningful insights from the extensive datasets gathered through high-throughput phenotyping and genotyping methods Kamilaris and Prenafeta-Boldú (2018); Zhong et al. (2019); Wang et al. (2022). Convolutional Neural Networks (CNNs) have gained popularity among Deep Learning methods for their ability to automatically extract valuable information from diverse features such as colour, shape, texture, size, and spectral information across different levels without the need for human expertise Krizhevsky et al. (2012); Grinblat et al. (2016); Lee et al. (2017). The exhaustive review from Singh et al. (2018) offers a thorough evaluation of DL methods applied to a broad spectrum of plant species, focusing on tasks such as identifying, classifying, quantifying, and predicting plant stress. The other studies of Kumar et al. (2020); Tejasri et al. (2022) explored UAV-captured imagery for predicting water stress-affected crops using CNN-based frameworks. These studies highlight that Red, Green, and Blue (RGB) bands are crucial for classifying water-stressed crops due to their rich properties of colour and texture. However, RGB bands are particularly light-sensitive and can only provide details within the visible spectrum Nijland et al. (2014). Moreover, multispectral data is of paramount importance due to its additional spectral information greatly aided Zarco-Tejada et al. (2012); Nijland et al. (2014); Wang et al. (2022) to overcome the light sensitivity issues in the visible spectral domain and helps in identifying the underlying information on crop water stress.

Earlier studies by Spišić et al. (2022); Barradas et al. (2021), utilized multispectral data and Supervised Machine Learning (ML) based methods to effectively detect drought stress in crops. These methods used MultiLayer Perceptron (MLP), Support Vector Machine (SVM), decision tree, Random Forest based classifiers, and gradient boosting techniques to classify water stressed plants. Virnodkar et al. (2020) conducted an extensive review on the use of supervised ML methods for crop water stress classification using UAV captured multispectral imagery. However, these described methods are mainly limited to manual feature extraction and thus are inefficient, particularly when dealing with high dimensional data or in complex environments Wang et al. (2022); Bouguettaya et al. (2022). This inherent limitation of traditional machine learning techniques has prompted a shift in focus towards machine learning methods based on DL LeCun et al. (2015).

By leveraging DL techniques with multispectral data, a significant transformation is occurring within the domain of data-centric agriculture. While CNNs show promising results in water stress detection and classification, as demonstrated by Kumar et al. (2020), they do not take temporal data into account. CNNs are limited by the assumption that data captured at different time points are equivalent. However, it is well-known that visual changes resulting from water stress in crop occur gradually and are not immediately discernible. This poses a challenge for CNNs, as they lack the ability to effectively learn temporal patterns, resulting in difficulties in confidently classifying stress conditions, as discussed by Singh et al. (2018); Gao et al. (2020). Moreover, the time-invariant nature of CNNs requires data displaying severe signs of stress for reliable detection, making it impractical for early identification and recovery of stressed plants. Therefore, there is an increasing need for a technique capable of analyzing the

progressive visual changes in stressed plants, enabling confident classification even in the absence of severe stress signs, facilitating early-stage water stress classification, and addressing a critical gap in current methods. In this context, Elsherbiny et al. (2022) explored a CNN-LSTM approach to assess the water status of wheat. This study aggregated features derived from RGB images, climatic conditions, and soil moisture, achieving a remarkably low loss of 0.0012. In our preliminary study Tejasri et al. (2023), we utilized CNNs (AlexNet, VGG-19, ResNet18, ResNet-50) for extracting the features from multi-temporal multispectral UAV-captured maize data. The extracted visual features are further fed to a single LSTM unit for capturing temporal dependencies. The results showed that the model based on fine-tuned ResNet-18 backbone, using multispectral data outperformed with a precision of 0.9765 and a recall of 0.9457 rather than just using RGB data with a precision of 0.9523 and a recall of 0.9487. On the other hand, considering the change in environment and the crop conditions, this analysis becomes difficult with the help of a single LSTM unit.

Thus, a series of LSTM units can be made use of where the input to these units are the sequences of visual features that are extracted by CNNs to preserve the temporal patterns as demonstrated by Azimi et al. (2021), for identifying water stress in chickpea plant. This approach gained more insights by providing a more accurate representation of the relationship between the environmental conditions and the crop's response. The sampling positions of standard convolution kernels remain constant. They cannot be adjusted to accommodate intricate spatial patterns in crop classification, as noted by Feng et al. (2020) in their work on multispectral image analysis. In addition, the classic pooling layers (average or max pooling) are also fixed and do not possess the capability to learn the downsampled features. Conversely, deformable convolution proposed by Dai et al. (2017), enables the neural network to adaptively adjust the sampling locations, allowing it to effectively capture the spatially varying patterns. Deformable convolution is an extension of standard CNN by introducing learnable offsets to the standard grid sampling locations of convolution kernels. Studies by Zhu et al. (2018) explored a deformable convolution neural network (DCNN) for hyperspectral image classification. Feng et al. (2020) adopted a deformable CNN-LSTM-based network for vegetable mapping from multi-temporal UAV-based RGB imagery. Motivated by the works mentioned above, we propose a model entitled StressNet which combines a deformable based CNN and a BiLSTM with weighted attention to dynamically adjust the receptive field to accommodate the size of the crop according to its growth stage.

In this study, we present a DL-based temporal analysis pipeline for classifying water-stressed crops, utilizing multispectral data captured by UAV. We aim to showcase the great performance of the proposed method compared to standard CNN, which is time-invariant and only spatial. The following contributions are obtained from the present work:

1. Dataset is created by using multispectral data of maize crop captured by UAV.
2. Our proposed model leverage the capabilities of CNN by adding deformable convolutional layer and BiLSTM for enhanced performance. It is specifically designed to learn spatial-spectral-temporal patterns for identifying water stressed crops.
3. We conducted a comparative analysis of the proposed method using CNN based architectures - AlexNet and VGG-16.
4. We performed an ablation study by evaluating the impact of temporal and spectral data using the proposed model. This involved systematically reducing the number of temporal data used and the number of spectral channels. In addition, we discussed the impact of the deformable convolutional layer, BiLSTM and weighted attention on the performance of the proposed method.

2 Materials and methods

2.1 Experimental site

The experimental study was conducted in a semi-arid zone of Hyderabad (Telangana, India) from October to February (post-monsoon season - Rabi) during 2018-19. The study area lies between 17°19'27.2"N – 17°19'28.3"N and 78°23'55.4"E – 78°23'56.2"E shown in Figures 1A, B. Rabi season was particularly chosen to precisely understand the water stress effect on the crop as the crop can be induced by heavy water stress conditions as the rainfall level is comparatively low during this period. The farm is situated in a semi-arid region, characterized by an average annual precipitation of 822 mm and annual potential evapotranspiration ranging from 1700 to 1960 mm. The soil in this area is predominantly composed of light red sandy loam and extends to a depth of approximately one meter and bedrock beneath it. For the study, maize crop (*Zea mays L.*) of the *'Cargill 900 M Gold'* variety is cultivated. The farm was maintained by *Agro Climate Research Center, Professor Jayashankar Telangana State Agriculture University* (PJTSAU), Hyderabad, India. The experimental field comprises 30 regions, each measuring 4.2 m × 4.8 m. The experimental field was designed in a split mode with three irrigation and nitrogen supply levels based on a climatic approach Halagalimath et al. (2017).

The determination of the irrigation schedule was based on Reddy and Reddy (2019) the ratio of Irrigation Water (IW) to Cumulative Pan Evaporation(CPE). Three distinct irrigation levels are chosen, with IW/CPE ratios of 0.6, 0.8, and 1 assigned to the respective regions. For each irrigation event, a uniform quantity of 50 mm water (IW) is provided to the designated plots using pipes equipped with water meters to ensure accurate measurement. Pan evaporimeters (in mm) are used to record daily readings, aiding in the calculation of the IW/CPE ratio. This ratio was crucial in determining the ideal timing for irrigation across various regions. Additionally, each type of irrigation plot is subjected to one of three nitrogen fertilization levels: 100, 200, and 300 kg nitrogen per hectare, as represented in Table 1. By combining the three irrigation levels with the three fertilization levels, a total of nine distinct regions are created. Furthermore, each plot is replicated three times, resulting in a total of 27 plots (3 water levels × 3 nitrogen levels × 3 replications), as

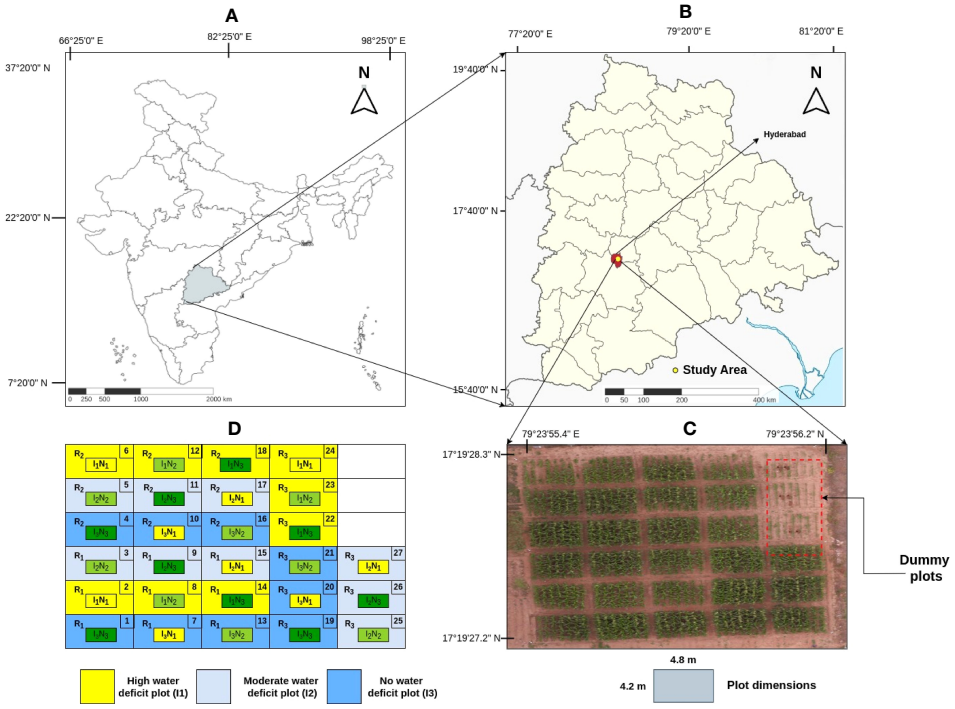


FIGURE 1
(A) Indian map. **(B)** Location of experiment field in Telangana map. **(C)** Top view of the field captured by the UAV with dummy plots highlighted. **(D)** Field layout of treatments where I_1, I_2, I_3 represent high, moderate and no water deficit plots respectively.

depicted in Figure 1C. In order to introduce diversity, each plot, that measures 4.2 m × 4.8 m, received one of three distinct combinations of water and nitrogen levels. This setup allowed for categorizing areas into conditions of low, moderate, and high water and fertilizer stress plots. In each plot within rows, the plants are spaced 20 cm apart from each other, and rows are spaced 60 cm apart for each treatment, resulting in an estimated plant density of 8.33 plants per square meter as shown in Figure 1D.

2.2 Dataset collection

To ensure an accurate geo-referenced data acquisition, we deployed nine Ground Control Points (GCPs) that are surveyed using a Trimble R10 GNSS Receiver within the field. The images are captured using a DJI Inspire-1 Pro UAV equipped with a Micasense RedEdge-MX multispectral camera included with a Downwelling Light Sensor (DLS) (represented in Supplementary Figure S1). This sensor is a 5-band light sensor that calculates the surrounding light conditions during a flight for each of the camera's five spectral bands and then stores this data within the metadata of the captured images. After calibration, this information is used to rectify the illumination changes in the middle of a flight that takes place due to cloud cover. Using Mission Planner version 4.3.1 (ArduPilot Dev team), the UAV flight path is predetermined at an altitude of 10 meters with a speed of 4 km/hr. The pixel resolution was set to 2 cm. Vertical overlap of 70-80% and horizontal overlap of 50-70% is maintained in consecutive images to ensure maximum coverage. The collected data consists of five spectral bands, blue (475 nm), green (560 nm), red (668 nm), red-edge (717 nm), and near-infrared (NIR) (842 nm) regions. In this study, crop cultivated from the tillering stage through the heading stage is considered. Radiometric calibration is carried out for the utilization of UAV-based multispectral imagery. It considers various factors, such as the position of the sensor and sun, camera gain, exposure information, and irradiance measurements that may affect the quality of image data. For radiometric calibration, images of the Calibrated Reflectance Panel (CRP) are captured by the camera and DL sensor before the UAV flight.

TABLE 1 Treatment information of the research farm for Rabi season (Winter 2018–19).

Treatment	Detail	Application Rate
I_1	High water stress	IW/CPE = 0.6
I_2	Moderate water stress	IW/CPE = 0.8
I_3	No water stress	IW/CPE = 1.2
N_1	High nitrogen stress	100 kg/ha
N_2	Optimum nitrogen	200 kg/ha
N_3	Overdose nitrogen	300 kg/ha

Here, IW means irrigated water in millimeter and CPE represents cumulative potential evaporation in mm. Nitrogen is supplied in kilogram per hectare (kg/ha).

2.3 Data pre-processing

Each CRP is associated with a calibration curve spanning the visible and NIR spectrum. Absolute reflectance values in the range

of 0 to 1 are related to the range of 400 - 850 nm (with a 1 nm increment). To perform radiometric calibration, the captured panel images are loaded with the above values provided by Micasense on Agisoft Metashape® Professional (Version 1.8.3 build 14331 64-bit) photogrammetry software. To obtain a complete field perspective, the raw photos are aligned, geo-rectified, and further stitched, based on similar image characteristics. After the alignment, the high-quality and mild filter mode options are used to create a dense point cloud. A Digital Elevation Model (DEM) and an orthomosaic (a panoramic picture stitched together and geometrically corrected) of each band, covered by the corresponding raw images, are exported (shown in Supplementary Figure S2A). The settings employed in the Agisoft Metashape software for the creation of orthomosaic are reported in Table 2. The shape files corresponding to orthomosaic are created using open source QGIS® tool, and using these files, subplot containing region of interest, are extracted using RStudio (shown in Supplementary Figure S2B). The net area is considered in the process to ensure that the impact of crops on the boundaries does not have any effect. This is obtained by removing 5% of the outer perimeter on each edge of the image. By performing the sliding window method on this extracted image, Region of Interest (ROI) of individual plants is extracted.

2.4 Methodology

Our proposed framework's workflow is illustrated in Figure 2, outlining all the steps undertaken in this study.

2.4.1 Overview of StressNet

Convolutional Neural Networks (CNNs) can be divided into two main components. The initial component, often referred to as the 'backbone,' comprises a series of convolutional and pooling layers aimed at extracting intricate features. These layers function as feature detectors, sampling the input image data to produce high-level feature maps. In simpler terms, specific neurons within these layers become active when certain features are detected in the input image. While the initial layers are proficient at capturing basic features like edges, the deeper layers excel at identifying more complex characteristics, such as textures and the shapes of specific objects. The second component, known as the 'head,' learns from the extracted features and produces results tailored to the specific application Zeiler and Fergus (2014).

As for the proposed model, StressNet, it comprises two key components. The first is a feature extraction module based on a CNN, while the second is a spatial-spectral-temporal feature fusion module using BiLSTM network and an attention mechanism. The feature extractor module captures spatial features across multiple spectral channels. These spatial-spectral and temporal features are then aggregated using the BiLSTM network and a weighted attention mechanism to achieve the final water stress classification. The architecture of the proposed model is depicted in Figure 3.

TABLE 2 The settings employed in the Agisoft Metashape software for the creation of orthomosaic.

Sparse point cloud	
Accuracy	Medium
Image pair selection	Ground control Point
Constrain features by mask	Exclude Stationary tie points
Maximum number of feature points	20,000
Dense point cloud	
Quality	Medium
Depth filtering	Mild
Digital Elevation Model(DEM)	
Type	Geographic
Coordinate system	WGS 84 (EPSG::4326)
Source data	Dense cloud
Orthomosaic	
Surface	DEM
Blending mode	Mosaic

2.4.2 Spatial-spectral feature extraction

The input for the feature extractor is in the form of k x k x c, where k x k represents the patch size and c denotes the number of channels. The final convolutional layer of the backbone network is replaced with a deformable convolutional layer. Deformable convolution is an extension of standard convolution that introduces additional parameters to control the sampling locations within the receptive field. Unlike the standard convolution, where the sampling grid is fixed, deformable convolution enables the network to learn spatial transformations and adapt its sampling locations dynamically Dai et al. (2017); Jin et al. (2019). The continuous increase in water stress leads to physiological changes in the crop, such as a decrease in the surface area of the leaf, which further leads to the twisting and rolling of the leaf Spišić et al. (2022). Deformable convolution enables the kernel to adjust its receptive field to the target size of the crop according to its growth stage and water stress condition with additional offsets. These offsets are updated during the training phase of the model Dai et al. (2017). Equation 1 is used for determining the output y at the location a_0, where x represents the input feature map, w stands for the learned weights, $a_i$specifies the i_{th}location and Δa_idenotes the offset to be learned.

$$y(a_0) = \sum w(a_i) \ * \ x(a_0 + a_i + \Delta a_i) \quad (1)$$

2.4.3 Spatial-spectral-temporal feature fusion

After extracting spatial and spectral features by deformable-based CNN, it is essential to capture the relationship between the temporal dependencies within the features. To achieve this, a BiLSTM network similar to that of Melamud et al. (2016) is employed. A BiLSTM layer is added to each feature extractor. The output of each feature extractor is given to the BiLSTM layer. Each BiLSTM is stacked with two LSTM layers, where the hidden state of the first LSTM is an input for the second LSTM, illustrated in Figure 4. By processing the sequential signals in reverse order, the

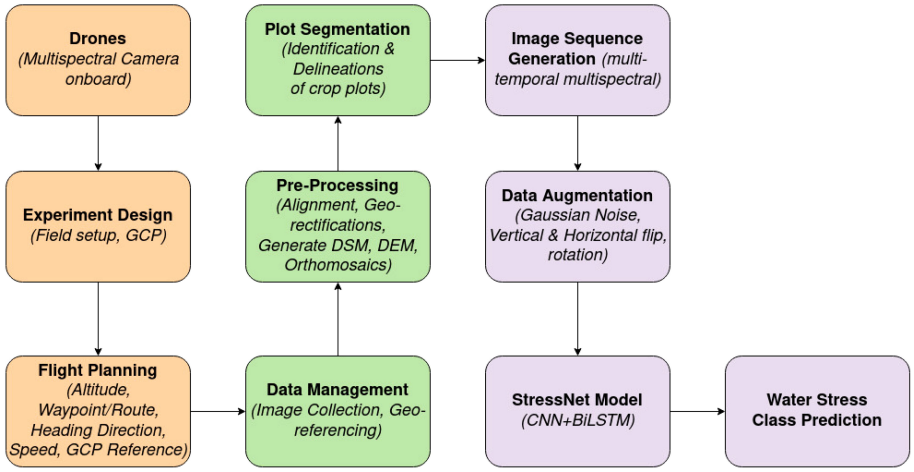


FIGURE 2
Our pipeline illustrates all the steps involved in water stress classification.

second LSTM layer enables a detailed understanding of the inter-dependencies within the data.

Equation 2 computes the input gate's output, determining how much of the new input shall be stored in the cell state c_t. On the other hand, Equation 3 corresponds to the forget gate f_t, which decides how much of the input $x_t$and previous cell state h_{t-1} is to be retained for the current time step. Further, Equation 4 updates the cell state $c_t$by removing some information based on the forget gate $f_t$and adding new information scaled by the input gate i_t. Equation 5 denotes the output gate that determines how much of the cell state's information should be passed to the hidden state. Finally, Equation 6 computes the new hidden state based on the cell state and the output gate's decision. In summary, these equations represent the working of an LSTM cell that helps the network learn and store information over longer sequences by controlling the flow through the cell state and hidden state using gates.

$$i_t = \sigma\ (W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma\ (W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t \tanh\ (W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{4}$$

$$o_t = \sigma\ (W_{ox}x_t + W_{oh}h_{t-1} + b_0) \tag{5}$$

$$h_t = o_t \tanh\ (c_t) \tag{6}$$

where, i refers to the input gate, f stands for the forget gate, o refers to the output gate, c is the memory cell and σ stands for the logistic sigmoid function.

To further improve the model's performance, a weighted attention layer is applied to the outcome of the second LSTM. By assigning varying degrees of importance to different input features, the attention layer dynamically adjusts the weights according to the input feature so that the model focuses on the most pertinent information. Consider H to be a matrix that contains the BiLSTM's output vectors $[h_1, h_2,\ldots, h_T]$, where T stands for the length of the

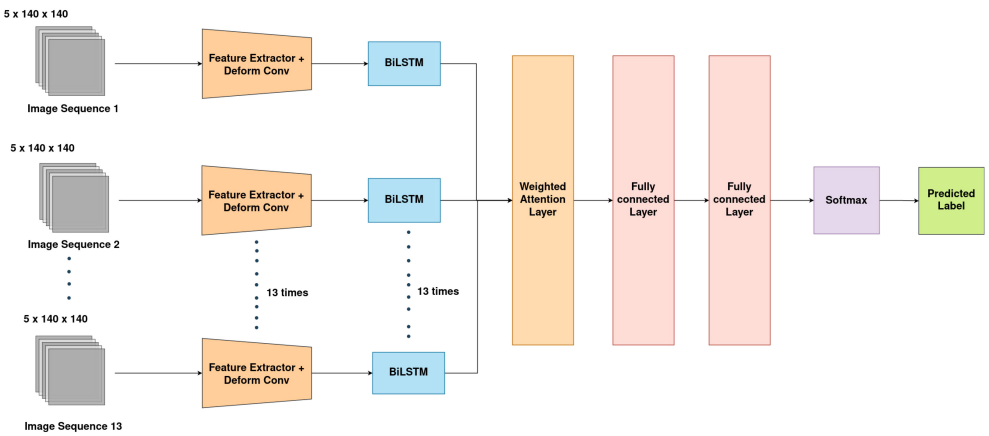


FIGURE 3
Overview of StressNet model. Input image sequence, Feature Extractor, Sequence processing BiLSTM network and Weighted attention modules are shown.

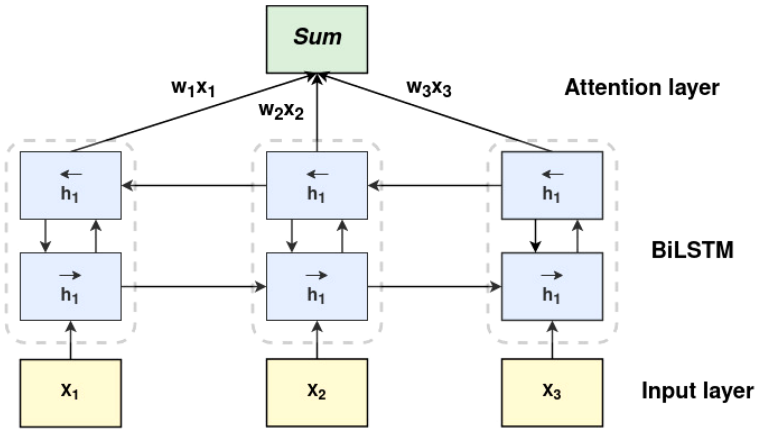


FIGURE 4
The architecture of the weighted attention-based bi-directional LSTM. x_1, x_2, x_3 correspond to features obtained by the feature extractor. h_1 typically refers to the hidden state output of the forward LSTM layer.

input features. The weighted sum of vectors adds up to the output of the attention layer and is described by the following equations 7, 8. The softmax activation function is a commonly used activation function in neural networks. It is used to transform the output of a neural network into a probability distribution. This transformation is defined by equation 9. Equation 10 refers to the 'combined' and 'attention-weighted' spatial-spectral-temporal features R_{att}, where α represents the attention vector. The BiLSTM-Attention features undergo an adaptive re-weighting or re-calibration, enhancing the significance of valuable feature vectors and diminishing the unwanted or noisy ones. Subsequently, these re-weighted features are connected to two fully connected layers and a softmax classifier. The output of the softmax classifier is a vector of probabilities where each element corresponds to the probability of the input belonging to a specific class.

$$M = \tanh(H) \tag{7}$$

$$\alpha = \text{softmax}\left(w^T M\right) \tag{8}$$

$$\text{where, softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1, \ldots, K \tag{9}$$

$$R_{\text{att}} = H\alpha^T \tag{10}$$

2.4.4 Data preparation

The training data is classified into three categories, namely, highly water-stressed, moderately waterstressed, and unaffected. Each class has 32 image sequences of 13 images of 5 channels. Each image has a dimension of 140 x 140 pixels. The Standard image normalization method is performed for all the channels by scaling all values to fit within the range of [0, 1] or adjusting the first- and second-order moments to achieve a mean of zero and a variance of one. All the channels of multispectral data are loaded into a sequence of the length of the days on which the data is captured using a custom data function. The ratio of training and validation is considered as 4:1.

2.4.5 Training details

Popular CNN-based models such as AlexNet Krizhevsky et al. (2012) and VGG-16 Simonyan and Zisserman (2014) architectures are employed as backbones of feature extractor. The first layer of CNN of the proposed model is modified to work with input of 5 channels instead of 3. Detailed configuration of the feature extractor with AlexNet and VGG-16 are shared in Tables 3, 4, respectively. During training, the model's weights are initialized using He initialization He et al. (2015), and biases are set to zero. The categorical cross-entropy loss function CE, represented in equation 11, is employed to train our model. This loss function considers the one-hot representation of the ground-truth label y, the predicted outcome y_p.

$$CE = -\sum_i y_i^p \log\ (y_i) \tag{11}$$

A batch size of 16 is utilized, and the Adam optimizer proposed by Kingma (2014) is employed with a learning rate of 1e-4. To address the limited data in the study, data augmentation technique is used. This involved rotating all training images by 90 degrees and randomly flipping them horizontally and vertically. The model is built using the PyTorch framework, and the training process is executed on a computer running on the Ubuntu 20.04 operating system. The training is implemented on Intel(R) Xeon(R) Platinum 8168 CPU with 24 cores and an NVIDIA Tesla V100-SXM3 Graphics Processing Unit (GPU) with 32 GB RAM.

2.4.6 Evaluation metrics

The assessment of the proposed model is conducted using the performance metrics that include Accuracy (Acc), Precision (Pre), and Sensitivity/Recall are defined in equations 12, 13, and 14 respectively. *FN* denotes False Negatives, *TN* corresponds to True Negatives, *TP* represents True Positives, and *FP* represents False Positives with respect to the actual and predicted water stress class.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{12}$$

TABLE 3 Detailed configuration of the feature extractor with AlexNet backbone.

Layer Name	Input Size (H x W x Channels)	Output size (H x W x Channels)	Kernel Size	Padding	Stride
Input	140 × 140 × 5	–	–	–	–
Conv1	140 × 140 × 5	– × – × 96	11 × 11	0	4
Conv2	– × – × 96	– × – × 256	5 × 5	2	1
Conv3	– × – × 256	– × – × 384	3 × 3	1	1
Conv4	– × – × 384	– × – × 384	3 × 3	1	1
Deform Conv Layer	– × – × 384	4 x 4 x 256	3 x 3	1	1

H,W denotes height and width of input respectively. Conv stands for Convolution. Deform Conv stands for Deformable convolutional layer.
x - is understood as the output size of feature map after convolution operation.

$$Precision = \frac{TP}{(TP + FP)} \tag{13}$$

$$Sensitivity/Recall = \frac{TP}{(TP + FN)} \tag{14}$$

3 Experiments and results

3.1 Results of the proposed model

We conducted spectral analysis and temporal analysis to highlight the efficiency of the proposed method. For the spectral analysis, we validated the model's performance by considering all 13 days' data of RGB channels or RGB with either NIR or red-edge channels. The results of spectral analysis are reported in Table 5. In the temporal analysis experiment, we assessed the model's performance by gradually adding the data from 3 to 13 days by utilizing all spectral channels. The results of the temporal analysis experiment are reported in Table 6. It is observed that the proposed model with VGG-16 backbone achieved the highest validation accuracy of 91.30%, a precision of 0.8888, and a sensitivity of 0.8857 when using all five spectral channels and data collected for up to 13 days. The class-level accuracies and the classification report of the best model are reported in Tables 7, 8, respectively. The training loss and validation accuracy graphs are represented in Figures 5A, B respectively.

3.2 Computational complexity

The best model (with the VGG-16 backbone) took 75 minutes to train for 100 epochs. The model consists of 14,060,611

TABLE 4 Detailed configuration of the feature extractor with VGG-16 backbone.

Layer Name	Input Size (H x W x Channels)	Output size (H x W x Channels)	Kernel Size	Padding	Stride
Input	140 × 140 × 5	–	–	–	–
Conv1	140 × 140 × 5	– × – × 64	3 × 3	1	1
Conv2	– × – × 64	– × – × 64	3 × 3	1	1
Conv3	– × – × 64	– × – × 128	3 × 3	1	1
Conv4	– × – × 128	– × – × 128	3 × 3	1	1
Conv5	– × – × 128	– × – × 256	3 × 3	1	1
Conv6	– × – × 256	– × – × 256	3 × 3	1	1
Conv7	– × – × 256	– × – × 256	3 × 3	1	1
Conv8	– × – × 256	– × – × 512	3 × 3	1	1
Conv9	– × – × 512	– × – × 512	3 × 3	1	1
Conv10	– × – × 512	– × – × 512	3 × 3	1	1
Conv11	– × – × 512	– × – × 512	3 × 3	1	1
Conv12	– × – × 512	– × – × 512	3 × 3	1	1
Deform Conv Layer	– × – × 512	4 x 4 x 512	3 x 3	1	1

H,W denotes height and width of input respectively. Conv stands for Convolution. Deform Conv stands for Deformable convolutional layer.

TABLE 5 Spectral analysis of StressNet model with AlexNet and VGG-16 backbones.

No. of Channels	AlexNet				VGG-16			
	Tr. Loss	Val. Acc.	Pre	Se	Tr. Loss	Val. Acc	Pre	Se
RGB	0.5521	73.913	0.5694	0.5206	0.5523	65.2174	0.7833	0.4777
RGB-NIR	0.5519	86.9565	0.7606	0.6793	0.5516	82.6087	0.7575	0.5936
RGB-Re	0.5516	73.913	0.6613	0.6682	0.5517	82.6087	0.6666	0.6349
All	0.5619	82.6087	0.7888	0.7888	0.5515	91.3043	0.8888	0.8857

(Tr. Loss, Training loss; Val. Acc., Validation Accuracy; Pre, Precision; Se, Sensitivity/Recall.).

parameters that include both trainable parameters (weights and biases) and non-trainable parameters. Considering that each parameter is stored as a 64-bit floating-point value, the estimated memory consumption of the proposed model is around 107.274 megabytes.

3.3 Ablation study

We performed an ablation study to assess the impact of temporal and spectral data on the proposed model's performance. This involved systematically reducing the number of temporal data used and spectral channels. Additionally, the study investigates the influence of the deformable convolution layer in comparison to standard convolution operation, along with the use of a BiLSTM network with weighted attention. These experiments aim to provide comprehensive evidence supporting the efficiency of our proposed method. The analysis includes the following cases.

1. Case I: Standard Convolution with BiLSTM.
2. Case II: Standard Convolution with BiLSTM and Weighted Attention.
3. Case III: Deformable Convolution with BiLSTM.

4 Discussion

For Spectral analysis, from Table 5, it can be inferred that our proposed model with AlexNet backbone achieves highest validation accuracy of 86.96% when using RGB-NIR channels as NIR band is good at highlighting the edges. With VGG-16 backbone, validation Accuracy is lowest of 65.22% when just using RGB bands. The addition of NIR and Re channels significantly increases accuracy and also with improvement in precision and sensitivity. The model's performance is highest when using all spectral channels. In summary, for AlexNet, the addition of NIR channels significantly improves performance, while for VGG-16, the inclusion of all channels, particularly RGB-NIR-Re, yields the highest performance. Both models benefit from the inclusion of multiple spectral channels, with VGG-16 (best model) showing higher overall accuracy and performance. In the temporal analysis, as shown in Table 6, our proposed model with the AlexNet backbone demonstrates strong performance with 3 and 6 days of data, achieving a high accuracy of 95.65%. Although there is a slight decrease in precision, sensitivity improves. However, when the number of temporal data increases, the model's performance drops to 82.60%, accompanied by a notable decrease in precision and sensitivity. On the other hand, our proposed model with the VGG-16 backbone exhibits a gradual increase in validation accuracy, going from 86.95% with 3 days of data to 95.65% with 9 days' data. However, there is a performance decrease when using 11 days of data. Notably, the model performs exceptionally well with 13 days of data, achieving a validation accuracy of 91.30% along with improved precision and recall. It's worth highlighting that this model achieves 95.65% validation accuracy using only 6 days of data, indicating the potential for early identification of water-stressed crops.

From Figure 6A, it is evident that the performance of the best model (StressNet with VGG-16 backbone) gradually improves with the addition of NIR and Re spectral bands alongside RGB bands, signifying that incorporating both red-edge and NIR channels

TABLE 6 Temporal Analysis of StressNet model with AlexNet and VGG-16 backbones, where N represent images of dataset of N days.

N	AlexNet				VGG-16			
	Tr. Loss	Val. Acc.	Pre	Se	Tr. Loss	Val. Acc	Pre	Se
3	0.5523	95.6522	0.9111	0.9111	0.5517	86.9525	0.8055	0.7603
6	0.5517	95.6522	0.9107	0.9333	0.5660	95.6522	0.8555	0.8079
9	0.5519	82.6087	0.8498	0.7523	0.5516	95.6522	0.8484	0.7904
11	0.5517	82.6087	0.8296	0.7746	0.6051	73.913	0.5726	0.5587
13	0.5619	82.6087	0.7888	0.7888	0.5515	91.3043	0.8888	0.8857

(Tr. Loss, Training loss; Val. Acc., Validation Accuracy; Pre, Precision; Se, Sensitivity/Recall).

TABLE 7 Class-level accuracy of the best StressNet model.

Class Name	Class Label	Accuracy Score
I1N2	0	0.900
I2N2	1	1.000
I3N2	2	0.833

enhances the model's capability. Figure 6B illustrates a progressive increase in the model's performance up to 9 days. Subsequently, there is a decrease in performance between days 9 and 11, followed by an increase again.

4.1 Spectral analysis

In the spectral analysis conducted as part of the ablation study, three experiments were considered: RGB, RGB+NIR, RGB+Re, and all bands (as shown in Table 9). In Case I, the VGG-16 model achieved the highest test accuracy of 95.65% using RGB and red-edge data, highlighting the significance of spectral information for model robustness. In Case II, the VGG-16 model achieved the highest test accuracy of 95.65% when using all spectral bands. In Case III, the AlexNet model achieved the highest accuracy of 91.30% with RGB and red-edge information. Notably, the model a precision of 0.9027 (as shown in Case I) with standard convolution using RGB and Re bands. In Case II, with standard convolution and the integration of the BiLSTM network and weighted attention, the VGG-16 backbone model achieved a precision of 0.8727. In Case III, when using deformable convolutional layer with BiLSTM and weighted attention, along with AlexNet as the backbone, the model achieved a precision of 0.9047 with RGB and red-edge information. However, in cases where VGG-16 served as the backbone, the NIR and Re bands introduced essential features, leading the deformable convolutional layer to capture redundant spatial feature vectors and ultimately resulting in a reduction in accuracy compared to RGB data.

4.2 Temporal analysis

In addition to spectral analysis, we conducted a temporal study, exploring various temporal windows ranging from 3 to 13 days (as shown in Table 10). In Case I, AlexNet model achieved the highest validation accuracy of 91.30% with three days of data. In Case II, VGG-16 model achieved the highest validation accuracy of 95.65% with nine days of data. In Case III, AlexNet model achieved the highest validation accuracy of 95.65% with six days of data. By introducing a deformable convolutional layer with six days of data, the accuracy increased to 95% from the 90% observed in Case I (Feature extractor + BiLSTM). In contrast, VGG-16 extracted more refined features were with nine days of data, capturing distinct water stress patterns. However, after that point, there was minimal change in accuracy. The test accuracy reached 95%, underscoring the significance of incorporating a weighted attention module. Nevertheless, the test accuracy dropped from 95% to 65% with the addition of deformable convolution, indicating that the deformable convolutional layer introduced unnecessary complexity and increased parameters, leading to overfitting.

4.3 Impact of deformable convolution

To assess the impact of deformable convolution, we examined Cases II and III in the ablation study (Tables 9, 10). In the spectral analysis experiment, the AlexNet model's performance increased from 56.52% validation accuracy to 78.26% with RGB bands. However, there was no change with RGB-NIR. Notably, with RGB-Re bands, the AlexNet model's accuracy surged to 91.30%. For the VGG-16 model, adding the deformable convolutional layer with RGB bands raised the validation accuracy to 82.60% from 43.47%. However, introducing additional spectral channels led to a 10-20% drop in validation accuracy, likely due to increased model complexity, overfitting, and feature redundancy. Regarding temporal analysis, the AlexNet model achieved its highest validation accuracy of 90% with 6 days' data. The model's performance gradually declined as the number of days increased. In contrast, the VGG-16 model's performance was more variable, reaching a peak of 82.60% (as shown in Case III). This suggests that deformable convolution enhances the extraction of spatial features, resulting in a richer vector representation across timestamps. As data increased from 3 to 6 days, the model's performance exhibited a decreasing trend, suggesting a potential absence of identified geometrical transformations. The introduction of the deformable convolution layer added unnecessary complexity and increased the number of parameters, resulting in overfitting.

4.4 Impact of weighted attention based BiLSTM

To assess the impact of deformable convolution, we investigated Cases I and II in the ablation study (Tables 9, 10). In the spectral analysis experiment, the AlexNet model achieved an impressive

TABLE 8 Classification report of the best StressNet model.

Class	Precision	Recall	F1-Score	Support
0	0.90	0.90	0.90	20
1	1.00	1.00	1.00	14
2	0.83	0.83	0.83	12

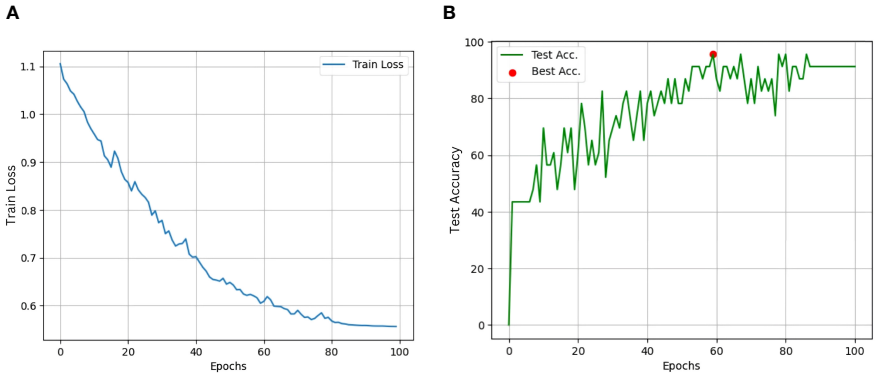


FIGURE 5
(A) Training loss of best StressNet model; **(B)** Validation Accuracy of best StressNet model.

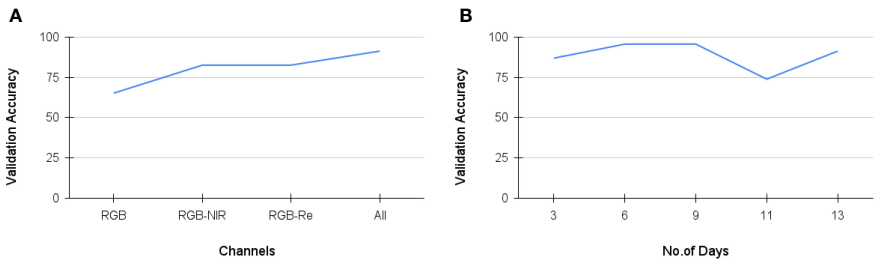


FIGURE 6
(A) Validation accuracy of best StressNet model with respect to Spectral Analysis; **(B)** Validation accuracy of best StressNet model with respect to Temporal Analysis.

TABLE 9 Spectral Analysis. Case-I: Feature Extractor with BiLSTM network, Case-II: Feature Extractor with BiLSTM network and Weighted Attention, Case-III: Feature Extractor with Deformable Convolution and BiLSTM network.

Case	Feature Extractor	Metric	RGB	RGB-NIR	RGB-Re	All
Case - I	**AlexNet**	Tr. Loss	0.5551	0.5543	0.5546	0.5534
		Val. Acc.	91.3043	91.3043	82.6087	86.9565
		Pre	0.9	0.9444	0.62	0.83
		Se	0.79	0.8968	0.56	0.8
	VGG - 16	Tr. Loss	0.883	0.5536	0.5785	0.5729
		Val. Acc.	82.6087	65.2174	**95.6522**	86.9565
		Pre	0.5087	0.6809	0.9027	0.856
		Se	0.5238	0.6015	0.8333	0.8238
Case - II	**AlexNet**	Tr. Loss	0.5527	0.5522	0.5532	0.5525
		Val. Acc.	56.5217	82.6087	56.5217	78.2609
		Pre	0.4583	0.7269	0.4814	0.7416
		Se	0.4539	0.7269	0.466	0.7349
	VGG - 16	Tr. Loss	1.0693	0.5717	0.562	0.5627
		Val. Acc.	43.4783	91.3043	82.6087	**95.6522**
		Pre	0.1449	0.8727	0.7051	0.787

(Continued)

TABLE 9 Continued

Case	Feature Extractor	Metric	RGB	RGB-NIR	RGB-Re	All
		Se	0.3333	0.8555	0.6634	0.7968
Case - III	**AlexNet**	Tr. Loss	0.5537	0.5534	0.5533	0.5517
		Val. Acc.	78.2609	82.6087	**91.3043**	78.2609
		Pre	0.6428	0.8214	0.9047	0.8333
		Se	0.6079	0.738	0.7936	0.7666
	VGG - 16	Tr. Loss	0.5515	0.6397	0.5877	0.6125
		Val. Acc.	82.6087	78.2609	69.5652	78.2609
		Pre	0.7306	0.7348	0.5958	0.6888
		Se	0.7111	0.6873	0.5539	0.673

Tr. Loss, Training Loss; Val. Acc., Validation Accuracy; Pre, Precision; Se, Sensitivity.

TABLE 10 Temporal Analysis. Case-I: Feature Extractor with BiLSTM network, Case-II: Feature Extractor with BiLSTM network and Weighted Attention, Case-III: Feature Extractor with Deformable Convolution and BiLSTM network.

Case	Feature Extractor	Metric	3	6	9	11	13
Case - I	AlexNet	Tr. Loss	0.5548	0.5746	0.5541	0.5532	0.5534
		Val. Acc.	**91.3043**	86.9565	86.9565	73.913	86.9565
		Pre	0.8714	0.8517	0.8634	0.7724	0.83
		Se	0.8634	0.8634	0.8634	0.6492	0.81
	VGG - 16	Tr. Loss	0.5619	1.0689	0.5625	1.069	1.069
		Val. Acc.	65.2174	56.5217	82.6087	43.4783	43.4783
		Pre	0.5444	0.1449	0.7571	0.1449	0.1449
		Se	0.5698	0.3333	0.7412	0.3333	0.3333
Case - II	AlexNet	Tr. Loss	0.5533	0.5535	0.5524	0.5539	0.5527
		Val. Acc.	78.2609	90	82.6087	65.2174	56.5217
		Pre	0.7248	0.9696	0.7471	0.7361	0.4583
		Se	0.7269	0.9444	0.7269	0.5079	0.4539
	VGG - 16	Tr. Loss	0.5724	0.5572	0.552	0.552	0.5621
		Val. Acc.	73.913	86.9565	**95.6522**	95.6522	95.6522
		Pre	0.3552	0.7962	0.863	0.7833	0.6974
		Se	0.4523	0.5222	0.8777	0.7761	0.6571
Case - III	AlexNet	Tr. Loss	0.5529	0.5325	0.5529	0.5531	0.5517
		Val. Acc.	82.6087	**95.6522**	86.9565	82.6087	78.2609
		Pre	0.7458	0.9696	0.744	0.75	0.8333
		Se	0.6222	0.9444	0.7555	0.7555	0.7666
	VGG - 16	Tr. Loss	0.7858	0.5954	0.5795	1.0695	0.5515
		Val. Acc.	78.2609	78.2609	65.2174	43.4783	82.6087
		Pre	0.6388	0.7727	0.3789	0.1449	0.7306
		Se	0.6253	0.6492	0.4904	0.3333	0.7111

Tr. Loss, Training Loss; Val. Acc., Validation Accuracy; Pre, Precision; Se, Sensitivity.
The bold values highlighted highest validation accuracies obtained in that specific case.

91.30% validation accuracy. However, there was no significant improvement in performance when either NIR or Re channels were added. This limitation can be attributed to complex background variations in the data, which challenged the limited feature representation capacity of the AlexNet model, making it challenging to distinguish foreground information. In contrast, the VGG-16 model, with its deeper layers and the support of the BiLSTM network and weighted attention mechanism, effectively addressed complex backgrounds, resulting in a substantial performance increase from 86.95% to 95.65%. In the context of temporal analysis, the performance of the AlexNet model exhibited an initial increase, followed by a subsequent decrease as the data extended from 3 days to 9 days (as demonstrated in Case I). Beyond the 9th day, this pattern persisted. A similar trend was observed after introducing weighted attention (Case II). In contrast, the VGG-16 model demonstrated higher performance in both Case I and II up to 9 days, indicating the model's resilience in managing temporal variations in images corresponding to the crop's growth over time. Beyond this point, the performance remained relatively constant with 11 and 13 days' data, suggesting negligible growth in the crops.

5 Conclusion

In this article, we propose a novel DL-based model titled StressNet, which aims to monitor water stress, especially in maize crop. StressNet consists of two key components, the first being CNN with a deformable convolutional layer, and the second is a BiLSTM network with weighted attention. The effectiveness of our framework is extensively validated through a comprehensive study utilizing multispectral and multi-temporal imagery captured by UAV. The best model achieved a validation accuracy of 91.30% with a training loss of 0.555. However, it is essential to acknowledge that our proposed method is validated using a dataset acquired from a controlled environment. However, the real-world scenario introduces more complexities. In such circumstances, it is essential to consider additional factors such as super-resolution, noise reduction, and plant shoot segmentation techniques. We will develop a DL pipeline with further additions in our future research. We encourage researchers to verify our findings using their datasets and expand upon our pipeline.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

TN, KK, and SC worked on the conceptualization and methodology of the paper; TN worked on data curation, generation and pre-processing. TN designed the experiments. KK and SC developed the code for the experiments. TN and KK conducted experiments and validated. TN wrote the original manuscript. TN, KK, and SC analyzed the findings and suggested the modifications in the manuscript. RP reviewed and supervised the work. RP, BN, and UD provided funding and resources for experimental site setup and data collection. All authors contributed to the article and approved the submitted version.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by Department of Science and Technology (DST) India and Japan Science and Technology (JST) Japan under the project "Data Science-Based Farming Support System For Sustainable Crop Production Under Climatic Change (DSFS)" project number: MST/IBCD/EE/F066/2016-17G48.

Acknowledgments

We would like to acknowledge Ajay Kumar, Mahesh (contributed when they were pursuing PhD and M.Tech in IIT Hyderabad respectively) and Naresh, Research Staff in WiNet lab for their support in flying the UAV and data collection, Praneela, intern in WiNet Lab, for helping with the pictorial illustrations and creating tables in latex. We would also thank reviewers for their valuable and constructive feedback to improve the quality of our work.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1241921/full#supplementary-material

References

Al-Tamimi, N., Langan, P., Bernád, V., Walsh, J., Mangina, E., and Negrão, S. (2022). Capturing crop adaptation to abiotic stress using image-based technologies. *Open Biol.* 12, 210353. doi: 10.1098/rsob.210353

Araus, J. L., and Cairns, J. E. (2014). Field high-throughput phenotyping: the new crop breeding frontier. *Trends Plant Sci.* 19, 52–61. doi: 10.1016/j.tplants.2013.09.008

Azimi, S., Wadhawan, R., and Gandhi, T. K. (2021). Intelligent monitoring of stress induced by water deficiency in plants using deep learning. *IEEE Trans. Instrumentation Measurement* 70, 1–13. doi: 10.1109/TIM.2021.3111994

Barradas, A., Correia, P. M., Silva, S., Mariano, P., Pires, M. C., Matos, A. R., et al. (2021). Comparing machine learning methods for classifying plant drought stress from leaf reflectance spectra in arabidopsis thaliana. *Appl. Sci.* 11, 6392. doi: 10.3390/app11146392

Berni, J. A., Zarco-Tejada, P. J., Suárez, L., and Fereres, E. (2009). Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Trans. Geosci. Remote Sens.* 47, 722–738. doi: 10.1109/TGRS.2008.2010457

Bouguettaya, A., Zarzour, H., Kechida, A., and Taberkit, A. M. (2022). Deep learning techniques to classify agricultural crops through uav imagery: A review. *Neural Computing Appl.* 34, 9511–9536. doi: 10.1007/s00521-022-07104-9

Boyer, J. S. (1982). Plant productivity and environment. *Science* 218, 443–448. doi: 10.1126/science.218.4571.443

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al. (2017). "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*. Venice. 764–773.

Dataset IIMR (2020). *Iimr annual report*.

Elsherbiny, O., Zhou, L., He, Y., and Qiu, Z. (2022). A novel hybrid deep network for diagnosing water status in wheat crop using iot-based multimodal data. *Comput. Electron. Agric.* 203, 107453. doi: 10.1016/j.compag.2022.107453

Feng, Q., Yang, J., Liu, Y., Ou, C., Zhu, D., Niu, B., et al. (2020). Multi-temporal unmanned aerial vehicle remote sensing for vegetable mapping using an attention-based recurrent convolutional neural network. *Remote Sens.* 12 1668. doi: 10.3390/rs12101668

Food and of the United Nations, A. O (2019). *Agriculture and climate change: Challenges and opportunities at the global and local level: Collaboration on climate-smart agriculture* (Food and Agriculture Organization of the United Nations).

Gago, J., Douthe, C., Coopman, R. E., Gallego, P. P., Ribas-Carbo, M., Flexas, J., et al. (2015). Uavs challenge to assess water stress for sustainable agriculture. *Agric. Water Manage.* 153, 9–19. doi: 10.1016/j.agwat.2015.01.020

Gao, Z., Luo, Z., Zhang, W., Lv, Z., and Xu, Y. (2020). Deep learning application in plant stress imaging: a review. *AgriEngineering* 2, 29. doi: 10.3390/agriengineering2030029

Gonzalez-Dugo, V., Durand, J.-L., and Gastal, F. (2010). Water deficit and nitrogen nutrition of crops. a review. *Agron. Sustain. Dev.* 30, 529–544. doi: 10.1051/agro/2009059

Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* 127, 418–424. doi: 10.1016/j.compag.2016.07.003

Halagalimath, S., et al. (2017). Effect of scheduling irrigation and mulching on growth and yield of maize (zea mays l.). *J. Farm Sci.* 30, 45–48.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*. Araucano Park. 1026–1034.

Jin, Q., Meng, Z., Pham, T. D., Chen, Q., Wei, L., and Su, R. (2019). Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Syst.* 178, 149–162. doi: 10.1016/j.knosys.2019.04.025

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.* 147, 70–90. doi: 10.1016/j.compag.2018.02.016

Kingma, D. P. (2014). A method for stochastic optimization. *ArXiv Prepr.* doi: 10.48550/arXiv.1412.6980

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60 (6), 84–90. AcM New York, NY, USA. doi: 10.1145/3065386

Kumar, A., Shreeshan, S., Tejasri, N., Rajalakshmi, P., Guo, W., Naik, B., et al. (2020). "Identification of water-stressed area in maize crop using uav based remote sensing," in *2020 IEEE India geoscience and remote sensing symposium (InGARSS)*. 146–149 (IEEE).

Laborde, D., Martin, W., Swinnen, J., and Vos, R. (2020). Covid-19 risks to global food security. *Science* 369, 500–502. doi: 10.1126/science.abc4765

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lee, S. H., Chan, C. S., Mayo, S. J., and Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern recognition* 71, 1–13. doi: 10.1016/j.patcog.2017.05.015

Li, D., Li, C., Yao, Y., Li, M., and Liu, L. (2020). Modern imaging techniques in plant nutrition analysis: A review. *Comput. Electron. Agric.* 174, 105459. doi: 10.1016/j.compag.2020.105459

Liu, C., Li, H., Su, A., Chen, S., and Li, W. (2020). Identification and grading of maize drought on rgb images of uav based on improved u-net. *IEEE Geosci. Remote Sens. Lett.* 18 (2), 198–202. doi: 10.1109/LGRS.2020.2972313

Melamud, O., Goldberger, J., and Dagan, I. (2016). "context2vec: Learning generic context embedding with bidirectional lstm," in *Proceedings of the 20th SIGNLL conference on computational natural language learning*. Berlin, Germany. 51–61.

Mueller, N. D., Gerber, J. S., Johnston, M., Ray, D. K., Ramankutty, N., and Foley, J. A. (2012). Closing yield gaps through nutrient and water management. *Nature* 490, 254–257. doi: 10.1038/nature11420

Nijland, W., De Jong, R., De Jong, S. M., Wulder, M. A., Bater, C. W., and Coops, N. C. (2014). Monitoring plant condition and phenology using infrared sensitive consumer grade digital cameras. *Agric. For. Meteorology* 184, 98–106. doi: 10.1016/j.agrformet.2013.09.007

Reddy, T. Y., and Reddy, G. (2019). *Principles of agronomy* (Kalyani publishers).

Semmens, K. A., Anderson, M. C., Kustas, W. P., Gao, F., Alfieri, J. G., McKee, L., et al. (2016). Monitoring daily evapotranspiration over two california vineyards using landsat 8 in a multi-sensor data fusion approach. *Remote Sens. Environ.* 185, 155–170. doi: 10.1016/j.rse.2015.10.025

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*. doi: 10.48550/arXiv.1409.1556

Singh, A. K., Ganapathysubramanian, B., Sarkar, S., and Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. *Trends Plant Sci.* 23, 883–898. doi: 10.1016/j.tplants.2018.07.004

Spišić, J., Šimić, D., Balen, J., Jambrović, A., and Galić, V. (2022). Machine learning in the analysis of multispectral reads in maize canopies responding to increased temperatures and water deficit. *Remote Sens.* 14, 2596. doi: 10.3390/rs14112596

Tejasri, N., Pachamuthu, R., Naik, B., and Desai, U. B. (2023). "Intelligent drought stress monitoring on spatio-spectral-temporal drone based crop imagery using deep networks," in *2nd AAAI Workshop on AI for Agriculture and Food Systems*. Washington, D.C., USA.

Tejasri, N., Rajalakshmi, P., Naik, B., Desai, U. B., et al. (2022). "Drought stress segmentation on drone captured maize using ensemble u-net framework," in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*. 1–6 (Genova, Italy: IEEE).

Thorp, K. R., Thompson, A. L., Harders, S. J., French, A. N., and Ward, R. W. (2018). High-throughput phenotyping of crop water use efficiency via multispectral drone imagery and a daily soil water balance model. *Remote Sens.* 10, 1682. doi: 10.3390/rs10111682

Tian, H., Wang, T., Liu, Y., Qiao, X., and Li, Y. (2020). Computer vision technology in agricultural automation—a review. *Inf. Process. Agric.* 7, 1–19. Elsevier. doi: 10.1016/j.inpa.2019.09.006

Vicente, R., Vergara-Díaz, O., Medina, S., Chairi, F., Kefauver, S. C., Bort, J., et al. (2018). Durum wheat ears perform better than the flag leaves under water stress: gene expression and physiological evidence. *Environ. Exp. Bot.* 153, 271–285. doi: 10.1016/j.envexpbot.2018.06.004

Virnodkar, S. S., Pachghare, V. K., Patil, V., and Jha, S. K. (2020). Remote sensing and machine learning for crop water stress determination in various crops: a critical review. *Precis. Agric.* 21, 1121–1155. doi: 10.1007/s11119-020-09711-9

Wang, D., Cao, W., Zhang, F., Li, Z., Xu, S., and Wu, X. (2022). A review of deep learning in multiscale agricultural sensing. *Remote Sens.* 14, 559. doi: 10.3390/rs14030559

Wang, X., and Xing, Y. (2016). Effects of irrigation and nitrogen fertilizer input levels on soil-n content and vertical distribution in greenhouse tomato (lycopersicum esculentum mill.). *Scientifica* 2016. Hindawi. doi: 10.1155/2016/5710915

Zarco-Tejada, P. J., González-Dugo, V., and Berni, J. A. (2012). Fluorescence, temperature and narrowband indices acquired from a uav platform for water stress detection using a micro-hyperspectral imager and a thermal camera. *Remote Sens. Environ.* 117, 322–337. doi: 10.1016/j.rse.2011.10.007

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. 818–833 (Springer).

Zhong, L., Hu, L., and Zhou, H. (2019). Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* 221, 430–443. doi: 10.1016/j.rse.2018.11.032

Zhou, L., Gu, X., Cheng, S., Guijun, Y., Shu, M., and Sun, Q. (2020). Analysis of plant height changes of lodged maize using uav-lidar data. *Agriculture* 10, 146. doi: 10.3390/agriculture10050146

Zhu, J., Fang, L., and Ghamisi, P. (2018). Deformable convolutional neural networks for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 15, 1254–1258. doi: 10.1109/LGRS.2018.2830403

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 08 December 2023
DOI 10.3389/fpls.2023.1308528

OPEN ACCESS

EDITED BY
José Dias Pereira,
Instituto Politecnico de Setubal (IPS), Portugal

REVIEWED BY
Vítor Viegas,
Naval School, Portugal
Jakub Nalepa,
Silesian University of Technology, Poland

*CORRESPONDENCE
Signe Marie Jensen
smj@plen.ku.dk

RECEIVED 06 October 2023
ACCEPTED 22 November 2023
PUBLISHED 08 December 2023

CITATION
Khan AT, Jensen SM, Khan AR and Li S (2023) Plant disease detection model for edge computing devices.
Front. Plant Sci. 14:1308528.
doi: 10.3389/fpls.2023.1308528

Plant disease detection model for edge computing devices

Ameer Tamoor Khan[1], Signe Marie Jensen[1]*, Abdul Rehman Khan[2] and Shuai Li[3]

[1]Department of Plant and Environmental Science, University of Copenhagen, Copenhagen, Denmark, [2]Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan, [3]Deparment of Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland

In this paper, we address the question of achieving high accuracy in deep learning models for agricultural applications through edge computing devices while considering the associated resource constraints. Traditional and state-of-the-art models have demonstrated good accuracy, but their practicality as end-user available solutions remains uncertain due to current resource limitations. One agricultural application for deep learning models is the detection and classification of plant diseases through image-based crop monitoring. We used the publicly available PlantVillage dataset containing images of healthy and diseased leaves for 14 crop species and 6 groups of diseases as example data. The MobileNetV3-small model succeeds in classifying the leaves with a test accuracy of around 99.50%. Post-training optimization using quantization reduced the number of model parameters from approximately 1.5 million to 0.93 million while maintaining the accuracy of 99.50%. The final model is in ONNX format, enabling deployment across various platforms, including mobile devices. These findings offer a cost-effective solution for deploying accurate deep-learning models in agricultural applications.

KEYWORDS

PlantVillage, deep learning, classifier, edge computing, MobileNetV3

1 Introduction

Plant diseases can be a major concern for farmers due to the risk of substantial yield loss. While applying pesticides can prevent or limit the impact of most plant diseases, their use should be restricted due to environmental considerations. Early and efficient detection of plant diseases and their distribution in the field is crucial for effective treatment. The implementation of automatic plant disease detection systems is, therefore, essential for efficient crop monitoring. Deep Learning Convolutional Neural Networks (CNNs) and computer vision are two developing AI technologies that have recently been employed to identify plant leaf diseases automatically.

Already in 1980, Fukushima (1980) presented a visual cortex-inspired multilayer artificial neural network for image classification. The network showed that the initial layer detects simpler patterns with a narrow receptive field, while later levels combine patterns from earlier layers to identify more complex patterns with wider fields. In 2012,

Krizhevsky et al. (2012) developed the AlexNet architecture, which helped them win the ImageNet Large Scale Visual Recognition Challenge. Several CNN (Convolutional Neural Network) designs have been introduced since then Krizhevsky et al. (2012); Fu et al. (2018); Yang et al. (2023); Dutta et al. (2016); Sarda et al. (2021). These models are called "deep learning" architectures due to their 5-200 layers. Early investigations employed manually created characteristics from leaf picture samples. Later, the trends shifted to DCNN (Deep Convolutional Neural Network) architectures capable of effectively classifying data and automatically extracting features. Plant disease picture classification has been used to test a variety of CNN architectures Amara et al. (2017); Sladojevic et al. (2016); Setiawan et al. (2021); Yang et al. (2023); Qiang et al. (2019); Swaminathan et al. (2021); Schuler et al. (2022).

Plant disease diagnosis through image analysis employs various machine learning techniques Ferentinos (2018). These methods identify and classify diseases affecting cucumbers, bananas Fujita et al. (2016), cassavas Amara et al. (2017), tomatoes Ramcharan et al. (2017), and wheat Fuentes et al. (2017). Ramcharan et al. (2017) tested five architectures—AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat, and VGG on 58 classes of healthy and sick plants. AlexNet achieved 99.06% and VGG 99.48% test accuracy. Despite the large variation in trainable parameters, these designs had test accuracy above 99%. Maeda-Gutiérrez et al. (2020) tested five architectures for tomato illnesses. All architectures tested had accuracies above 99%. However, when tested on field pictures, Ramcharan et al. (2017) encountered shadowing and leaf misalignment. These factors greatly affected classification accuracy.

Amara et al. (2017) classified banana leaf diseases using 60×60 pixel pictures and a simple LeNet architecture. Grayscale images had 85.94%, and RGB images had 92.88% test accuracy. Chromatic information Mohanty et al. (2016) is essential in plant leaf disease classification. Mohanty et al. (2016) used AlexNet and GoogLeNet (Inception V1) designs to study plant leaf diseases and found RGB images to be more accurate than their grayscale counterparts. Likewise, Schuler et al. (2022) split the Inception V3 architecture into two branches, one dealing with the grayscale part of the RGB image and the other branch dealing with the other two channels of the RGB image. The resultant architecture has 5 million trainable parameters and achieved an accuracy of 99.48% on the test dataset.

While these studies demonstrate the effectiveness of deep learning in plant disease classification, they often do not address the critical challenge of deploying these models on resource-constrained edge devices. In contrast, our work not only achieves high accuracy but also emphasizes optimizing deep learning models for such constraints. Recent advancements in the field substantiate this focus. For instance, Hao et al. (2023) discusses system techniques that enhance DL inference throughput on edge devices, a key consideration for real-time applications in agriculture. Similarly, the DeepEdgeSoc framework Al Koutayni et al. (2023) accelerates DL network design for energy-efficient FPGA implementations, aligning with our resource efficiency goal. Moreover, approaches like resource-frugal quantized CNNs Nalepa et al. (2020) and knowledge distillation methods Alabbasy et al. (2023) resonate with our efforts to compress model size while maintaining performance. These studies highlight the importance of balancing computational demands with resource limitations, a core aspect of our research. Thus, our work stands out by not only addressing the accuracy of plant disease detection but also ensuring the practical deployment of these models in real-world agricultural settings where resources are limited.

One major drawback in the broader field is that deep-learning approaches often have computational requirements, *i.e.*, higher memory and computing capacity, which are not always feasible for edge computing devices. Our paper tackles this challenge head-on, focusing on maximizing accuracy while operating within the resource constraints inherent to edge computing devices, thereby significantly enhancing the real-life applicability of deep learning models in agriculture.

The remaining part of the paper is organized as follows: Section 2 will look into the PlantVillage dataset, then we will explore the MobileNetV3-small architecture, model training, and finally, the post-training quantization. Section 3 will discuss the results and the comparison with existing methods. In Section 4, we will discuss the importance of the problem and the relevance of our results. Finally, Section 5 will conclude the paper with final remarks.

2 Materials and methods

2.1 PlantVillage dataset

The present work used the publicly available PlantVillage-Dataset (2016). All images in the PlantVillage database were captured at experimental research facilities connected to American Land Grant Universities. The dataset included 54,309 images of 14 crop species, including tomato, apple, bell pepper, potato, raspberry, soybean, squash, strawberry, and grape. A few sample images of the plants are shown in Figure 1. It could be seen that some samples were healthy,

FIGURE 1
Sample images of the PlantVillage dataset. It is a diverse dataset with 14 plant species, including healthy and infected plants. The dataset includes a total of 54,309 image samples.

and some were infected. There were 17 fungal infections, 4 bacterial diseases, 2 viral diseases, 1 mite disease, and 1 mold (oomycete). There were images of healthy leaves from 12 crop species, showing no obvious signs of disease. In total, the dataset included 38 classes of healthy and unhealthy crops. A detailed description of the distribution of species and diseases in the dataset is shown in Table 1. It included 14 crop species with 6 types, *i.e.*, fungi, bacteria, mold, virus, mite, and healthy. The dataset is imbalanced and not equally distributed across all 6 types.

To further elaborate on the imbalanced nature of the dataset, t-SNE analysis was performed. t-SNE, or tDistributed Stochastic Neighbor Embedding, is a machine learning technique used to reduce dimensionality and visualize high-dimensional data. It attempts to represent complex, high-dimensional data in a lowerdimensional space while maintaining data point relationships. The data overlapping is quite visible in Figure 2, where the dimensions of the PlantVillage dataset were reduced to 2.

2.2 MobileNetV3-small

Recent research has focused on deep neural network topologies that balance accuracy and efficiency. Innovative handcrafted structures and algorithmic neural architecture search have advanced this discipline.

SqueezeNet used 1×1 convolutions with squeeze-and-expand modules to reduce parameters Iandola et al. (2016). Recent research has focused on minimizing MAdds (Million Additions) and latency instead of parameters. Depthwise separable convolutions boosted

TABLE 1 Distribution of observations in the PlantVillage dataset.

	Fungi	Bacteria	Mold	Virus	Mite	Healthy
Apple (**3172**)	1521					1645
Blueberry (**1502**)						1502
Bell Pepper (**2475**)		997				1478
Cherry (**1906**)	1052					854
Corn (**3852**)	2690					1162
Grape (**4063**)	3640					423
Orange (**5507**)		5507				
Peach (**2657**)		2291				360
Potato (**2152**)	1000		1000			152
Raspberry (**371**)						371
Soybean (**5090**)						5090
Squash (**1835**)	1835					
Strawberry (**1565**)	1109					456
Tomato (**18,162**)	5127	2127	1910	5730	1676	1592

The bold values represent the total number of images for that class in the dataset.

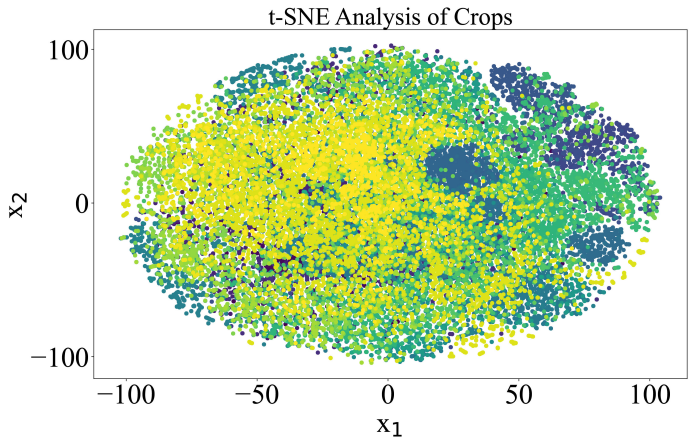


FIGURE 2
Visualization of the 38 classes in the PlantVillage data in two dimensions based on a t-SNE analysis. Each color in the spectrum represents one class in the PlantVillage dataset.

computational efficiency in MobileNetV1 Howard et al. (2017). MobileNetV2 added a resource-efficient block with inverted residuals and linear bottlenecks to improve efficiency Howard et al. (2018).

Later, MobileNetV3 Howard et al. (2019) extended MobileNetV2's efficient neural network design. MobileNetV3's backbone network, "MobileNetV3-Large," used linear bottlenecks and inverted residual blocks to increase accuracy and efficiency. Hierarchical squeeze-and-excitation (HSqueeze-and-Excitation) blocks adaptively recalibrated feature responses in MobileNetV3. Hard-Swish and Mish activation functions balanced computing efficiency and non-linearity. MobileNetV3 used neural architecture search to find optimal network architectures.

MobileNetV3-small was created for resource-constrained situations. Its tiny, lightweight neural network system is efficient and accurate. MobileNetV3-small achieved this through architectural optimizations, a simplified design, and decreased complexity. A reduced network footprint reduced parameters and operations. MobileNetV3-compact solved several real-world problems with low computing resources or edge device deployment with a compact but efficient architecture. It introduced several key components to optimize performance and achieve high accuracy with fewer parameters.

2.2.1 Initial convolution

An RGB image of size (*B,H,W*,3), where *B* is the batch size, *H* is the height, and *W* is the width, is used as an input. The image is passed through a standard convolutional layer with a small filter size (*e.g.*, 3x3) and a moderate number of channels (*e.g.*, 16).

2.2.2 Bottleneck residual blocks

MobileNetV3-small uses inverted bottleneck residual blocks, similar to its predecessor, MobileNetV2. The architecture is shown in Figure 3. Each block begins with a depth-wise convolution, which convolves each input channel separately with its small filter (*e.g.*, 3x3), significantly reducing the computational cost. The depth-wise convolution is followed by a point-wise convolution with 1×1 filters to increase the number of channels. A nonlinear activation function (*e.g.*, ReLU) is applied to introduce nonlinearity.

2.2.3 Squeeze-and-excite module

The Squeeze-and-Excite (SE) module is incorporated into the MobileNetV3-small architecture to improve feature representation and adaptively recalibrate channel-wise information. The SE module contains two steps:

- Squeeze: Global average pooling is applied to the feature maps, reducing spatial dimensions to 1×1.
- Excite: Two fully connected (FC) layers are used to learn channel-wise attention weights. These weights are multiplied with the original feature maps to emphasize essential features and suppress less relevant ones.

2.2.4 Stem blocks

MobileNetV3-small introduces stem blocks to further enhance feature extraction at the beginning of the network. The stem block consists of a combination of depth-wise and point-wise convolutions with nonlinear activation.

2.2.5 Classification head

After multiple stacked bottleneck blocks and SE modules, the final feature maps are passed through a classification head to make predictions. Global average pooling is applied to the feature maps to reduce spatial dimensions to 1×1. The output of global average pooling is then fed into a fully connected layer with "softmax" activation to produce *K* class probabilities, as shown in Figure 4. The overall architecture is shown in Table 2.

The architecture focuses on reducing the number of parameters while maintaining competitive accuracy. The number of parameters in MobileNetV3-small is 1.5 million, which makes it suitable for deployment on resource-constrained devices and applications that require real-time inference.

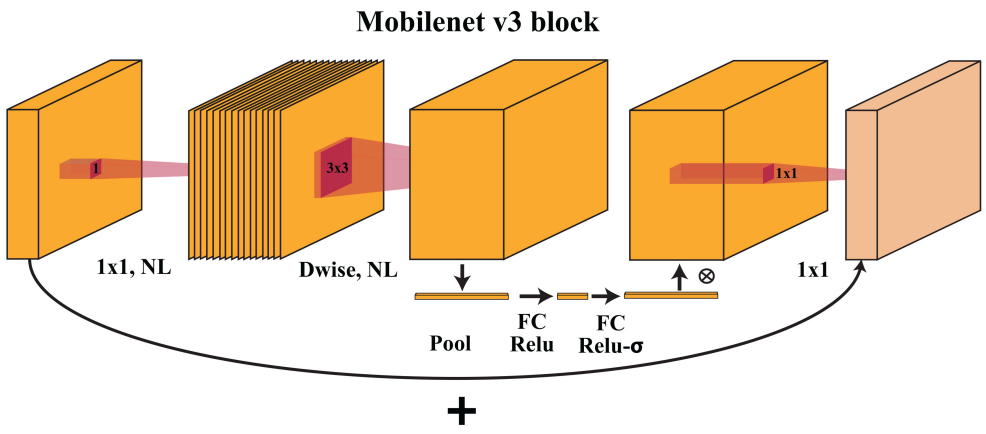


FIGURE 3
The MobileNetV3 block uses depthwise and pointwise convolutions to collect spatial patterns and integrate features. These blocks balance computing performance and precision, helping MobileNetV3 interpret complicated visual data.

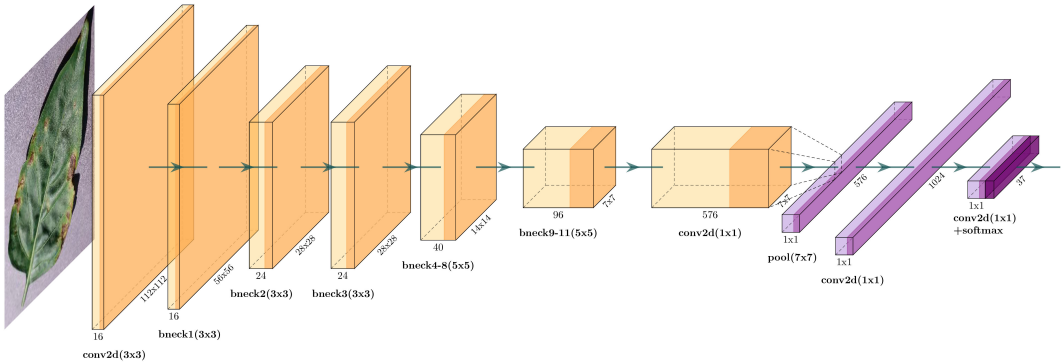


FIGURE 4
It shows the overall architecture of MobileNet-V3 Small. It includes a lightweight neural network design featuring depth-wise convolutions, inverted residuals, and a squeeze-and-excitation module for efficient feature extraction targeted for mobile and edge devices.

TABLE 2 Specification of MobileNetV3-Small.

Input	Operator	Exp-Size	#Out	SE	NL	Stride
224 × 224 × 3	Conv2d, 3×3	-	16	-	HS[b]	2
112 × 112 × 16	BottleNeck, 3 × 3	16	16	✓	RE[c]	2
56 × 56 × 16	BottleNeck, 3 × 3	72	24	-	RE	2
28 × 28 × 24	BottleNeck, 3 × 3	88	24	-	RE	1
28 × 28 × 24	BottleNeck, 5 × 5	96	40	✓	HS	2
14 × 14 × 40	BottleNeck, 5 × 5	240	40	✓	HS	1
14 × 14 × 40	BottleNeck, 5 × 5	240	40	✓	HS	1
14 × 14 × 40	BottleNeck, 5 × 5	120	48	✓	HS	1
14 × 14 × 48	BottleNeck, 5 × 5	144	48	✓	HS	1
14 × 14 × 48	BottleNeck, 5 × 5	288	96	✓	HS	2
7 × 7 × 96	BottleNeck, 5 × 5	576	96	✓	HS	1
7 × 7 × 96	BottleNeck, 5 × 5	576	96	✓	HS	1
7 × 7 × 96	Conv2d, 1×1	-	576	✓	HS	1
7 × 7 × 576	Pool, 7 × 7	-	-	-	-	1
1 × 1 × 576	Conv2d, 1 × 1, NBN[a]	-	1024	-	HS	1
1 × 1 × 1024	Conv2d, 1×1, NBN	-	K	-	-	1

Conv 2 d, Convolution 2 DBottleNeck: Bottleneck Residual Blocks.
NBN, No Batch Normalization HS: Hard-Swish activation function.
RE, Rectified Exponential Linear Unit activation function Pool: Pooling Layer.
"✓" represents that squeeze-excitation (SE) layer is used in that bottleneck block and "-" represents SE-layer is not utilized.

2.3 Model optimization

Model optimization, or quantization, is an essential deep-learning technique that reduces a neural network's memory footprint and computational complexity. Quantization enables efficient deployment on resource-constrained devices, such as mobile phones, peripheral devices, and microcontrollers, by converting the weights and activations of a full-precision model into lower-precision representations (e.g., 8-bit integers) Zhu et al. (2016). The procedure entails careful optimization to minimize the impact on model performance while achieving significant gains in model size reduction and faster inference times. Static quantization quantifies model weights and activations during training, whereas dynamic quantization quantifies model weights and activations based on the observed activation range at runtime.

For model quantization, the "Pytorch" built-in quantization tool was used Pytorch (2023). The PyTorch library's torch.quantization.quantize dynamic function was used to dynamically quantify particular layers in a given classifier model. The torch.quantization.quantize dynamic function clones the input

"model" before converting it into a quantized form. It then locates the cloned model's layers corresponding to the requested classes, such as Linear (2D convolutional layers) and Conv2d (2D convolutional layers). The weights and activations of each recognized layer are subjected to dynamic quantization. The activations are quantized at runtime depending on the observed dynamic range during inference, whereas the weights are quantized to int8 (Integer stored with 8 bit). The cloned model replaces the quantized layers while leaving the other layers in their original floating-point format. Compared to the original full-precision model, the quantized model has less memory and better computational efficiency, and it is prepared for inference on hardware or platforms that support integer arithmetic.

While quantization is our chosen method, it is important to acknowledge that there are other effective techniques for compressing deep learning models. These include knowledge distillation, where a smaller model is trained to emulate a larger one Hinton et al. (2015), pruning, which involves removing less important neurons Han et al. (2015), and low-rank factorization, a technique for decomposing weight matrices Jaderberg et al. (2014). Each of these methods offers unique advantages in model compression and can be particularly beneficial in scenarios with limited computational resources. However, for the goals and constraints of our current study, quantization emerged as the most suitable approach.

The above technique was employed to quantize "Linear" and "Conv2d" layers with lower-precision representations, *i.e.*, 8-bit.

2.4 Model training

For the model training, the MobileNetV3-small model from PyTorch, trained on ImageNet data, was employed. The training pipeline was simple as it did not involve any preprocessing of the image data. The model was fed with PlantVillage images of resolution 224×224. The hardware specifications were as follows:

- Processor: 11th Gen Intel(R) Core(TM) i9-11950H @ 2.60 GHz 2.61 GHz
- RAM: 64 GB
- GPU: Intel(R) UHD Graphics & NVIDIA RTX A3000

Although the model was trained on a GPU, the final quantized model was intended for CPU and edge devices. The optimizer parameters were as follows:

- Optimizer: Adam optimizer
- Betas: (0.5,0.99)
- Learning rate: 0.0001

Some additional model-training hyperparameters included:

- Batch Size: 64
- Epochs: 200
- Training Data Percentage: 80%
- Validation & Test Data Percentage: 10% each.

3 Results

The training and testing dataset included samples from all 38 classes. "Cross-entropy" was used as the loss function for the classification. The model's performance was evaluated based on two key metrics: Accuracy (Equation 1) and F1 score (Equation 4). Accuracy, defined as the proportion of correctly identified classes to the total number of classes, reflects the overall effectiveness of the model in classification tasks. In our study, the initial accuracy of the pre-trained model was 97%, which increased to a maximum test accuracy of 99.50% at the 154-th epoch. This metric essentially gauges the model's ability to label classes correctly. On the other hand, the F1 score, a harmonic mean of precision (Equation 2) (the proportion of true positive predictions in the total positive predictions) and recall (Equation 3) (the proportion of true positive predictions in the actual positive cases), measures the model's ability to accurately identify positive examples while minimizing false positives. This metric is especially useful in understanding the model's precision and robustness in identifying correct classifications without mistakenly labeling incorrect ones as correct. The trajectory of the model's accuracy with MobileNetV3-Small is shown in Figure 5. Similarly, the training loss, *i.e.*, cross-entropy loss, rapidly approached 0 and was ultimately reduced to 0 at the 136-th epoch. The trajectory of the training loss for MobileNetV3-Small is depicted in Figure 6.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad \text{(Eq. 1)}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad \text{(Eq. 2)}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad \text{(Eq. 3)}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{(Eq. 4)}$$

Later, the model was quantized, and the parameters were reduced to 0.9 million without reducing the accuracy of 99.50%. The inference time of the model was 0.01 seconds, and it achieved a frame rate of 100 frames per second (FPS) when running on a CPU. The higher-dimensional latent space of the model was also visualized using t-SNE Van der Maaten and Hinton (2008). 54,309 images of 38 classes were input to the trained model, and the output from the second-to-last layer of the MobileNetV3-small, which had dimensions of 1024, was obtained. Using t-SNE, the dimensions were reduced to 2, and the results were plotted to see the underlying classification modeling of the model. The results are shown in Figure 7. By forming distant clusters, it can be seen that the model efficiently classified 38 classes of plants.

Finally, the model was compared with other state-of-the-art architectures applied to the PlantVillage dataset. The comparison was based on three parameters, *i.e.*, the number of model parameters, model accuracy, and F1 score. The comparison is shown in Table 3. In the list of architectures, Schuler Schuler et al. (2022) had the highest accuracy and F1 score, and

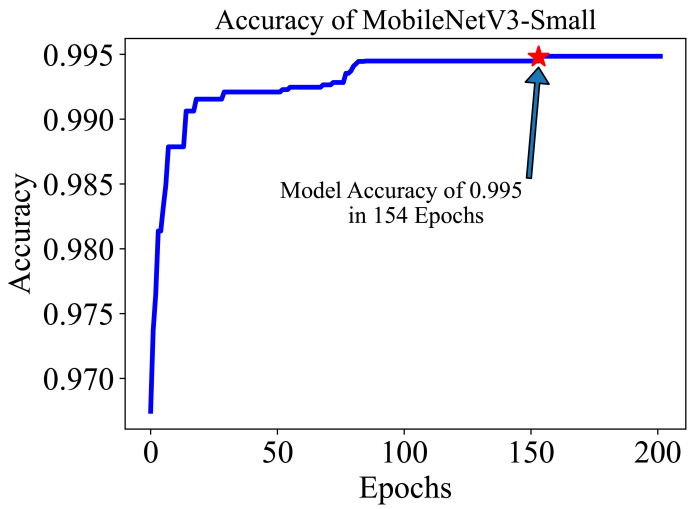


FIGURE 5
After training for 200 in epochs, the MobileNetV3-small gained an accuracy of 99.50 in roughly 154 epochs. The initial accuracy is approximately 97.0% because we used a pre-trained model.

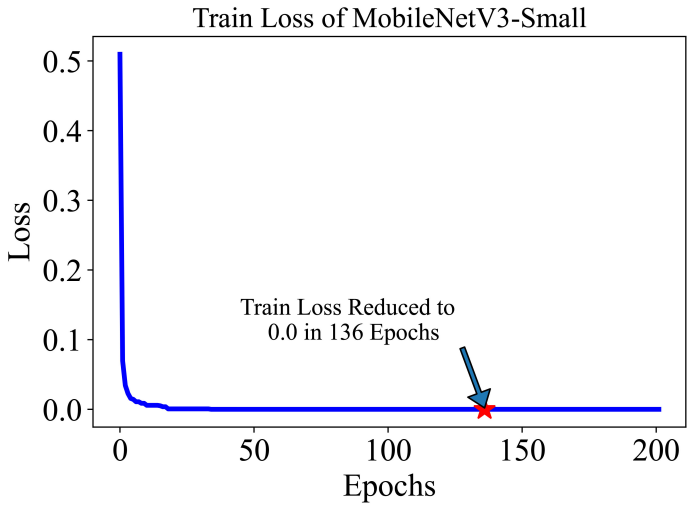


FIGURE 6
The training loss of MobileNetV3-small in 200 epochs quickly decreases and settles to 0.0 at 136 Epoch. The lower initial loss is the result of the pre-trained model.

Geetharamani Geetharamani and Pandian (2019) had the least number of parameters, 0.2M. The proposed solution had the highest accuracy (99.50%) and F1 score (0.9950). However, the number of parameters was 0.9M, which was 5 times less than the model suggested by the Schuler et al. (2022) model.

4 Discussion

Large model sizes can pose significant challenges to their practical application in classification problems within agriculture. Such problems often necessitate real-time or near-real-time solutions, especially when identifying pests and diseases or assessing crop health. Bulky models can slow the processing of data, causing delays that might compromise timely interventions. Deploying these models on edge devices, frequently used in agriculture for on-site analysis, becomes problematic due to their computational and memory constraints. Furthermore, in regions with limited connectivity, transferring data for cloud-based processing by large models can be bandwidth-intensive, leading to additional lags. The energy and financial costs of running extensive models can also be prohibitive for many agricultural applications, especially for small-scale or resource-constrained farmers. Additionally, the adaptability of these models can be limited; training and fine-tuning them to cater to the diverse and evolving classification needs of different agricultural contexts can be

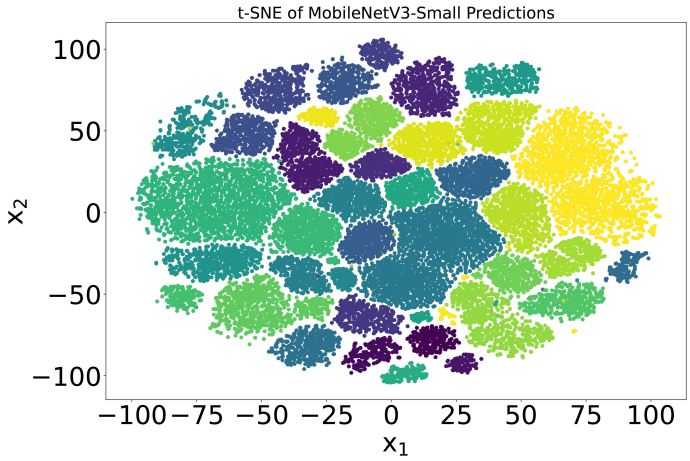


FIGURE 7
The t-SNE visualization of latent space of trained MobilenetV3-small model The output is from the second-last layer with a dimension of 1024, which is reduced to 2 using t-SNE. Each color in the spectrum represents one plant class in the PlantVillage dataset.

TABLE 3 Results comparison on PlantVillage dataset.

Author	Architecture	Parameters	Accuracy	Fl-score
Proposed	**MobileNetV3-small**	0.9**M**	**99.50**%	**0.9950**
Schiller Schuler et al. (2022)	Inception V3 (Modifed)	5**M**	99.48%	0.9923
Mohanty Mohanty et al. (2016)	GoogLeNet	5**M**	98.37%	0.9836
Mohanty Mohanty et al. (2016)	AlexNet	60**M**	97.82%	0.9782
Toda Toda and Okura (2019)	Inception V3	5**M**	97.15%	0.9720
Geetharamani Geetharamani and Pandian (2019)	9 layers CNN	**0.2M**	96.46%	0.9815
Mohanty Mohanty et al. (2016)	GoogLeNet	5**M**	96.21%	0.9621
Mohanty Mohanty et al. (2016)	AlexNet	60**M**	94.52%	0.9449

TThe bold values correspond to the best value in each column.

challenging. In essence, while large models might boast superior accuracy, their size can often impede their practicality and responsiveness in addressing agricultural classification problems.

Previously proposed state-of-the-art solutions Schuler et al. (2022); Mohanty et al. (2016) for plant disease classifications achieve good accuracy. However, they have practical limitations in size and deployment. To overcome this issue, we proposed a solution with MobileNetV3-small. Its compact and efficient architecture enables rapid data processing, facilitating real-time agricultural interventions, such as pest detection or disease identification. The model's low power consumption makes it ideal for battery-operated field devices, and its adaptability ensures relevance to diverse agricultural needs. Furthermore, its cost-effectiveness and ease of maintainability make it a practical choice for agricultural scenarios, offering a balance of high performance and resource efficiency.

While MobileNetV3 offers impressive efficiency and is optimized for edge devices, it has certain tradeoffs. The primary disadvantage is that, in pursuit of a lightweight and compact design, it might not always achieve the highest possible accuracy, especially when compared to larger, more complex models designed for high-performance tasks. This reduction in accuracy can be a limitation for applications where even a slight drop in precision can have significant consequences. Additionally, certain customizations or fine-tuning required for specific tasks might not be as straightforward, given its specialized architecture. Thus, while MobileNetV3 is advantageous for many scenarios, it may not be the best fit for situations demanding the utmost accuracy and complex model customizations.

The PlantVillage dataset, while comprehensive, exhibits an unbalanced nature with respect to the number of images available for different plant diseases. Unbalanced data can significantly impact deep learning model performance. Such datasets have extremely skewed class distributions, with one or a few classes having disproportionately more samples. This imbalance causes many issues. Deep learning models trained on unbalanced data tend to focus accuracy on the dominant class over the minority classes, biasing them towards the majority class. As a result, the model's ability to generalize and forecast underrepresented classes falls, resulting in poor training and evaluation performance. Due to

their rarity, the model may have trouble learning significant patterns from minority classes, making it less likely to recognize and classify cases from these classes.

MobileNetV3's efficient and compact design offers a strategic advantage in addressing the imbalances inherent in datasets like PlantVillage. By leveraging transfer learning, a pre-trained MobileNetV3 is later fine-tuned on PlantVillage classes, harnessing generalized features to counteract dataset disparities. Its lightweight nature facilitates rapid training, enabling extensive data augmentation to enhance underrepresented classes. Furthermore, MobileNetV3 can serve as a potent feature extractor, with the derived features being suitable for synthetic sample generation techniques like SMOTE or ADASYN to achieve class balance. The model's cost-effectiveness allows for swift iterative experiments, incorporating regularization techniques to deter overfitting dominant classes. Overall, MobileNetV3 presents a versatile toolset for researchers to navigate and mitigate the challenges of unbalanced datasets.

Training MobileNetV3 on the PlantVillage dataset and applying it to new images introduces challenges related to generalization. Absent categories, like healthy orange and squash, might be misclassified into familiar classes the model has seen. Diseases not in the training data, such as brown spots on soybeans, could be wrongly identified as another visually similar ailment or even as a healthy state. The model might also grapple with new images that differ in lighting, resolution, or background, especially if not exposed to such variations during training. The inherent class imbalance in the PlantVillage dataset, if unaddressed, can further bias the model towards overrepresented classes, affecting its performance on new or underrepresented classes. In essence, while MobileNetV3 is efficient, its accuracy on unfamiliar data hinges on the diversity and comprehensiveness of its training data.

Quantization compresses neural models by reducing the bit representation of weights and activations, enhancing memory efficiency and inference speed. "Weight quantization" reduces weight precision after training. This post-training quantization can introduce errors, as the model was not trained to accommodate the reduced precision. This can sometimes lead to a significant drop in model performance. Whereas "quantization-aware training" adjusts the model during training to a lower precision. PyTorch's torch.quantization.quantize dynamic is notable, dynamically quantizing mainly the linear layers. This balances reduced model size and computational efficiency, preserving accuracy and making it apt for models with varied layer intensities.

The proposed pipeline, while efficient in its current application, does have certain limitations. Firstly, the pipeline is optimized for a specific dataset and task; scaling it to handle larger datasets or adapting it to different types of plants and diseases might require additional modifications. Secondly, the maintenance and updating of the model could present minor challenges. Ensuring that the model remains current with the latest data and continuously performs at its peak might necessitate regular updates and maintenance, which can be resource-intensive over time.

As we move forward from this study, we plan to extend our research to include a wider range of real-world datasets, such as those suggested by Tomaszewski Tomaszewski et al. (2023) and Ruszczak Ruszczak and Boguszewska-Mańkowska (2022). Our current focus on a controlled dataset lays the groundwork for this expansion. In future work, we aim to test and refine our models against the complexity of real-world agricultural scenarios, enhancing their generalization capabilities. This step-by-step approach, progressing from controlled conditions to more diverse datasets, aims to develop robust and adaptable deep-learning models for effective plant disease detection in practical agricultural settings.

5 Conclusion

The traditional and cutting-edge models have shown good accuracy; however, their suitability for onthe-ground applications with limited resources is often limited. By focusing on maximizing accuracy within resource constraints, we demonstrated the real-life usability of deep learning models in agricultural settings. Using the MobileNetV3-small model with approximately 1.5 million parameters, we achieved a test accuracy of around 99.50%, offering a cost-effective solution for accurate plant disease detection. Furthermore, post-training optimization, including quantization, reduced the model parameters to 0.9 million, enhancing inference efficiency. The final model in ONNX format enables seamless deployment across multiple platforms, including mobile devices. These contributions ensure that deep learning models can be practically and efficiently utilized in real-world agricultural applications, advancing precision farming practices and plant disease detection.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

ATK: Methodology, Software, Writing – original draft. SJ: Supervision, Writing – review & editing. ARK: Conceptualization, Methodology, Validation, Writing – original draft. SL: Formal analysis, Methodology, Validation, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alabbasy, F. M., Abohamama, A., and Alrahmawy, M. F. (2023). Compressing medical deep neural network models for edge devices using knowledge distillation. *J. King Saud. University-Computer Inf. Sci.*, 101616.

Al Koutayni, M. R., Reis, G., and Stricker, D. (2023). Deepedgesoc: End-to-end deep learning framework for edge iot devices. *Internet Things* 21, 100665. doi: 10.1016/j.iot.2022.100665

Amara, J., Bouaziz, B., and Algergawy, A. (2017). "A deep learning-based approach for banana leaf diseases classification," in *Datenbanksysteme für Business, Technologie und Web (BTW 2017) Workshopband*.

Dutta, A., Gupta, A., and Zissermann, A. (2016) *Vgg image annotator (via)*. Available at: http://www.robots.ox.ac.uk/~vgg/software/via.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* 145, 311–318. doi: 10.1016/j.compag.2018.01.009

Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., et al. (2018). Kiwifruit detection in field images using faster r-cnn with zfnet. *IFAC-PapersOnLine* 51, 45–50. doi: 10.1016/j.ifacol.2018.08.059

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17, 2022. doi: 10.3390/s17092022

Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., and Iyatomi, H. (2016). "Basic investigation on a robust and practical plant diagnostic system," in *2016 15th IEEE international conference on machine learning and applications (ICMLA)* (IEEE). 989–992.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics* 36, 193–202. doi: 10.1007/BF00344251

Geetharamani, G., and Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electrical Eng.* 76, 323–338. doi: 10.1016/j.compeleceng.2019.04.011

Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv*.

Hao, J., Subedi, P., Ramaswamy, L., and Kim, I. K. (2023). Reaching for the sky: Maximizing deep learning inference throughput on edge devices with ai multi-tenancy. *ACM Trans. Internet Technol.* 23, 1–33. doi: 10.1145/3546192

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv*.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019). "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*. 1314–1324.

Howard, A., Zhmoginov, A., Chen, L.-C., Sandler, M., and Zhu, M. (2018). Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv*.

Jaderberg, M., Vedaldi, A., and Zisserman, A. (2014). Speeding up convolutional neural networks with low rank expansions. *arXiv*. doi: 10.5244/C.28.88

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25.

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., et al. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Appl. Sci.* 10, 1245. doi: 10.3390/app10041245

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7,1419. doi: 10.3389/fpls.2016.01419

Nalepa, J., Antoniak, M., Myller, M., Lorenzo, P. R., and Marcinkiewicz, M. (2020). Towards resourcefrugal deep convolutional neural networks for hyperspectral image segmentation. *Microprocessors Microsys.* 73, 102994. doi: 10.1016/j.micpro.2020.102994

PlantVillage-Dataset (2016). GitHub. Available at: https://github.com/spMohanty/PlantVillage-Dataset/tree/master.

Pytorch. (2023). *Quantization pytorch 2.0 documentation*.

Qiang, Z., He, L., and Dai, F. (2019). "Identification of plant leaf diseases based on inception v3 transfer learning and fine-tuning," in *International Conference on Smart City and Informatization* (Springer). 118–127.

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., and Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Front. Plant Sci.* 8, 1852. doi: 10.3389/fpls.2017.01852

Ruszczak, B., and Boguszewska-Mańkowska, D. (2022). Deep potato–the hyperspectral imagery of potato cultivation with reference agronomic measurements dataset: Towards potato physiological features modeling. *Data Brief* 42, 108087. doi: 10.1016/j.dib.2022.108087

Sarda, A., Dixit, S., and Bhan, A. (2021). "Object detection for autonomous driving using yolo algorithm," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)* (IEEE). 447–451.

Schuler, J. P. S., Romani, S., Abdel-Nasser, M., Rashwan, H., and Puig, D. (2022). Color-aware two-branch dcnn for efficient plant disease classification. *MENDEL* 28, 55–62. doi: 10.13164/mendel.2022.1.055

Setiawan, W., Ghofur, A., Rachman, F. H., and Rulaningtyas, R. (2021). Deep convolutional neural network alexnet and squeezenet for maize leaf diseases image classification. *Kinetik: Game Technol. Inf. Sys. Comput. Netw. Comput. Electr. Control.*

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* 2016. doi: 10.1155/2016/3289801

Swaminathan, A., Varun, C., Kalaivani, S., et al. (2021). Multiple plant leaf disease classification using densenet-121 architecture. *Int. J. Electr. Eng. Technol.* 12, 38–57.

Toda, Y., and Okura, F. (2019). How convolutional neural networks diagnose plant disease. *Plant Phenomics*. doi: 10.34133/2019/9237136

Tomaszewski, M., Nalepa, J., Moliszewska, E., Ruszczak, B., and Smykała, K. (2023). Early detection of solanum lycopersicum diseases from temporally-aggregated hyperspectral measurements using machine learning. *Sci. Rep.* 13, 7671. doi: 10.1038/s41598-023-34079-x

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9.

Yang, L., Yu, X., Zhang, S., Long, H., Zhang, H., Xu, S., et al. (2023). Googlenet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agric.* 204, 107543. doi: 10.1016/j.compag.2022.107543

Zhu, C., Han, S., Mao, H., and Dally, W. J. (2016). Trained ternary quantization. *arXiv*.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 11 January 2024
DOI 10.3389/fpls.2023.1303429

OPEN ACCESS

EDITED BY
José Dias Pereira,
Instituto Politecnico de Setubal (IPS), Portugal

REVIEWED BY
Sapna Langyan,
Indian Council of Agricultural Research (ICAR), India
Ali Parsaeimehr,
Delaware State University, United States
Carlos Banha,
Instituto Politecnico de Setubal (IPS), Portugal

*CORRESPONDENCE
Emilio Vello
emilio.vello@mcgill.ca
Thomas E. Bureau
thomas.bureau@mcgill.ca

RECEIVED 29 September 2023
ACCEPTED 20 December 2023
PUBLISHED 11 January 2024

CITATION
Vello E, Letourneau M, Aguirre J and Bureau TE (2024) Integrated web portal for non-destructive salt sensitivity detection of *Camelina sativa* seeds using fluorescent and visible light images coupled with machine learning algorithms.
Front. Plant Sci. 14:1303429.
doi: 10.3389/fpls.2023.1303429

Integrated web portal for non-destructive salt sensitivity detection of *Camelina sativa* seeds using fluorescent and visible light images coupled with machine learning algorithms

Emilio Vello*, Megan Letourneau, John Aguirre and Thomas E. Bureau*

Department of Biology, McGill University, Montreal, QC, Canada

Climate change has created unprecedented stresses in the agricultural sector, driving the necessity of adapting agricultural practices and developing novel solutions to the food crisis. *Camelina sativa* (Camelina) is a recently emerging oilseed crop with high nutrient-density and economic potential. Camelina seeds are rich in essential fatty acids and contain potent antioxidants required to maintain a healthy diet. Camelina seeds are equally amenable to economic applications such as jet fuel, biodiesel and high-value industrial lubricants due to their favorable proportions of unsaturated fatty acids. High soil salinity is one of the major abiotic stresses threatening the yield and usability of such crops. A promising mitigation strategy is automated, non-destructive, image-based phenotyping to assess seed quality in the food manufacturing process. In this study, we evaluate the effectiveness of image-based phenotyping on fluorescent and visible light images to quantify and qualify Camelina seeds. We developed a user-friendly web portal called SeedML that can uncover key morpho-colorimetric features to accurately identify Camelina seeds coming from plants grown in high salt conditions using a phenomics platform equipped with fluorescent and visible light cameras. This portal may be used to enhance quality control, identify stress markers and observe yield trends relevant to the agricultural sector in a high throughput manner. Findings of this work may positively contribute to similar research in the context of the climate crisis, while supporting the implementation of new quality controls tools in the agri-food domain.

KEYWORDS

phenotyping, phenomics, artificial intelligence, AI, abiotic stress, salinity, *Camelina sativa*, image analysis

1 Introduction

In recent years, an ever increasing demand for land along with unprecedented environmental consequences due to climate change has significantly impacted agricultural productivity. The prevalence of saline soils is increasing worldwide due to a lack of fresh water, prolonged periods of drought and rising sea levels (Hassani et al., 2021). It is estimated that over one billion hectares (ha) of global land are currently affected by salinity, with this number increasing by two Mha per year. The issue is widespread and affects over 100 countries with severe impacts in India, China, the United States, Turkey and many other regions. For example, over 30% of land in Iran is salt-affected, leading to ongoing economic and environmental implications including decreased productivity and soil erosion, which numerous countries stand to face (Singh, 2021). Increased concentrations of sodium chloride (NaCl), lead to ionic toxicity and osmotic stress in plants. While some plants such as halophytes have the ability to tolerate salt stress, traditional crops for food use are severely impacted by NaCl, leading to inhibition of growth and low yield production (Morales et al., 2017). When coupled with other abiotic stresses such as drought, heavy metal exposure, high temperatures, and reduced humidity, these factors become limiting for crop production, leading to huge economic losses and social concerns regarding food security (Shah et al., 2018; Razzaq et al., 2021).

Camelina sativa (Camelina) is an undervalued oilseed crop belonging to the *Brassicaceae* family, closely related to *Arabidopsis thaliana* and other economically relevant *Brassicaceae* such as canola and the cabbage (Berti et al., 2016). This crop is native to East European/West Asian regions and was first domesticated in the late Neolithic era before being largely replaced by other competitor crops. Despite being well adapted to Canada and the northern United States due to the semi-arid, temperate and short-season climates, Camelina is not widely produced in North America (Vollmann and Eynck, 2015). It is only in more recent decades that Camelina has begun to receive a renewed interest due to its advantageous properties including low input requirements, tolerance to cold temperatures and pests and a high nutrient-density (Masella et al., 2014). Camelina seeds also contain uncommonly high levels of alpha-linolenic acid, an essential omega-3 fatty acid required for proper physical and cognitive maintenance, making it a nutritious food source (Kagale et al., 2014; Berti et al., 2016).

In recent years, there has been a surge in plant phenomics equipment and platforms, ranging from compact desktop setups to large-scale field phenotyping machines and even unmanned aerial vehicles (Vello et al., 2022; Sarkar et al., 2023). However, there is a limited availability of user-friendly tools for analyzing the vast amount of data generated by these systems, and many of the existing tools are challenging for non-computer science users to navigate (Vello et al., 2015). Furthermore, Camelina, being an emerging crop, has not been as extensively investigated as other established crops such as *Brassica napus* (canola). Our understanding of the effect of abiotic stresses such as NaCl concentration on Camelina seeds therefore remains limited (Zanetti et al., 2021). In this study, we aim to address these challenges by investigating the potential of image-based phenotyping and automated analysis through a user-friendly web portal. The SeedML portal enables the analysis of morpho-colorimetric attributes of Camelina seeds and can in turn predict their salt status. This prediction is based on image analysis and machine learning algorithms, utilizing fluorescent or visible light images acquired from a plant phenomics platform. As phenomic systems continue to innovate in response to adapting needs in the agricultural sector, the availability of accessible and powerful analysis tools will play a vital role in their success.

2 Materials and methods

2.1 Plant growth and salt treatment

Protocol 1. Three *Camelina sativa* (Camelina) seeds (Celine variety), were sown in 5" pots with 250 g of Sunshine mix (75-85% Canadian Sphagnum peat moss, perlite and dolomite limestone) and 450 mL of water. Plants were grown in the McGill phytotron greenhouse with a 14 h / 10 h light/dark photoperiod at a temperature of 27°C/20°C day/night. Seven days after sowing (DAS), seedlings were thinned to one per pot based on size similarity. At DAS 20, salt stress was induced through saline water treatment (final NaCl concentrations of 0, 50, 100, 150 and 200 mM), prepared using a final volume of 450 ml of water (soil water capacity). Salt treatment was progressively applied twice a day over two days. Pots were watered every day to 700g to maintain a constant NaCl concentration. Classic 20-20-20 (N-P-K) fertilizer diluted 1:10 was applied at DAS 15. Plants were randomized three times a week to avoid any positional effect in the greenhouse.

Protocol 2. Similar to protocol 1 but using a water capacity of 350 mL and a final weight of 600 g. Environmental temperature was set at 24°C/20°C day/night and three salt concentrations were used (NaCl at 0, 200 and 250 mM). Fertilizer was applied at DAS 8 and 15.

Protocol 3. Similar to protocol 2 but plants were watered twice a week without weight control and only two levels of salt concentration were used (NaCl at 0 and 200 mM). Plants were fertilized once a week.

Protocol 4. Similar to protocol 1 but using 200 g of soil and a 400 mL water capacity at 0 and 200mM of salt. Plants were watered to 600 g twice a week. Salt stress was induced at DAS 34.

Plant batches: Four different batches of plants were grown in different seasons and using different protocols in a semi-controlled environment (greenhouse) in which light and temperature may fluctuate according to the external environmental conditions. Plants in batch A were grown using protocol 1 in Spring-Summer 2018, batch B using protocol 2 in Spring 2019, batch C using protocol 3 in Fall 2020 and batch D using protocol 4 during Winter 2018.

2.2 Seed preparation and imaging

Harvested seeds were dried for 30 days at room temperature and then stored at 4°C. A weighing pan and an electronic balance (PB3002 DeltaRange) were used to select 0.1g or 0.05 g seeds from each plant according to the set (Table 1). Seeds were then transferred to petri dishes and identified with barcodes. The image acquisition was performed with the LemnaTec HTS installed at the McGill Plant Phenomics Platform (MP3, http://mp3.biol.mcgill.ca), using the visible light camera piA2400-17gc and the fluorescent light camera scA1400-17gc. Three configurations were selected; visible light top illumination (VISFRONT); visible light back illumination (VISBACK); and fluorescent illumination between 400 and 500 nm (FLUO).

2.3 Software development

The three main components of the web portal software (the web interface, the image analysis and the machine learning implementation), were implemented on Java OpenJDK 17 + 35 and Apache Tomcat 10.1.10. The web portal was developed using JSP, HTML, JavaScript, CSS. The image analysis and machine learning modules were developed using ImageJ 1.53a (Schneider et al., 2012), Fiji (Schindelin et al., 2012) and weka 3.9.4 (Frank et al., 2016), respectively as main packages and Java as programming language. An adapted version of the "combined contour tracing and region labeling" proposed by Burger and Burge (2008, 2016) was implemented as part of the segmentation algorithm. SeedML was assigned as the name of the portal.

2.4 SeedML web portal

The web portal runs on a Dell R910 server with 512 GB of RAM and two MD1200 storage devices 72 TB at McGill University. The SeedML web portal is accessible through the internet address https://sites.google.com/view/seedml or http://mp3.biol.mcgill.ca/seedml. The prediction of the salt status analysis is performed in the following steps. 1) Seed detection setup; 2) Training images; 3) Testing images; 4) Process; 5) Phenotypic traits; 6) Seed classification. The portal could also be used to analyze morpho-colorimetric traits alone. In this case, steps 1, 3, 4 and 5 are required.

2.5 Seed detection setup

In this step, the user can select different thresholds for some image properties or the application of determined algorithms in order to set up the segmentation parameters, seed and background identification. It is possible to set the scale of pixels per centimeter assuming a pixel aspect ratio of one. The segmentation parameters are easily set up by clicking or dragging and dropping a sample image of a plate on the box under the title "original image". After clicking the refresh button, the processed images on the right box will give a preview of some intermediary (pre-processed) and final results of the segmentation. The adjustment and refreshing of the segmentation parameters is performed until the identification of the seeds is archived. This configuration can be downloaded to the local disk to be reused in future analysis. The portal has three pre-set configurations used for this article, visible light top illumination, visible light back illumination and fluorescent light.

2.6 Training images

One or more images for each growth condition (salt and normal) are uploaded by clicking or dragging and dropping to the respective panel. These images are used to train the different machine learning algorithms. The garbage icon allows the user to clean up the content of the panel. The uploading operation is successfully achieved when a scaled image and its names are shown in the corresponding list.

2.7 Testing images

The center panel is designed to upload the images of the seed plates to be analyzed by dragging and dropping or clicking. This section is also used if a morpho-colorimetric analysis only is desired. Before moving to the next step, the user has to wait until a small-scale copy of each image is shown in the center panel.

TABLE 1 Seed sets and plate batches.

Set	Plate number	Seed weight x plate	Image date	Batch	Growth season	Salt concentration (mM)
1	23	0.10 g	2019-11-14	B	Spring 2019	0, 200, 250
2	35	0.10 g	2020-01-17	A	Spring-Summer 2018	0,50, 100, 150, 200
3	18	0.10 g	2021-02-25	C	Fall 2020	0, 200
4	40	0.10 g	2021-03-02	A	Spring-Summer 2018	0, 50, 100, 150, 200
5	10	0.05 g	2021-03-02	A	Spring-Summer 2018	0, 200
6	15	0.10 g	2021-04-27	D	Winter 2018	0, 200
M	18	0.10 g	2020-01-22	A/B	A and B mixed	0, 200

2.8 Image analysis and classification process

Once the training and testing images are uploaded, the user can run the process of image analysis and classification using the start button. The classification process can be based on all, only morpho or only color attributes (Tables 2, 3 respectively). The button in the middle panel allows the user to change the option. Once the process is complete, the third panel central label will change from "X" to "✓".

2.9 Phenotypic traits

A summary table with the seed count and the average seed size, seed length, seed width and seed circularity per plate is shown. If the pixels/metric scale is set up, the metric attributes are displayed in millimeters. Clicking on the image name, a new web page is presented with the object (seed) research region, the original objects (seeds), the color classification, the false color representation and a table with selected morpho-colorimetric attributes per seed (Joly-Lopez et al., 2017; Vello et al., 2022). Each seed can be traced into the image using the ID attribute of the table in the "original objects" image. Most of the table can be downloaded in a comma-separated values (CSV) file format supported by a large variety of software such as Microsoft Excel, Google Sheets, LibreOffice, R.

2.10 Seed classification

The salt status of each plate is determined by the average of the percentage of salt/non-salt among all algorithms Table 4 included in the portal (Figure 1). If the percentage is greater than 50, then the plate is marked with the stress status. This section of the software displays a table containing the individual percentages for each algorithm and the predicted status of the plate. As described in "phenotypic traits", the details of the plate can be obtained by clicking on its name.

TABLE 2 Description of morphological features.

Identification	Definitions
Area	Number of pixels representing the seed in the image. (Joly-Lopez et al., 2017)
Perimeter	Length of the outer contour of the pixels representing the seed in the image. (Joly-Lopez et al., 2017)
Circularity	Ratio between the circumference square and the area. (Camargo et al. 2014)
Compactness	Ratio between the area and the perimeter (Burger and Burge, 2008, 2016).
Major axis	Axis where a physical body requires less effort to rotate. It extends from the centroid (center of gravity) to the widest part of the object (Burger and Burge, 2008, 2016), in this case the pixels presenting the seed in the image.
Minor axis	Axis perpendicular to the major axis.
Eccentricity	Ratio between the major axis and the minor axis of the digital plant (Burger and Burge, 2008, 2016). The minor axis extends from the centroid to the narrowest part perpendicular to the major axis.

2.11 Testing procedure

All the output data shown in this work has been processed using the SeedML portal in order to assess its power to identify morpho-colorimetric features of seeds and predict the salt status of the plates. The exception is the performance of the machine learning algorithms that has been done before the portal implementation. After uploading and processing the images into the portal, the morpho-colorimetric features were downloaded using the phenotypic traits option and plotted in R. The prediction tests

TABLE 3 Description of colorimetric features.

Identification	Description
Grey intensity peak (hisgreypeak)	Intensity value having the bigger frequency from the pixels representing the seed. It is the higher peak of the intensity value histogram. (Joly-Lopez et al., 2017)
Q_1 grey pixels (q1grey)	First quartile of the pixel grey value distribution. (R+G +B)/3.
Q_2 grey pixels (q2grey)	Second quartile of the pixel grey value distribution. (R+G +B)/3.
Q_3 grey pixels (q3grey)	Third quartile of the pixel grey value distribution. (R+G +B)/3.
Q_1 red channel pixels (q1r)	First quartile of the pixel red channel value distribution.
Q_2 red channel pixels (q2r)	Second quartile of the pixel red channel value distribution.
Q_3 red channel pixels (q3r)	Third quartile of the pixel red channel value distribution.
Q_1 green channel pixels (q1g)	First quartile of the pixel green channel value distribution.
Q_2 green channel pixels (q2g)	Second quartile of the pixel green channel value distribution.
Q_3 green channel pixels (q3g)	Third quartile of the pixel green channel value distribution.
Q_1 blue channel pixels (q1b)	First quartile of the pixel blue channel value distribution.
Q_2 blue channel pixels (q2b)	Second quartile of the pixel blue channel value distribution.
Q_3 blue channel pixels (q3b)	Third quartile of the pixel blue channel value distribution.
Higher 16 color class (hue16max)	Color class having the higher number of pixels from a hue channel 16 class pixel division in the HSB color space.
Higher 32 color class (hue32max)	Color class having the higher number of pixels from a hue channel 32 class pixel division in the HSB color space.
Higher 64 color class (hue64max)	Color class having the higher number of pixels from a hue channel 64 class pixel division in the HSB color space.

were divided into two groups: inside sets and between sets. For inside sets, three tests for each camera (FLUO: fluorescent, VISFRONT: visible top light, VISBACK: visible back light), attribute (all, only morpho, only color), set (1-6) and salt concentration (50 mM, 100 mM, 150 mM, 200 mM, and 250 mM) were performed (Supplementary Table 1). For between sets, the k-fold cross-validation method with k=10 (Sakeef et al., 2023) was used on 200 mM only since this concentration is present in all sets. The k-fold cross-validation prevents underfitting or overfitting of the model, aligning with the sample size and the split between testing and training in the various tests (Saharan et al., 2021; Charilaou and Battat, 2022; Prusty et al., 2022). The portal has been tested in Firefox and QuteBrowser.

2.12 Evaluation of the prediction process

The performance and effectiveness of the prediction status of seeds and plates is measured using five metrics commonly used in benchmarks of machine learning algorithms: accuracy (Equation 1), sensitivity Equation 2, specificity Equation 3, precision Equation 4 and F1 score Equation 5 (Xu et al., 2022; Yang et al., 2023).

$$Accuracy = \frac{\Sigma(true\ positives) + \Sigma(true\ negatives)}{total} \quad (1)$$

$$Sensitivity = \frac{\Sigma(true\ positive)}{\Sigma(true\ positive) + \Sigma(false\ negative)} \quad (2)$$

$$Specificity = \frac{\Sigma(true\ negatives)}{\Sigma(true\ negatives) + \Sigma(false\ positives)} \quad (3)$$

$$Precision = \frac{\Sigma(true\ positives)}{\Sigma(true\ positives) + \Sigma(false\ positives)} \quad (4)$$

$$F1\ Score = \frac{2 \times Precision \times Sensitivity}{Precision \times Sensitivity} \quad (5)$$

2.13 Portal availability

The SeedML portal can be accessed at https://sites.google.com/view/seedml, where images, additional information, and access to the portal, including current and future mirrors, can be found. Alternatively, it is possible to access it directly at http://mp3.biol.mcgill.ca/seedml. For any inquiries or issues, including mirror installations, please contact the corresponding authors.

3 Results

3.1 Morpho-colorimetric features under normal and salt conditions

Morpho-colorimetric seed features were compared between any concentration of salt and non-salt growing conditions under visible back light (VISBACK) and fluorescent light cameras (FLUO). The area, perimeter, major and minor axis have shown higher values in the salt group under FLUO (Figure 2). However, this pattern was not observed in the VISBACK (Figure 3). In both cameras, the eccentricity has shown higher values in the non-salt group among all the sets. The color related features in the VISBACK have not presented defined patterns among the sets. For example, the red lower quartile feature in the non-salt group is lower in set number 1 and higher in set number 3. The grey intensity peak non-salt value is higher in set number 2 but it is lower in sets 4 and 6. In the case of FLUO, a pattern was found in some of the color-related features. This is the case in the red lower, median and higher quartiles where the salt group has shown higher values. Almost no signal was observed from the blue channel. This was expected as the fluorescent information is represented in the red channel under FLUO.

The values of the area in non-salt condition groups are approximately 150 px for sets 1, 2, 4, 5 and M and slightly higher than 200 px for sets 3 and 6 (Figure 2A), under FLUO. This pattern is observed for the perimeter, major and minor axis as well (Figures 2B, D, E). The sets 2, 4 and 5 come from batch A and set 1 from batch B. The M set is a mix of A and B. Set 3 and 6 are taken from batch C and D respectively. In the VISBACK images, the area values for sets 3 and 6 are slightly higher than the other batches. The non-area related features, circularity, compactness and eccentricity show the same patterns among the sets under the VISBACK and FLUO as expected (Figures 2C, F, G, 3C, F, G).

3.2 Pixel to metric conversion agreement and seed count

The conversion from pixels to metrics was done using the inside diameter of the petri dish plate at 8.50 cm. The diameter of the plate under the FLUO is 846.50 pixels (px) giving 99.58 px/cm (Supplementary Figure 1A). The same diameter under the VISBACK is 1812 px giving 213.17 px/cm (Supplementary Figure 1B). The double of the major and the minor axes can be used as a proxy to the length and width respectively. The average major and minor axes in the FLUO are the 9.76 px and 5.17 px giving a length of 1.95 mm and a width of 1.03 mm. In the case of the VISBACK, the averages are 21.75 px and 10.83 px giving a length of 2 mm and a width of 1 mm. Our manual calculation using a ruler on the actual seeds (Supplementary Figure 1C), has shown a length of 2 mm. The automatic seed count from the images having 0.10 g/seeds per plate revealed that the average number of seeds is 92, (95% CI [89.53, 95.20]) for FLUO, 84, (95% CI [80.46, 88.25]) for VISFRONT and 95, (95% CI [92.54, 99.31]) for VISBACK (Supplementary Figure 1D).

3.3 Performance of machine learning algorithms in individual seeds

The accuracy of the 13 pre-selected machine learning algorithms from the WEKA package (Frank et al., 2016) to

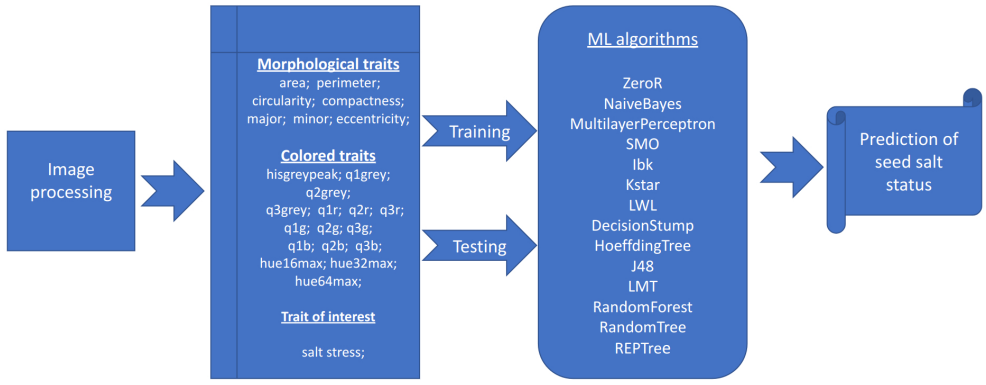


FIGURE 1
Image and data analysis pipeline. Graphical representation of the analysis pipeline implemented in the SeedML portal for plants grown under normal or salt stress conditions.

predict salt status of the seeds was tested using set 1 and 2 on individual seeds. FLUO and VISBACK images were computed all together (Figure 4), using one or two plates as training for each condition. The ZeroR showed an accuracy of 52%, NaiveBayes 74%, MultilayerPerceptron 73%, SMO 73%, IBk 70%, Kstar 71%, LWL 72%, DecisionStump 73%, HoeffdingTree 73%, J48 72%, LMT 75%, RandomForest 74%, RandomTree 71% and REPTree 72%. The ZeroR algorithm was not implemented in the portal because of its low accuracy compared to the rest of the algorithms.

3.4 Portal performance inside sets using 0 and 200 mM (0-200mM) salt concentrations

The performance of the portal was evaluated within various sets, specifically focusing on salt concentrations of 0 mM and 200 mM (0-200mM). This assessment encompassed both the predictive capabilities of the portal and the type of camera used (fluorescent and visible light), across different groups. Each concentration of salt and non-salt plates was subjected to triplicate testing. During the training phase, either one or three plates were employed, depending on the specific test. The majority of tests were conducted with just one training plate per group, which represents the minimum information necessary for the classification algorithms.

In Figure 5, confusion matrices for the 0-200 mM salt concentrations, utilizing one training plate for each group, are presented. Among the 243 plates analyzed, 96 plates were accurately classified as non-salt, and 130 were correctly identified as salt (Figure 5A). Only 3 were incorrectly classified as salt, and 14 were misclassified as non-salt when using fluorescent images (FLUO) with all attributes.

When only color attributes were considered, 2 plates were wrongly classified as non-salt, and 11 were misclassified as salt. However, 97 plates were accurately categorized as non-salt, and 133 were correctly identified as salt (Figure 5B). The classification of plates using solely morphological attributes resulted in 3 misclassified plates and 96 correctly classified as non-salt. However, 60 plates were wrongly classified as non-salt, but 84 were correctly identified as salt (Figure 5C). For visible back light (VISBACK) with all attributes, the portal incorrectly grouped 26 plates as salt and 28 plates as non-salt. Nonetheless, 73 plates were accurately categorized as non-salt, and 116 were correctly identified as salt (Figure 5D).

In the color and morphological features of VISBACK images (Figures 5E, F), 76 plates were correctly classified, and 23 were misclassified as non-salt. Notably, 109 plates exhibited accurate salt classification when considering color attributes, surpassing the 86 plates correctly classified using morphological attributes. Conversely, there were 35 instances of misclassification for color attributes and 58 for morphological attributes. The classification of VISFRONT images was similar in the number of plates to that of VISBACK. However, when considering all attributes, VISFRONT achieved higher accuracy in classifying 5 more plates as non-salt but was 13 plates less accurate in classifying salt content. The classification results were identical to VISBACK when using only color attributes. In the case of morphological attributes, VISFRONT outperformed VISBACK by accurately classifying 2 more plates as salt but underperformed by 9 plates in the non-salt classification (Figures 5G–I).

Table 5 provides an overview of the five selected metrics employed to evaluate the portal's performance across all sets, using a concentration level of 0 and 200 mM (0-200nM). When utilizing just one training plate, the FLUO analysis achieved impressive results, with an accuracy of 0.93, a sensitivity of 0.90, a specificity of 0.96, a precision of 0.97, and an F1 score of 0.93 across all attributes. In comparison, the color feature subset yielded slightly higher results, with an accuracy of 0.94, a sensitivity of 0.92, a specificity of 0.97, a precision of 0.98, and an F1 score of 0.95. On the other hand, the morphological subset exhibited metrics of 0.74, 0.58, 0.96, 0.96, and 0.72, respectively.

For VISBACK with all attributes, the system achieved an accuracy of 0.77, a sensitivity of 0.80, a specificity of 0.73, a precision of 0.81, and an F1 score of 0.81. In contrast, the color and morphological tests generated results of 0.76, 0.77, 0.77, 0.83, and 0.79, as well as 0.67, 0.60, 0.77, 0.79, and 0.60, respectively.

10.3389/fpls.2023.1303429

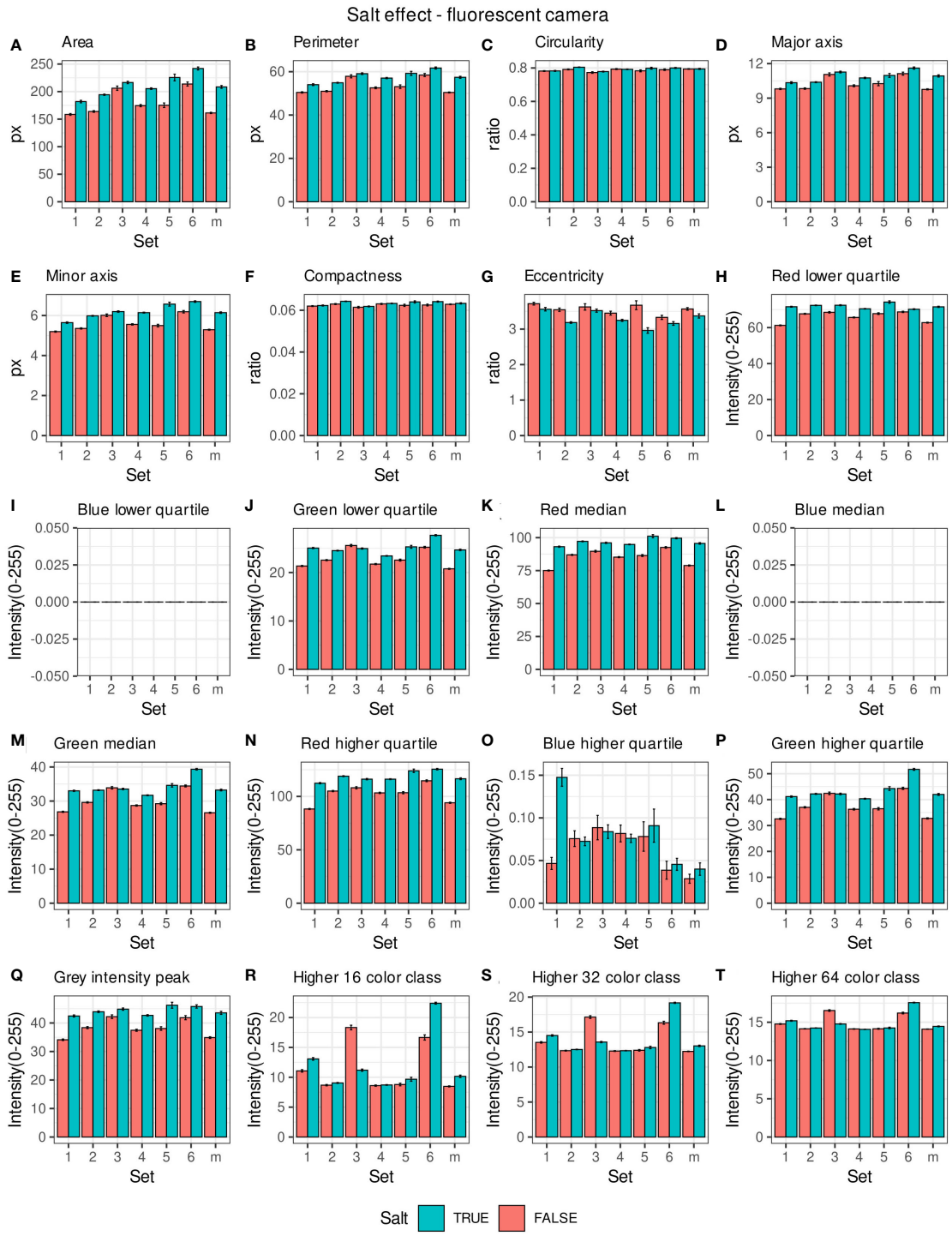


FIGURE 2
Morpho-colorimetric features from the back light visible light camera. Means and SEMs of the morpho-colorimetric features under normal and salt conditions for the 6 sets as well as the mix set (m). **(A)** Area, **(B)** Perimeter, **(C)** Circularity, **(D)** Major axis, **(E)** Minor axis, **(F)** Compactness, **(G)** Eccentricity, **(H)** Red lower quartile, **(I)** Blue lower quartile, **(J)** Green lower quartile, **(K)** Red median, **(L)** Blue median, **(M)** Green median, **(N)** Red higher quartile, **(O)** Blue higher quartile, **(P)** Green higher quartile, **(Q)** Grey Intensity peak, **(R)** Higher 16 color class, **(S)** Higher 32 color class, **(T)** Higher 64 color class.

When assessing VISFRONT, considering all attributes, an accuracy of 0.76, a sensitivity of 0.72, a specificity of 0.83, a precision of 0.86, and an F1 score of 0.78 were achieved. Using only the color attributes, the results were 0.76, 0.76, 0.77, 0.83 and 0.79. Meanwhile, employing only the morphological attributes yielded scores of 0.64, 0.61, 0.68, 0.73 and 0.67, respectively.

These findings suggest that the FLUO analysis outperforms VISBACK, and in turn, VISBACK outperforms VISFRONT.

10.3389/fpls.2023.1303429

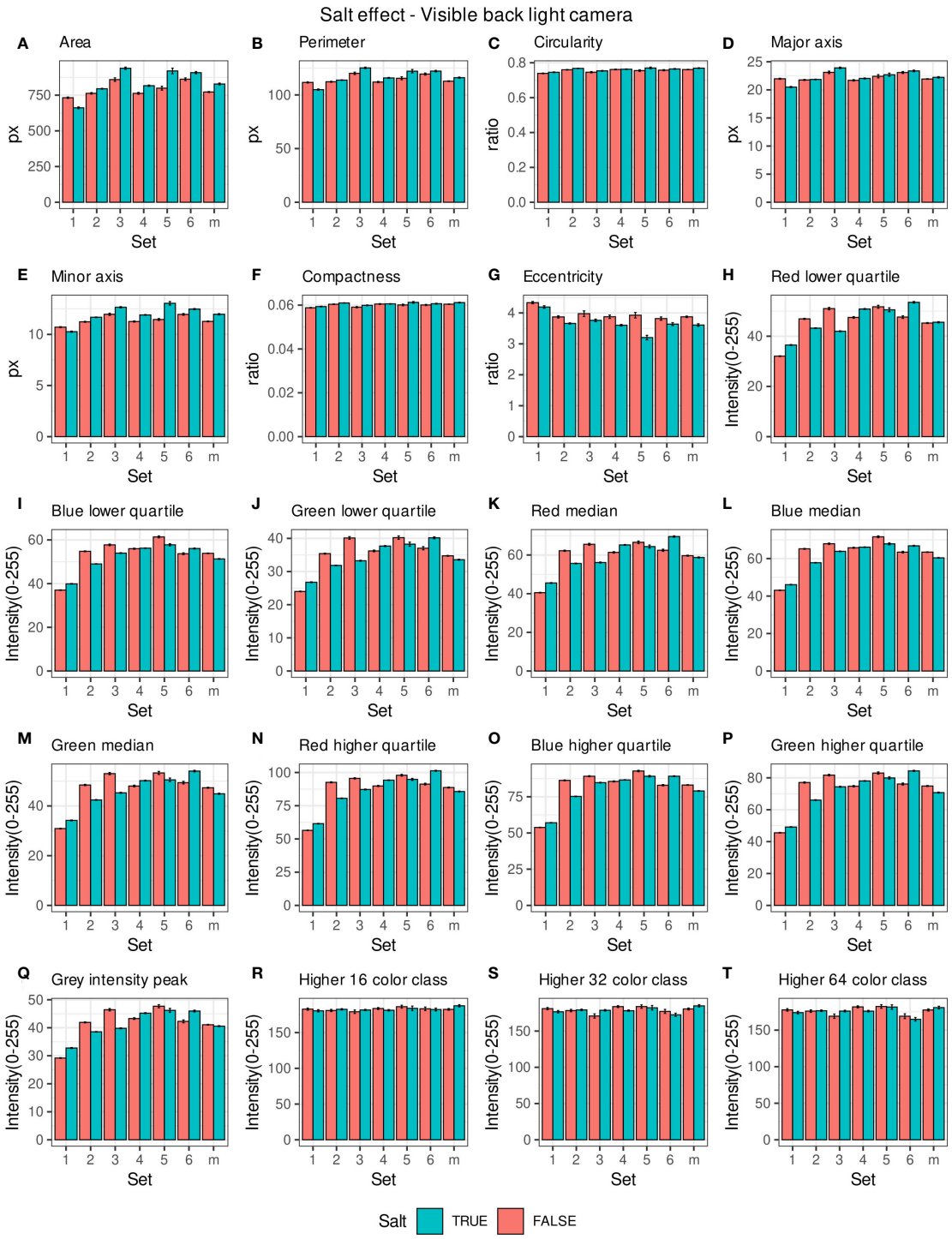


FIGURE 3
Morpho-colorimetric features from the back light visible light camera. Means and SEMs of the morpho-colorimetric features under normal and salt conditions for the 6 sets as well as the mix set (m). **(A)** Area, **(B)** Perimeter, **(C)** Circularity, **(D)** Major axis, **(E)** Minor axis, **(F)** Compactness, **(G)** Eccentricity, **(H)** Red lower quartile, **(I)** Blue lower quartile, **(J)** Green lower quartile, **(K)** Red median, **(L)** Blue median, **(M)** Green median, **(N)** Red higher quartile, **(O)** Blue higher quartile, **(P)** Green higher quartile, **(Q)** Grey Intensity peak, **(R)** Higher 16 color class, **(S)** Higher 32 color class, **(T)** Higher 64 color class.

Moreover, it becomes evident that color attributes exhibit greater effectiveness than morphological attributes in accurately predicting the salt status of seeds in plates.

When three training plates were used in FLUO (as presented in Table 5), the five metrics consistently demonstrated values ranging from 0.96 to 1, whether considered across all sets collectively or individually. The lowest recorded value, which was 0.96, occurred in accuracy and sensitivity for set 6, and in the F1 score for set 1. These results indicate a near 100% effectiveness in detection.

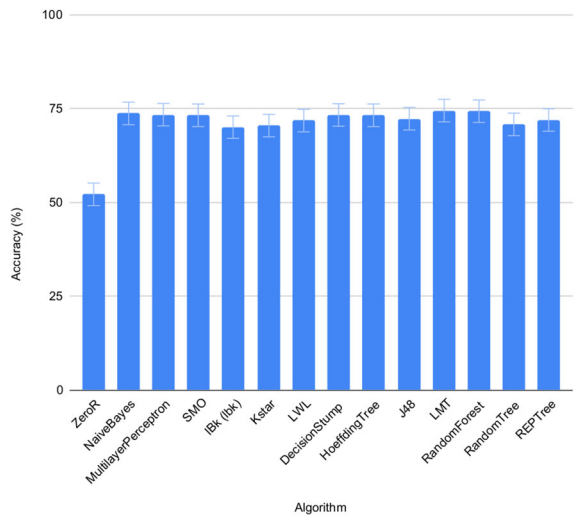


FIGURE 4
Performance of machine learning algorithms on individual seeds. Mean accuracy and SEMs for selected machine learning algorithms in set 1 and 2 under normal and salt conditions shown together.

3.5 Portal performance inside sets using other salt concentrations

To evaluate the performance of the portal and the type of camera (fluorescent or visible light) across various concentrations, sets 2 and 4 were tested using 50 mM, 100 mM and 150 mM in addition to 200 mM of salt versus non-salt under both fluorescent (FLUO) and visible backlight (VISBACK) images. The performance metrics are presented in Tables 6 and 7.

For the 0-200 mM concentrations, employing all attributes resulted in an accuracy of 0.95, a sensitivity of 0.90, a specificity and precision of 1, and an F1 score of 0.95, with a Fisher's exact test p-value lower than 2.2e-16. When testing at 0-150 mM, the metrics displayed an accuracy of 0.72, a sensitivity of 0.61, a specificity of 0.80, a precision of 0.73, and an F1 score of 0.67, along with a p-value of 1.815e-4. In the case of 0-100 mM, the performance metrics indicated an accuracy of 0.77, a sensitivity of 0.51, a specificity and precision of 1, and an F1 score of 0.67, with a p-value of 4.257e-06. For the 0-50 mM tests using all attributes, the results included an accuracy of 0.64, a sensitivity of 0.75, a specificity of 0.54, a precision of 0.59, an F1 score of 0.65, and a p-value of 0.01.

When considering only the color attributes, the results for 0-200 mM included an accuracy of 0.94, a sensitivity of 0.88, a specificity and precision of 1, an F1 score of 0.93, and a p-value lower than 2.2e-16. For 0-150 mM, the values were 0.80, 0.71, 0.88, 0.83, 0.77, and a p-value of 9.294e-08. For 0-100 mM, the results were 0.75, 0.48, 1, 1, 0.66, and a p-value of 4.551e-08. In the case of 0-50 mM, the values were 0.56, 0.50, 0.61, 0.52, 0.51, and no significant p-value was observed.

When using only the morphological attributes for 0-200 mM, an accuracy of 0.94, a sensitivity of 0.88, a specificity and precision of 1, an F1 score of 0.88, and a Fisher's exact test p-value lower than 2e-16 were achieved. In the 0-150 mM group, the metrics were 0.74, 0.86, 0.64, 0.67, 0.75, and the p-value was 7.432e-06. For the 0-100 mM and 0-50 mM groups, the values obtained were 0.74, 0.58, 0.88, 0.82, 0.69, and 1.498e-05, and 0.53, 0.94, 0.19, 0.50, 0.65, and 0.098, respectively.

Table 7 displays the performance metrics for VISBACK in sets 2 and 4. When considering all attributes in the 0-200 mM concentration range, the metrics included an accuracy of 0.71, a sensitivity of 0.90, a specificity of 0.52, a precision of 0.66, and an F1 score of 0.76, with a Fisher's exact test p-value of 3.601e-05. For the 0-150 mM tests, the metrics showed results of 0.73 for accuracy, 0.67 for sensitivity, 0.79 for specificity, 0.73 for precision, and an F1 score of 0.70, with a p-value of 7.798e-05. In the case of 0-100 mM and 0-50 mM, the metrics values were 0.47, 0.56, 0.38, 0.46, and 0.50, and 0.42, 0.63, 0.23, 0.42, and 0.51, respectively. In both cases, the p-values were not significant.

When using only the color attributes, the performance metrics at the 0-200 mM concentrations were as follows: an accuracy of 0.74, a sensitivity of 0.69, a specificity of 0.78, a precision of 0.76, and an F1 score of 0.72. In the 0-150 mM group, the metrics displayed values of 0.73, 0.77, 0.69, 0.68, and 0.78, respectively. For 0-100 mM and 0-50 mM, the metrics indicated 0.49, 0.52, 0.43, 0.48, and 0.51, and 0.60, 0.83, 0.40, 0.55, and 0.66, respectively. Notably, only 0-200 mM and 0-150 mM presented significant p-values ($p < 0.01$). The performance metrics when considering only the morphological attributes exhibited an accuracy of 0.82, a sensitivity of 0.64, a specificity and precision of 1, and an F1 score of 0.78. In the 0-150 mM group, the values were 0.60, 0.63, 0.80, 0.61, and 0.46. For 0-100 mM, the metrics indicated values of 0.61, 0.36, 0.36, 0.86, and 0.47, and for 0-50 mM, the values were 0.44, 0.44, 0.45, 0.41, and 0.42. Only the 0-200 mM group showed a significant p-value ($p < 0.01$).

3.6 K-fold validation portal performance among groups

The performance of the portal and the type of sensor was performed using the k-fold validation technique which is normally used to test machine learning algorithms with a k equal to 10 (Sakeef et al., 2023), on fluorescent images (FLUO). The salt concentration chosen was 0-200 mM since it is present in all the sets. Out of 93 plates, 30 were well classified as non-salt and 51 as salt against 9 misclassified as salt and 3 as non-salt for all attributes (Figure 6A). Using only the color attributes, 33 and 52 were well classified as non-salt and salt and 6 and 2 misclassified as salt and non-salt (Figure 6B). In the case of only morphological attributes, 29 and 50 were well classified against 10 and 4 respectively (Figure 6C).

An accuracy of 0.87 was attained using all attributes, accompanied by a sensitivity of 0.94, a specificity of 0.76, a precision of 0.85, and an F1 score of 0.89. When exclusively employing color attributes, an accuracy of 0.91 was achieved, along with a sensitivity of 0.96, a specificity of 0.84, a precision of 0.90, and an F1 score of 0.93. In the case of using only morphological attributes, results included an accuracy of 0.84, a sensitivity of 0.90, a specificity of 0.74, a precision of 0.83, and an F1 score of 0.88. Significance ($p < 0.01$) in all cases was shown using Fisher's exact test (Table 8).

TABLE 4 Description of the machine learning algorithms.

Classifier	Description	Reference weka packages
ZeroR	A rule algorithm that predicts the majority class in case of normal data or the average value.	https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/ZeroR.html
NaiveBayes	Implements a standard probabilistic naive Bayes algorithm using estimator classes.	https://weka.sourceforge.io/doc.dev/weka/classifiers/bayes/NaiveBayes.html
MultilayerPerceptron	Implements a type of artificial neural network algorithm which can be expressed as standard mathematical functions.	https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/MultilayerPerceptron.html
SMO	Sequential minimal optimization. This class implements a support vector classification that can be expressed as standard mathematical functions.	https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/SMO.html
IBk (lbk)	Implements a k-nearest-neighbour classification algorithm.	https://weka.sourceforge.io/doc.dev/weka/classifiers/lazy/IBk.html
Kstar	Implements the nearest neighbour algorithm with a generalized distance function.	https://weka.sourceforge.io/doc.dev/weka/classifiers/lazy/KStar.html
LWL	Implements a general algorithm for locally weighted learning.	https://weka.sourceforge.io/doc.dev/weka/classifiers/lazy/LWL.html
DecisionStump	Implements a decision tree using only one level for splitting.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/DecisionStump.html
HoeffdingTree	Implements a Hoeffding tree algorithm.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/HoeffdingTree.html
J48	Implements a C4.5 decision tree learner algorithm.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html
LMT	Logistic model trees. It builds classification trees with regression functions at their leaves.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/LMT.html
RandomForest	Implements the algorithm for building a forest of random trees.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomForest.html
RandomTree	Given a number of random features for each node, this class builds a tree without pruning.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomTree.html
REPTree	Implements a fast tree learning that reduces the error pruning.	https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/REPTree.html

(Witten et al., 2011; Smith and Frank, 2016).

TABLE 5 Performance descriptors within groups in 0 versus 200mM.

Set	Camera	Attributes	Training Plates*	Accuracy Equation 1	Sensitivity Equation 2	Specificity Equation 3	Precision Equation 4	F1 score Equation 5
1-6	Fluo	All	1	0.93	0.90	0.96	0.97	0.93
1-6	Fluo	Color	1	0.94	0.92	0.97	0.98	0.95
1-6	Fluo	Morpho	1	0.74	0.58	0.96	0.96	0.72
1-6	VisBack	All	1	0.77	0.80	0.73	0.81	0.81
1-6	VisBack	Color	1	0.76	0.77	0.77	0.83	0.79
1-6	VisBack	Morpho	1	0.67	0.60	0.77	0.79	0.60
1-6	VisFront	All	1	0.76	0.72	0.83	0.86	0.78
1-6	VisFront	Color	1	0.76	0.76	0.77	0.83	0.79
1-6	VisFront	Morpho	1	0.64	0.61	0.68	0.73	0.67
1-6	Fluo	All	3	0.98	0.97	1	1	0.98
1	Fluo	All	3	0.97	0.93	1	1	0.96

(Continued)

TABLE 5 Continued

Set	Camera	Attributes	Training Plates*	Accuracy Equation 1	Sensitivity Equation 2	Specificity Equation 3	Precision Equation 4	F1 score Equation 5
2	Fluo	All	3	1	1	1	1	1
3	Fluo	All	3	0.98	0.97	1	1	0.98
4	Fluo	All	3	1	1	1	1	1
5	Fluo	All	3	1	1	1	1	1
6	Fluo	All	3	0.96	0.96	1	1	0.98

* Training plates per condition group. (Fisher's exact test p<0.01 for all cases). Fluo, fluorescent images; VisBack, Visible back light images; VisFront, Visible top light images.

3.7 Alternative applications of SeedML

To assess the usability of the portal for working with various types of data, a series of side-view images of Camelina plants were captured and analyzed using this portal. The parameters for quantifying and qualifying pods per plant were adjusted through the user interface section "seed detection setup". Manual counting was also completed to evaluate performance. The strength of the relationship was assessed using the Pearson coefficient (r=0.90), revealing a strong positive correlation (Figure 7).

4 Discussion

The morpho-colorimetric seed features using the fluorescent light images displayed a greater sensitivity to salt than the visible light images (Figures 2, 3). In fact, the area-related features showed higher values in the fluorescent images under salt conditions as well as the lower, median and higher quartiles of the red intensity value. This may be explained by the fluorescence emission intensity which increases with the increase in concentration of salt (Adenier et al., 1998; Sharma et al., 2018). A variation in the morpho-colorimetric seed features was also observed among the sets. This variation may be attributed to differences in the chemical composition of seed oil (Dogruer et al., 2021), which could be influenced by variations in growing conditions, including watering regimes. It has been shown that seed oil content can change in response to factors such as nitrogen fertilizer, suggesting that soil content including the prevalence of salts may play a key role in seed oil composition (Li et al., 2017).

The conversion from pixels to the metric system is important not only for the purpose of comparing and sharing information, as it does not depend on the image, but also for validating the results of seed detection. This feature is included in the portal. We used the measurements of the plate in both cameras to calculate the conversion and we compared seeds manually measured using a ruler (Supplementary Figure 1). Our manual observation and pixel-converted calculation both yielded a length of 2 mm, which aligns with the measurements reported by Francis and Warwick (2009). Additionally, the portal calculated a width of 1 mm, half of the

TABLE 6 Performance descriptors within groups in set 2 and 4 using one training plate for each condition group under fluorescent light images (FLUO).

Concentration	Attributes	Accuracy Equation 1	Sensitivity Equation 2	Specificity Equation 3	Precision Equation 4	F1 score Equation 5	p-value*
0-200nM	All	0.95	0.90	1	1	0.95	< 2.2e-16
0-150mM	All	0.72	0.61	0.80	0.73	0.67	1.815e-4
0-100mM	All	0.77	0.51	1	1	0.67	4.257e-06
0-50mM	All	0.64	0.75	0.54	0.59	0.65	0.01098
0-200mM	Color	0.94	0.88	1	1	0.93	< 2.2e-16
0-150mM	Color	0.80	0.71	0.88	0.83	0.77	9.294e-08
0-100mM	Color	0.75	0.48	1	1	0.66	4.551e-08
0-50mM	Color	0.56	0.50	0.61	0.52	0.51	0.3616
0-200mM	Morpho	0.94	0.88	1	1	0.88	< 2e-16
0-150mM	Morpho	0.74	0.86	0.64	0.67	0.75	7.432e-06
0-100mM	Morpho	0.74	0.58	0.88	0.82	0.69	1.498e-05
0-50mM	Morpho	0.53	0.94	0.19	0.50	0.65	0.09707

* Fisher's exact test.

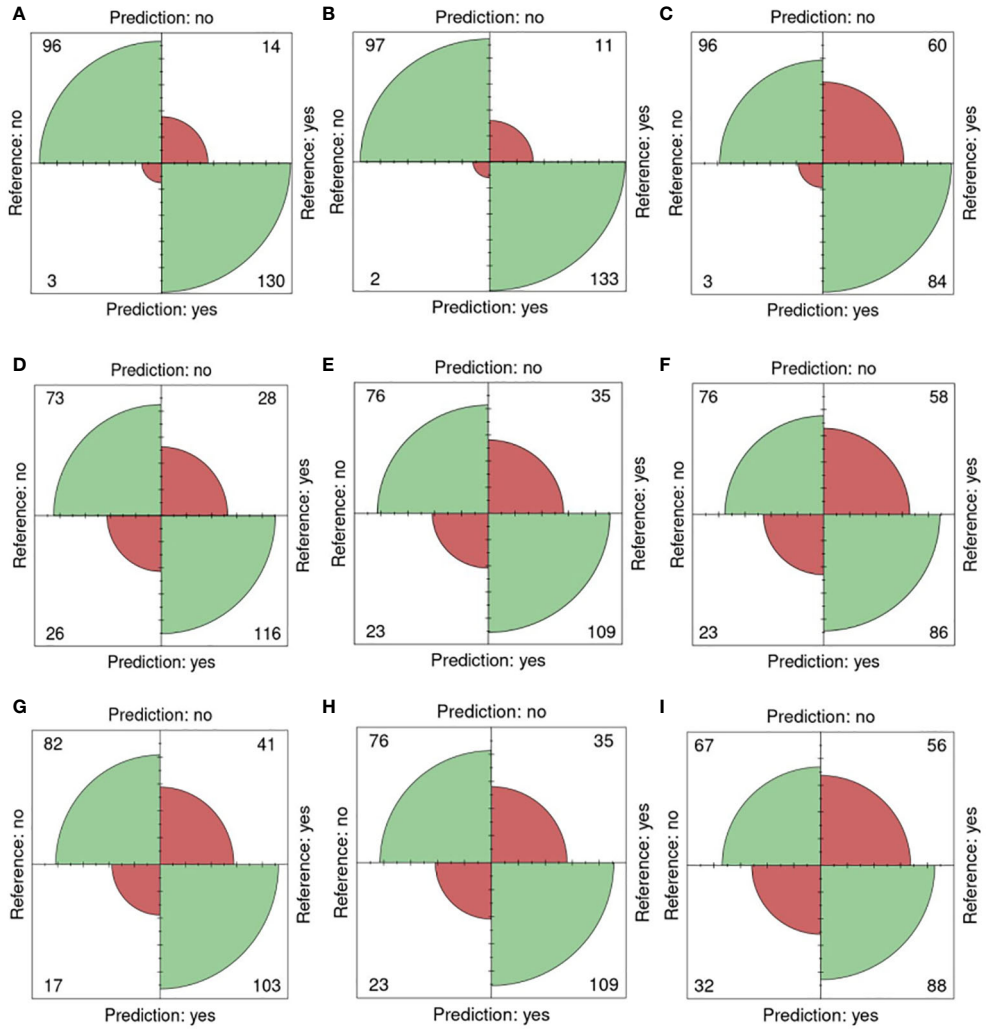


FIGURE 5
Confusion matrices for 0 and 200mM. Reference (real) versus prediction plots for sets 1 through 6 using one training plate for each condition (salt/non-salt) and set with n=3 for. **(A)** FLUO, all attributes. **(B)** FLUO, color attributes. **(C)** FLUO, morphological attributes. **(D)** VISBACK, all attributes. **(E)** VISBACK, color attributes. **(F)** VISBACK, morphological attributes. **(G)** VISFRONT, all attributes. **(H)** VISFRONT, color attributes. **(I)** VISFRONT, morphological attributes. (FLUO, Fluorescent images; VISBACK, visible back light images; VISFRONT, visible top light images).

length, in line with the findings of Fleenor (2011). An amount of 1000 seeds weighs between 0.8 to 2.0 g (Ehrensing et al., 2008), meaning that the number of seeds expected in 0.1 g is in the range of 50 to 125 seeds which has been corroborated in our analysis with an average of 92, 84 and 95 normally distributed between 60-140.

The machine learning algorithms evaluated on the classification of individual seeds were taken from the WEKA package (Frank et al., 2016), namely ZeroR, NaiveBayes, MultilayerPerceptron, SMO, IBk, Kstar, LWL, DecisionStump, HoeffdingTree, J48, LMT, RandomForest, RandomTree and REPTree (Figure 1, Table 4). All of them show an accuracy equal or greater than 70% except for the ZeroR which showed an accuracy of 52% (Figure 4). For this reason, the ZeroR algorithm was not implemented in the portal since it did not significantly contribute to the classification process.

The consensus achieved by the machine learning algorithms analyzing morpho-colorimetric features in the image analysis process, in conjunction with the universally accessible user-friendly web interface and a wide range of customizable parameters, endows the portal with exceptional performance. The outputs may be tailored to accommodate various types of images, to inform on a wide range of data sets. Most of the analyses were conducted using a different plate for training in each group or set, as it represents the minimum information that can be provided. However, a three-plate training approach was implemented to uphold this principle. The best performance, achieved using the one-plate training method, was observed in the case of the fluorescent light images, with scores of 90% or higher in all five effectiveness metrics. This was followed by the visible light back images and then by the visible light top images. In the case of three-plate training, almost 100% classification performance was obtained in the five metrics (Figure 5, Table 5). This demonstrates the robustness of the algorithms implemented in the portal, as well as the effect of salt on fluorescent light reflectance (Adenier et al., 1998; Sharma et al., 2018). Furthermore, utilizing color attributes alone resulted in an overperformance compared to using only morphological attributes (Table 5).

TABLE 7 Performance descriptors within groups in set 2 and 4 using one training plate for each condition group under visible back light images (VISBACK).

Concentration	Attributes	Accuracy Equation 1	Sensitivity Equation 2	Specificity Equation 3	Precision Equation 4	F1 score Equation 5	p-value*
0-200mM	All	0.71	0.90	0.52	0.66	0.76	3.601e-05
0-150mM	All	0.73	0.67	0.79	0.73	0.70	7.798e-05
0-100mM	All	0.47	0.56	0.38	0.46	0.50	0.6561
0-50mM	All	0.42	0.63	0.23	0.42	0.51	0.3616
0-200mM	Color	0.74	0.69	0.78	0.76	0.72	7.328e-06
0-150mM	Color	0.73	0.77	0.69	0.68	0.78	4.174e-05
0-100mM	Color	0.49	0.52	0.43	0.48	0.51	1
0-50mM	Color	0.60	0.83	0.40	0.55	0.66	0.0264
0-200mM	Morpho	0.82	0.64	1	1	0.78	2.628e-11
0-150mM	Morpho	0.60	0.36	0.80	0.61	0.46	0.1251
0-100mM	Morpho	0.61	0.36	0.36	0.86	0.47	0.03802
0-50mM	Morpho	0.44	0.44	0.45	0.41	0.42	0.4959

* Fisher's exact test.

The reduction in salt concentration resulted in a decrease in the effectiveness of the classification. This effect was observed in two sets of fluorescent light images where lower concentrations were available (Table 6). This finding supports the influence of salt on fluorescent reflectance and may indicate a lower concentration of salt within the seeds when grown in less saline soils. In 0-200 mM, the F1 score is 0.95 compared to 0.65 in 0-50mM. This may represent a correlation between the seed salt content and the fluorescent seed reflectance.

The k-fold validation is a widely used method to estimate the performance of machine learning algorithms on many performance indicators, in this case, accuracy, sensitivity, specificity, precision and F1 score (Refaeilzadeh et al., 2009). A k value equal to 10 was used since it is the most acceptable value for testing these kinds of algorithms (Refaeilzadeh et al., 2009; Sakeef et al., 2023). The 0-200 mM concentrations were selected from sets 1 to 6 (Figure 6, Table 8). This allows us to test the performance of the prediction process among groups growing in different conditions using fluorescent light images. Surprisingly, an accuracy of 0.87 and 0.91 was achieved with all and color attributes only and a sensitivity of 0.94 and 0.96 respectively even though the fluorescent reflectance is also affected by the oil composition which is affected by the growing conditions (Boschi et al., 2011; Li et al., 2017; Cober and Malcolm, 2019; Dogruer et al., 2021).

The SeedML portal offers a versatile solution for addressing various phenotypic questions using plant images. As an illustrative case, this research showcases the automated counting of pods in side-view images of Camelina. This data is crucial for evaluating

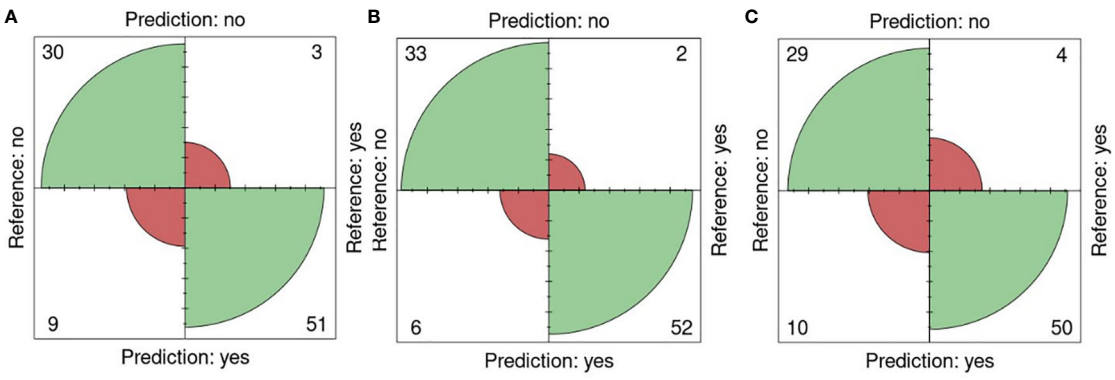


FIGURE 6
K-fold validation confusion matrices. Reference (real) versus prediction plots among groups using one training plate for each condition (salt/nonsalt) with a k=10 for 0 and 200mM. **(A)** All attributes, **(B)** Color attributes, **(C)** Morphological attributes.

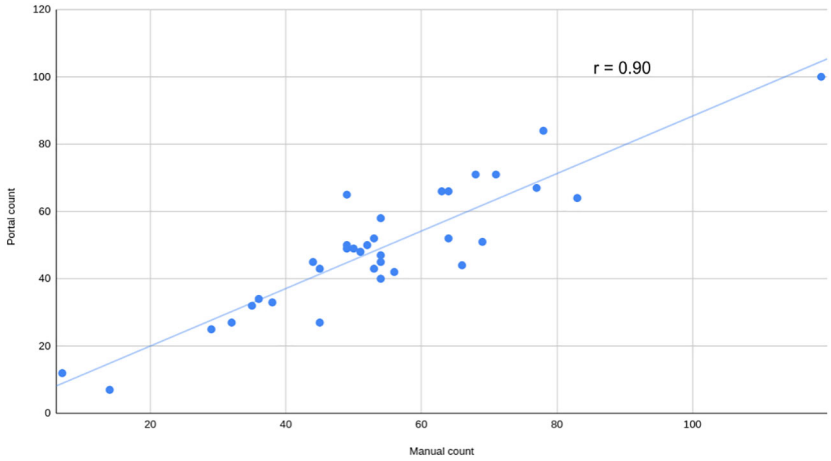


FIGURE 7
Manual versus automatic count of pods per plant. Manual count of pods versus automatically calculated by the portal using visible light side view images from a mix of Camelina plants grown in salt and non-salt condition at 52 days after sowing. The Pearson correlation coefficient (r) is shown.

TABLE 8 Performance descriptors of k-fold validation tests among groups using 0 and 200mM from fluorescent images (FLUO).

Attributes	Accuracy Equation 1	Sensitivity Equation 2	Specificity Equation 3	Precision Equation 4	F1 score Equation 5	p-value*
All	0.87	0.94	0.76	0.85	0.89	3.339e-13
Color	0.91	0.96	0.84	0.90	0.93	< 2.2e-16
Morpho	0.84	0.93	0.74	0.83	0.88	1.289e-11

* Fisher's exact test.

yield production and would otherwise demand significant human resources and time if handled manually. In this case, achieving the objective was accomplished by simply adjusting parameters through the user interface. A high Pearson correlation coefficient (r = 0.90) was obtained, indicating the effectiveness of this analysis. It should be noted that this was just one illustrative example and the SeedML portal can be used to perform a wide range of image-based phenotyping analyses.

In this study, the capability of combining fluorescent and visible light images with image analysis and machine learning algorithms to assess the color-morphological characteristics of Camelina seeds to predict the soil's salinity status has been demonstrated. An easy to navigate portal was devised and designed to be accessible to individuals with minimal computer skills and compatible with any device, including smartphones. The utility of the portal in addressing other phenomics analyses along with its implications in oil assessment and quality control have been illustrated. The findings of this research may positively inform related studies in the context of agricultural innovation and related fields such as animal feed production, in response to climate change. SeedML may further aid in the development and implementation of new quality control tools within the agri-food industry, enhancing productivity and sustainability in the manufacturing process.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

EV: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. ML: Conceptualization, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. JA: Conceptualization, Investigation, Writing – review & editing. TB: Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This project was funded by grants from Natural Sciences and Engineering Research Council (NSERC) of Canada [funding reference numbers: RGPIN-2016-05439 and STPGP 506642-17] and

Canada Foundation for Innovation (CFI) [funding reference number: 28991] to TB.

Acknowledgments

We would like to thank Lea Collin for contributing code. We also thank Mahnaz Mansoori for her help with the greenhouse rooms.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1303429/full#supplementary-material

References

Adenier, A., Duville, F., and Aaron, J. J. (1998). Effects of various salts on the spectral properties of merocyanine 540, a fluorescent probe, in aqueous media. *Proc. Indian Acad. Sciences: Chem. Sci.* 110 (3), 311–317. doi: 10.1007/BF02870009

Berti, M., Gesch, R., Eynck, C., Anderson, J., and Cermak, S. (2016). Camelina uses, genetics, genomics, production, and management. *Ind. Crops Products* 94, 690–710. doi: 10.1016/j.indcrop.2016.09.034

Boschi, F., Fontanella, M., Carderan, L., and Sbarbati, A. (2011). Luminescence and fluorescence of essential oils. Fluorescence imaging in *vivo* of wild chamomile oil. *Eur. J. histochemistry: EJH* 55 (2), 97–100. doi: 10.4081/ejh.2011.e18

Burger, W., and Burge, M. (2008, 2016). *Digital Image Processing. An Algorithmic Introduction Using Java. 2nd ed.* (London, UK: Springer-Verlag London).

Camargo, A., Papadopoulou, D., Spyropoulou, Z., Vlachonasios, K., Doonan, J. H., and Gay, A. P. (2014). Objective definition of rosette shape variation using a combined computer vision and data mining approach. *PloS One* 9 (5), e96689. doi: 10.1371/journal.pone.0096889

Charilaou, P., and Battat, R. (2022). Machine learning models and over-fitting considerations. *World J. Gastroenterol.* 28 (5), 605–607. doi: 10.3748/wjg.v28.i5.605

Cober, E. R., and Malcolm, M. (2019). Soybean yield and seed composition changes in response to increasing atmospheric CO2 concentration in short-season Canada. *Plants* 8 (8), 250. doi: 10.3390/plants8080250

Dogruer, I., Uyar, H., Uncu, O., and Ozen, B. (2021). Prediction of chemical parameters and authentication of various cold pressed oils with fluorescence and mid-infrared spectroscopic methods. *Food Chem.* 345, 1–12. doi: 10.1016/j.foodchem.2020.128815

Ehrensing, D. T., Guy, S. O.Extension Service, Oregon State University (2008) *Camelina*. Available at: https://ir.library.oregonstate.edu/concern/open_educational_resources/n583xv355.

Fleenor, R. (2011). *Plant Guide for Camelina (Camelina sativa)* ((Spokane, WA, USA: USDA-Natural Resources Conservation Service).

Francis, A., and Warwick, S. I. (2009). The Biology of Canadian Weeds. 142. Camelina alyssum (Mill.) Thell.; C. microcarpa Andrz. ex DC.; Camelina sativa (L.) Crantz. *Can. J. Plant Sci.* 89, 791–810. doi: 10.4141/CJPS08185

Frank, E., Hall, M. A., and Witten, I. H. (2016). "The WEKA workbench. Online appendix," in *Data Mining: Practical Machine Learning Tools and Techniques*, *4th ed.* (New York, USA: Morgan Kaufmann).

Hassani, A., Azapagic, A., and Shokri, N. (2021). Global predictions of primary soil salinization under changing climate in the 21st century. *Nat. Commun.* 12 (1), 6663. doi: 10.1038/s41467-021-26907-3

Joly-Lopez, Z., Forczek, E., Vello, E., Hoen, D. R., Tomita, A., and Bureau, T. E. (2017). Abiotic stress phenotypes are associated with conserved genes derived from transposable elements. *Front. Plant Sci.* 8 (November). doi: 10.3389/fpls.2017.02027

Kagale, S., Koh, C., Nixon, J., Bollina, V., Clarke, W. E., Tuteja, R., et al. (2014). The emerging biofuel crop Camelina sativa retains a highly undifferentiated hexaploid genome structure. *Nat. Commun.* 5 (3706), 1–11. doi: 10.1038/ncomms4706

Li, W. P., Shi, H. B., Zhu, K., Zheng, Q., and Xu, Z. (2017). The quality of sunflower seed oil changes in response to nitrogen fertilizer. *Agron. J.* 109 (6), 2499–2507. doi: 10.2134/agronj2017.01.0046

Masella, P., Martinelli, T., and Galasso, I. (2014). Agronomic evaluation and phenotypic plasticity of Camelina sativa growing in Lombardia, Italy. *Crop Pasture Sci.* 65 (5), 453–460. doi: 10.1071/CP14025

Morales, D., Potlakayala, S., Soliman, M., Daramola, J., Weeden, H., and Jones, A. (2017). Effect of biochemical and physiological REsponse to salt stress in camelina sativa. *Commun. Soil Sci. Plant Anal.* 48 (7), 716–729. doi: 10.1080/00103624.2016.1254237

Prusty, S., Patnaik, S., and Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Front. Nanotechnology* 4. doi: 10.3389/fnano.2022.972421

Razzaq, A., Wani, S. H., Saleem, F., Yu, M., Zhou, M., and Shabala, S. (2021). Rewilding crops for climate resilience: Economic analysis and *de novo* domestication strategies. *In J. Exp. Bot.* 72 (18), 6123–6139. doi: 10.1093/jxb/erab276

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. *Encyclopedia Database Syst.* 4210, 532–538. doi: 10.1007/978-0-387-39940-9_565

Saharan, S. S., Nagar, P., Creasy, K. T., Stock, E. O., Feng, J., Malloy, M. J., et al. (2021). Machine learning and statistical approaches for classification of risk of coronary artery disease using plasma cytokines. *BioData Min.* 14 (1), 1–14. doi: 10.1186/s13040-021-00260-z

Sakeef, N., Scandola, S., Kennedy, C., Lummer, C., Chang, J., Uhrig, R. G., et al. (2023). Machine learning classification of plant genotypes grown under different light conditions through the integration of multi-scale time-series data. *Comput. Struct. Biotechnol. J.* 21, 3183–3195. doi: 10.1016/j.csbj.2023.05.005

Sarkar, S., Zhou, J., Scaboo, A., Zhou, J., Aloysius, N., and Lim, T. T. (2023). Assessment of soybean lodging using UAV imagery and machine learning. *Plants* 12 (2893), 1–20. doi: 10.3390/plants12162893

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* 9 (7), 676–682. doi: 10.1038/nmeth.2019

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* 9, 671–675. doi: 10.1038/nmeth.2089

Shah, T., Xu, J., Zou, X., Cheng, Y., Nasir, M., and Zhang, X. (2018). Omics approaches for engineering wheat production under abiotic stresses. *Int. J. Mol. Sci.* 19 (8), 2390. doi: 10.3390/ijms19082390

Sharma, A., Bueno, D., Bhand, S., Marty, J. L., and Muñoz, R. (2018). Evaluation of various factors affecting fluorescence emission behavior of ochratoxin A: effect of pH, solvent and salt composition. *Biomed. J. Sci. Tech. Res.* 10 (4), 4–9. doi: 10.26717/bjstr.2018.10.001979

Singh, A. (2021). Soil salinity: A global threat to sustainable development. *Soil Use Manage.* 38 (1), 39–67. doi: 10.1111/sum.12772

Smith, T. C., and Frank, E. (2016). Introducing machine learning concepts with WEKA. *Methods Mol. Biol.* 1418, 353–378. doi: 10.1007/978-1-4939-3578-9_17

Vello, E., Aguirre, J., Shao, Y., and Bureau, T. (2022). "Camelina sativa high-throughput phenotyping under normal and salt conditions using a plant phenomics platform," in *High-Throughput Plant Phenotyping: Methods and Protocols*. Eds. A. Lorence and K.M. Jimenez (New York, USA: Humana Press), 25–36.

Vello, E., Tomita, A., Diallo, A. O., and Bureau, T. E. (2015). A comprehensive approach to assess arabidopsis survival phenotype in water-limited condition using a

non-invasive high-throughput phenomics platform. *Front. Plant Sci* 6. doi: 10.3389/fpls.2015.01101

Vollmann, J., and Eynck, C. (2015). Camelina as a sustainable oilseed crop: contributions of plant breeding and genetic engineering. *Biotechnol. J.* 10 (4), 525–535. doi: 10.1002/biot.201400200

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques. 3rd ed.* (New York, USA: Elsevier).

Xu, Z., York, L. M., Seethepalli, A., Bucciarelli, B., Cheng, H., and Samac, D. A. (2022). Objective phenotyping of root system architecture using image augmentation and machine learning in alfalfa (Medicago sativa L.). *Plant Phenomics* 2022, 1–15. doi: 10.34133/2022/9879610

Yang, C., Baireddy, S., Méline, V., Cai, E., Caldwell, D., Iyer-Pascuzzi, A. S., et al. (2023). Image-based plant wilting estimation. *Plant Methods* 19 (52), 1–16. doi: 10.1186/s13007-023-01026-w

Zanetti, F., Alberghini, B., Marjanović Jeromela, A., Grahovac, N., Rajković, D., Kiprovski, B., et al. (2021). Camelina, an ancient oilseed crop actively contributing to the rural renaissance in Europe. A review. *Agron. Sustain. Dev.* 41 (2), 1–18. doi: 10.1007/s13593-020-00663-y/Published

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 11 January 2024
DOI 10.3389/fpls.2023.1292766

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Chunlei Xia,
Chinese Academy of Sciences (CAS), China
Jinrong He,
Yan'an University, China

*CORRESPONDENCE
Guoxu Liu
liuguoxu@wfu.edu.cn
Sungkyung Park
fspark@pusan.ac.kr

RECEIVED 12 September 2023
ACCEPTED 20 December 2023
PUBLISHED 11 January 2024

CITATION
Touko Mbouembe PL, Liu G, Park S and Kim JH (2024) Accurate and fast detection of tomatoes based on improved YOLOv5s in natural environments.
Front. Plant Sci. 14:1292766.
doi: 10.3389/fpls.2023.1292766

Accurate and fast detection of tomatoes based on improved YOLOv5s in natural environments

Philippe Lyonel Touko Mbouembe[1], Guoxu Liu[2,3]*, Sungkyung Park[1]* and Jae Ho Kim[1,4]

[1]Department of Electronics Engineering, Pusan National University, Busan, Republic of Korea, [2]School of Computer Engineering, Weifang University, Weifang, China, [3]R&D Center, Univalsoft Joint Stock Co., Ltd., Shouguang, China, [4]Exsolit Research Center, Yangsan, Republic of Korea

Uneven illumination, obstruction of leaves or branches, and the overlapping of fruit significantly affect the accuracy of tomato detection by automated harvesting robots in natural environments. In this study, a proficient and accurate algorithm for tomato detection, called SBCS-YOLOv5s, is proposed to address this practical challenge. SBCS-YOLOv5s integrates the SE, BiFPN, CARAFE and Soft-NMS modules into YOLOv5s to enhance the feature expression ability of the model. First, the SE attention module and the C3 module were combined to form the C3SE module, replacing the original C3 module within the YOLOv5s backbone architecture. The SE attention module relies on modeling channel-wise relationships and adaptive re-calibration of feature maps to capture important information, which helps improve feature extraction of the model. Moreover, the SE module's ability to adaptively re-calibrate features can improve the model's robustness to variations in environmental conditions. Next, the conventional PANet multi-scale feature fusion network was replaced with an efficient, weighted Bi-directional Feature Pyramid Network (BiFPN). This adaptation aids the model in determining useful weights for the comprehensive fusion of high-level and bottom-level features. Third, the regular up-sampling operator is replaced by the Content Aware Reassembly of Features (CARAFE) within the neck network. This implementation produces a better feature map that encompasses greater semantic information. In addition, CARAFE's ability to enhance spatial detail helps the model discriminate between closely spaced fruits, especially for tomatoes that overlap heavily, potentially reducing the number of merging detections. Finally, for heightened identification of occluded and overlapped fruits, the conventional Non-Maximum-Suppression (NMS) algorithm was substituted with the Soft-NMS algorithm. Since Soft-NMS adopts a continuous weighting scheme, it is more adaptable to varying object sizes, improving the handling of small or large fruits in the image. Remarkably, this is carried out without introducing changes to the computational complexity. The outcome of the experiments showed that SBCS-YOLOv5s achieved a mean average precision (mAP (0.5:0.95)) of 87.7%,

which is 3.5% superior to the original YOLOv5s model. Moreover, SBCS-YOLOv5s has a detection speed of 2.6 ms per image. Compared to other state-of-the-art detection algorithms, SBCS-YOLOv5s performed the best, showing tremendous promise for tomato detection in natural environments.

KEYWORDS

artificial intelligence, tomato detection, attention mechanism, BiFPN, YOLOv5, computer vision, agriculture

1 Introduction

The tomato is one of the world's most important crops (Peixoto et al., 2017), but harvesting tomatoes under natural conditions remains labor-intensive. Fruit harvesting has undergone a significant transformation through advances in artificial intelligence within laboratory research. This evolution has paved the way for the emergence of fruit-picking robots anticipated to supplant manual labor. The vision system plays a vital role in a fruit-picking robot. This is because of its fundamental role in accurately identifying fruits, a crucial initial step hinging on the robot's precision, efficiency, and resilience. Nevertheless, the challenges posed by natural conditions introduce complexities, such as unbalanced lighting, occlusions, overlapping, and other unforeseeable elements (Gongal et al., 2015), all of which affect the detection accuracy of fruit-picking robots. Consequently, enhancing the accuracy, efficiency, and robustness of harvesting robots under these natural conditions is essential.

Many researchers have studied fruit detection to deal with the problems mentioned above. Some digital image processing approaches, such as color features (Goel and Sehgal, 2015; Yang et al., 2020), shape (Jana and Pareskh, 2017), and texture (Rakun et al., 2011), have been proposed to obtain reasonable detection results. Zhao et al. (2016a) developed a technique for detecting immature citrus fruits in natural environments based on cascaded pixel segmentation. A combination of color feature maps and a block-matching method were employed to identify potential fruit pixels. Subsequently, further refinement is adopted utilizing an SVM classifier to eliminate false positives. On the other hand, in the initial stages of segmentation, by relying solely on color features, numerous fruit instances remain undetected due to the resemblance between green fruit and the background. Kurtulmus et al. (2011) introduced a new eigenfruit feature for identifying green citrus. This characteristic was paired with color information and a study of circular Gabor texture. Despite including the texture characteristics alongside color features, their method has encountered challenges distinguishing some fruit from the background and has struggled to detect heavily obscured fruit effectively.

Other methods include K-means clustering (Jiao et al., 2020), Support Vector Machine (SVM) (Azarmdel et al., 2020), and AdaBoost algorithms (Payne et al., 2014). In tomato detection, Liu et al. (2019) developed an approach to identify mature tomatoes within natural environments. A naive Bayesian classifier with an oriented gradient histogram was used to distinguish each tomato. Subsequently, a color analysis step was used to remove false positives. Nevertheless, adapting this method to natural settings posed a challenge owing to the inherent limitations of manually crafted features in terms of their capacity for high-level abstraction. Similarly, Zhao et al. (2016b) used Haar-like feature thresholding and AdaBoost classifier to detect tomatoes. Their study revealed a recognition rate of 96% for tomatoes within their testing dataset. Nevertheless, a long training time was required in their approach.

The aforementioned methods relying on manually designed features have inherent limitations, particularly in scenarios where intricate feature extraction is demanded. The introduction of deep learning successfully addressed these challenges. For example, Rahnemoofar and Sheppard (2017) showcased commendable fruit-counting capabilities through a customized Inception-ResNet architecture (Szegedy et al., 2017). On the other hand, this model focused exclusively on fruit counting and failed to detect them. Santo et al. (2020) introduced a method to detect and track grape clusters within images captured in vineyards. This approach utilized the Mask-RCNN algorithm (He et al., 2017) for the precise detection of individual grape bunches. Furthermore, structure from motion techniques were applied to achieve the 3D alignment of images, enabling the effective mapping of features across various images. Their method achieved an F1-score of 91%. In recent years, the emergence of YOLO models has revolutionized object detection (Redmon et al., 2016; Redmon and Farhadi, 2017; Redmon and Farhadi, 2018; Boschkovskiy et al., 2020; Jocher et al., 2020; Wang et al., 2022). These YOLO models exhibited exceptional improvement in accuracy and speed, surpassing traditional two-stage pipelines (He et al., 2017; Girshick et al., 2014; Ren et al., 2015). They used a single feed-forward network to detect bounding boxes and corresponding classes. Wang et al. (2021) introduced an innovative method anchored in an enhanced YOLOv3-tiny model to identify disease occlusion and overlapping tomato leaves. This model strategically mitigated information loss during network transmission, resulting in a commendable mAP score of 93.1%. Bresilla et al. (2019) used DCNN architectures based on single-stage detectors. Leveraging deep learning techniques eliminates the need to manually code specific features tailored to particular fruit shapes,

colors, or other attributes. This method achieved an accuracy of more than 90%. Liu et al. (2020) introduced YOLO-Tomato, a resilient model based on YOLOv3. This model achieved an Average Precision (AP) of 96.40% and a rapid detection speed of 54 ms. Chen et al. (2022) introduced a modified YOLOv4 to detect citrus. Their approach used an attention mechanism and a depth-wise separable convolution module. Moreover, they applied a pruning algorithm to eliminate the impact of irrelevant latent factors in the data. Their average improved from 92.89% to 96.15%, with 0.06s to detect each image.

Expanding the scope, Cao et al. (2023) proposed a technique for persimmon recognition in natural environments. They harnessed an enhanced YOLOv5 model, achieving an average accuracy of 95.53%. Mbouembe et al. (2023) proposed a modified YOLOv4-tiny model for tomato recognition. Their enhancements included a refined backbone design, reducing computational complexity while augmenting feature extraction. A simplified CSP (Cross-Stage Partial Connections) - Spatial Pyramid Pooling was incorporated to improve the receptive field of the backbone. This modification aimed to enhance the ability of the model to capture information from a wider area of the input data. The CARAFE module in the neck network further improved the quality of the feature map. Their method produced an mAP of 82.8%.

Despite extensive research in the fruit recognition domain within natural conditions, it is essential to improve the detection accuracy and robustness to fulfill the requirements of fruit detection. This study introduced a precise and resilient tomato detection methodology grounded in the YOLOv5s model to address these persisting challenges. Figure 1 provides a concise overview of the proposed SBCS-YOLOv5s. The pivotal modifications of this research are outlined as follows:

1. "C3SE Integration": By amalgamating the SE attention module and the C3 module into a cohesive C3SE module, the conventional C3 module within the YOLOv5s backbone network is upgraded. This integration augments the capacity of the model to provide useful information, bolstering feature extraction.
2. "Bi-directional Feature Pyramid Network Integration": The original multi-scale PANet feature fusion network is replaced with an efficient weighted Bi-directional Feature Pyramid Network. This alteration enhances feature propagation and reuse, thereby refining overall feature representation.
3. "CARAFE Module Adoption": Positioned within the network's neck, the CARAFE module is harnessed to generate an improved feature map enriched with more intricate semantic information.
4. "Soft-NMS Algorithm Implementation": A noteworthy shift occurs in the detection post-processing stage, where the conventional NMS algorithm yields to the enhanced Soft-NMS algorithm. This transition amplifies the capacity to identify overlapping and occluded fruit.
5. "Performance Evaluation": Rigorous evaluation using tomato datasets unveils that the proposed SBCS-YOLOv5s model surpasses the original YOLOv5s model and other contemporary update object-detection methods in terms of accuracy.

Focusing on these goals, the study aimed to contribute to advancing tomato harvesting robots by developing an accurate tomato detection model that outperforms existing models in terms of accuracy and efficiency.

2 Theoretical background

2.1 YOLOv5 network

The YOLOv5s model (Jocher et al., 2020), pioneered by Ultralytics LLC in 2020, is composed of three core components: backbone, neck, and head networks. This study targets the YOLOv5s variant because of its superior performance compared to other iterations within the YOLO series. The backbone network employs a series of convolutional operations and fusion steps to extract the feature maps from input images. Subsequently, the neck network integrates feature maps of diverse dimensions, obtained from the backbone network. This amalgamation yields an upgraded, innovative feature map that effectively preserves contextual information, mitigating information loss. It is important to highlight that this process leverage the FPN (Feature Pyramid Network) structure (Lin et al., 2017) to facilitate the propagation of robust semantic features from higher-level feature maps to their lower-level counterparts. Simultaneously, the PANet

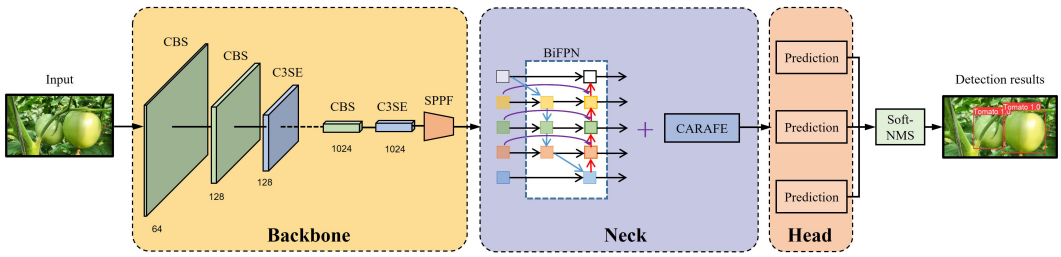


FIGURE 1
Overview of the SBCS-YOLOv5s.

(Path Aggregation Network) architecture (Liu et al., 2018) facilitates the transmission of robust localization features from lower-level feature maps to their higher-level counterparts. The head network, the final segment of the model, consists of three layers that generate output feature maps at distinct scales.

The CBS (Convolution, batch normalization, and SiLU activation function) is a conventional convolution layer in the YOLOv5s network. It encompasses a sequence of operations, including convolution, batch normalization (Ioffe and Szegedy, 2015), and the SiLU activation function (Elfwing et al., 2018). YOLOv5 originally employed the BottleneckCSP module instead of the C3 module for feature extraction. The BottleneckCSP module combines the concepts of Bottleneck (He et al., 2016) and CSP (Cross-Stage Partial connections) (Wang et al., 2020a). It involves three successive convolutional kernel operations, with the output of the first being processed through two more convolutional kernels. This sequence culminates in the fusion of unprocessed and convolved features. The primary objective of the BottleneckCSP module is to deepen the model.

The CSP module introduced by Wang et al. (2020a) splits the input into two segments; one undergoes processing via a block (like Bottleneck), while the other proceeds directly through a 1×1 convolutional layer. These two streams are then recombined. The C3 module supplants a 1×1 convolutional layer within the BottleneckCSP module, simplifying the network architecture to enable the extraction of feature maps and minimize the computation complexity. The C3 module comprises two branches, each involving a convolution operation that reduces the feature map channel count by half. The output from these two branches is concatenated using the Bottleneck module, followed by a convolutional layer within the second branch. These processes tightly integrate the output feature maps from both branches, with a final convolutional layer generating the output feature map of the module. Furthermore, SPPF (Spatial Pyramid Pooling Fusion) augments the ability of the backbone to express features. This module employs a sequence of three convolutions with identical kernels, focusing on the amalgamation of features from various resolutions.

2.2 Content-aware reassembly of features

The YOLOv5 model uses a nearest neighbor interpolation method for its up-sampling process, utilizing the same kernel for up-sampling across the feature map. Nevertheless, this approach does not leverage the semantic information in the feature map during the up-sampling process, resulting in a significant loss of features. This study integrates the CARAFE module (Wang et al., 2019), a novel technique, to address these limitations. The CARAFE module consists of two main components: a content-aware reassembly module and a kernel prediction module. It anticipates and assembles the recombined kernel, reconstructing the features within predetermined local regions at each point while using the underlying content details. The CARAFE module dynamically adjusts and optimizes the reassembled kernels at distinct points based on the content information, offering superior performance compared to alternative up-sampling methods like interpolation. For every predefined location, the utilization of a reassembly kernel becomes imperative, with the kernel size denoted as k_{up}. The reassembly procedure is illustrated using (Equation 1):

$$O_{l'} = \sum_{n=-r}^{r} \sum_{m=-r}^{r} \mathcal{W}_{l'_{(n,m)}} \cdot I_{(i+n,j+m)} \quad (1)$$

where O and I represent the output and input, respectively. $\mathcal{W}_{l'}$ denotes the location-wise kernel associated with each location l' based on the input. l' signifies the neighboring location of l, and $r = \frac{k_{up}}{2}$.

The CARAFE approach significantly enhances the semantic richness of the reassembled feature maps compared to the nearest neighbor interpolation up-sampling technique. This approach is achieved by strategically emphasizing crucial points within localized regions. In scenarios where tomatoes overlap or are densely packed, CARAFE's ability to enhance spatial detail helps the model distinguish between closely spaced fruits, potentially reducing the number of merge detections. It also helps the model to improve localization accuracy in tomato detection. In addition, CARAFE encompasses a wider scope of observation, adept content handling, and its lightweight design, culminating in expedited computations. Figure 2 shows the architectural representation of CARAFE.

3 Materials and methods

3.1 Image acquisition

Images of tomatoes were taken from December 2017 to November2019 in the greenhouses of a tomato production base, located in Shouguang city, China, with a digital camera (DSC-W170, Sony, Tokyo, Japan) at a resolution of 3648×2056 pixels. The camera was equipped with a 5×Carl Zeiss Vario-Tessar precision zoom lens. The distance between the camera and the target was from 500 mm to 1000 mm. Nine hundred and sixty-six images were captured under natural daylight (sunny and cloudy days) with different conditions such as shading, sunlight, occlusions, and overlaps. The training set had 725 images, while the test set contained 241 images. The scale of tomatoes in the images varies greatly, ranging from 200 to 1500 pixels in diameter.

3.2 Image augmentation

This study used data augmentation to counteract potential issues, such as over-fitting or non-convergence, that could arise during training. The augmentation of images was accomplished by applying diverse techniques, such as brightness transformation, blur, horizontal flip, noise, and rotation. These methods were employed to enhance the resilience of the model against noise and its ability to remain unaffected by variations in camera positioning. In particular, introducing a Random Gaussian blur makes the model more resistant to camera blur, with a threshold of

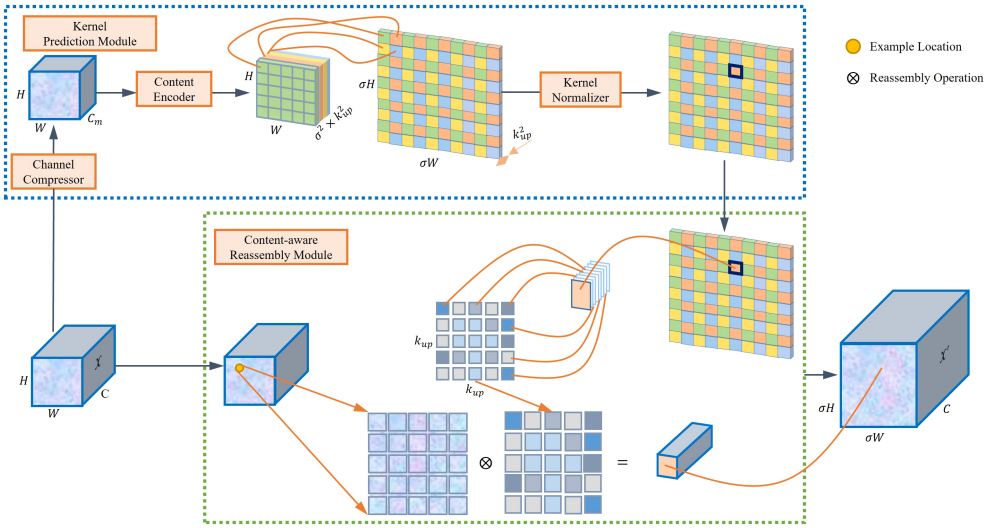


FIGURE 2
Overall architecture of the CARAFE module.

25 pixels for maximum blur. In addition, the incorporation of horizontal and vertical flips played a role in fortifying the capacity of the model to perform consistently regardless of the orientation of the subject. A visual representation of data enhancement techniques can be observed in Figure 3.

3.3 The SBCS-YOLOv5s architecture

The YOLOv5 model represents a single-stage object detection algorithm that introduces substantial enhancements over other YOLO models. On the other hand, the challenge of achieving high accuracy and fast speed persists in the tomato detection case, primarily because of the intricacies of the natural environment, such as occlusions and overlapping. This study proposes an SBCS-YOLOv5s model to address this issue, with the incorporation of SE, BiFPN, CARAFE, Soft-NMS into the YOLOv5s. The first module of this approach is used for feature extraction, merging the SE module (Hu et al., 2018) and the C3 module of the YOLOv5s model. This fusion enhances the network focus on useful information, refines the feature extraction process, and improves the model's robustness to variations in environmental conditions. The neck network integrates BiFPN (Tan et al., 2020) and CARAFE modules (Wang et al., 2019) into YOLOv5s, enriching features with more profound semantic information. The conventional NMS algorithm (Hosang et al., 2017) used in YOLOv5s was substituted with the Soft-NMS algorithm (Bodla et al., 2017) to make the network more efficient in detecting occluded and overlapped fruits. Additional intricacies of this approach are elaborated upon in subsequent sections. Figure 4 presents the architecture of SBCS-YOLOv5s.

3.3.1 The modified backbone network

The SE attention module (Hu et al., 2018) in Figure 5A is fused with the C3 module structure into an improved C3SE module. The

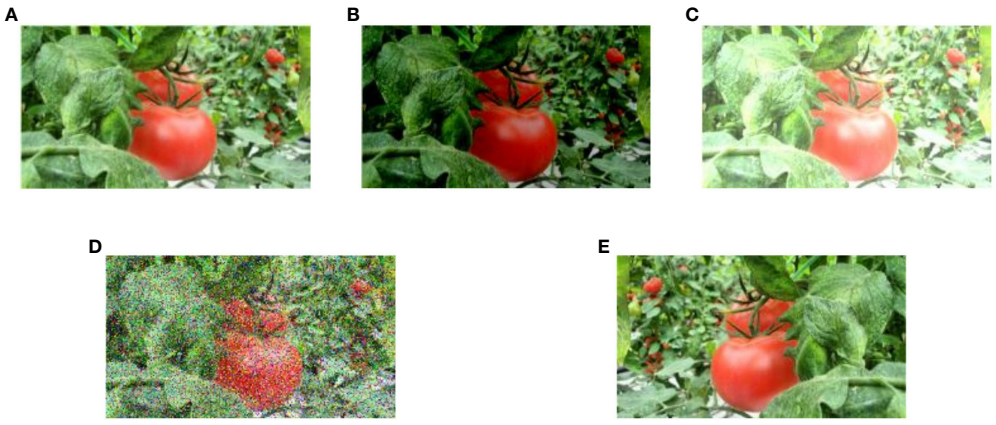


FIGURE 3
Examples of data enhancement techniques. **(A)** Input image, **(B, C)** varied exposure, **(D)** Noise (salt and pepper), and **(E)** Horizontal Flip.

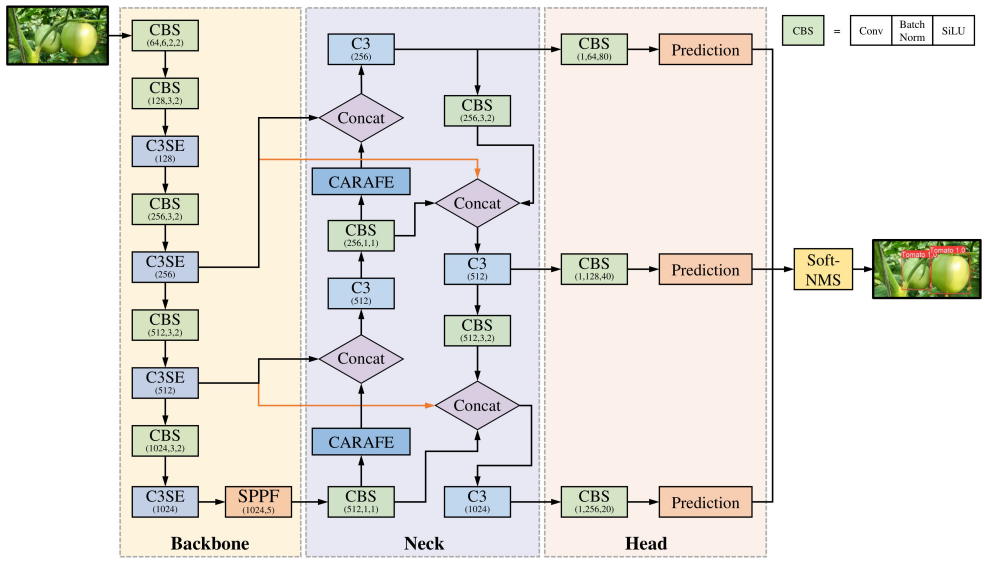


FIGURE 4
The architecture of SBCS-YOLOv5s.

SE module solves the issue of feature maps containing informative and less relevant channels. The re-calibration process empowers the network to prioritize informative channels while suppressing the less useful ones. In addition, the SE module's ability to adaptively re-calibrate features helps to improve the robustness of the model to variations in illumination and environmental conditions. It also helps to reduce over-fitting, which is essential for tomato detection to accurately identify the boundaries of individual tomatoes in an image. Figure 5B presents the structure of the C3SE module. The weight of each channel is allocated using the interdependence of the feature channels to facilitate the neural network to focus on significant feature information and to minimize the impact of feature redundancy. The SE attention module comprises three key operations: squeeze, excitation, and scale.

The squeeze operation, also called compression, involves applying a global average pooling operation to each channel of the feature map. This compresses the spatial dimensions of the feature map, converting its size into multiple features while maintaining the overall channel dimension. For example, if the input feature map holds a size of $H \times W \times C$, and $V = [v_1, v_2, \ldots, v_c]$ is an example input set, the transformation of the squeeze operation can be expressed using (Equation 2).

$$F_{sq}(V_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} V_c(i,j) \tag{2}$$

where $c \in C$, and C denotes the number of feature channels, while W signifies the feature map width; H corresponds to the height of the feature map; F_{sq} denotes the specific squeeze operation being discussed.

The excitation operation consists of two primary components: a fully connected layer and a sigmoid activation function. The fully connected layer incorporates comprehensive information from all input features. Subsequently, the sigmoid function transforms the input into a range confined within [0,1]. This process is visually represented by (Equation 3).

$$F_{ex}(F_{sq}, B) = \sigma(B_1 \cdot \delta(B_2 \cdot F_{sq})) \tag{3}$$

where σ symbolizes the sigmoid activation function, δ signifies the ReLU activation function, and F_{ex} denotes the excitation operation. B_1 and B_2 denote the weights of the fully connected layer, respectively.

Finally, the scale operation combines or multiplies the input channel weight with the weight derived from the channel feature of the two preceding operations. (Equation 4) shows the rescaling operation:

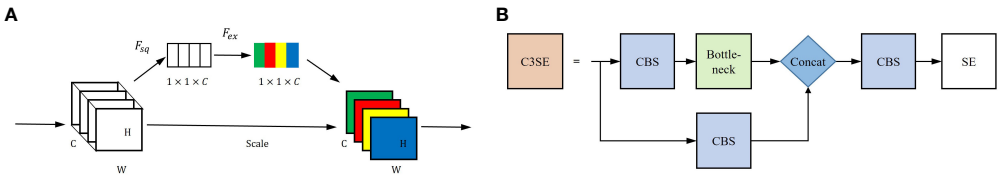


FIGURE 5
Original SE module and improved C3SE module architecture. **(A)** SE module architecture, **(B)** C3SE module architecture.

$$F_{scale}(V_c, S_c) = S_c V_c \tag{4}$$

where $F_{scale}(V_c, S_c)$ refers to channel-wise multiplication that takes place between S_c and V_c.

3.3.2 The modified neck network

The FPN+PANet network was replaced in the YOLOV5s neck with the weighted BiFPN in this study. The rationale stems from large-scale objects possessing many pixels, whereas small objects have few. The features of large objects can be easily maintained in the convolution process, while the features of the smaller ones can be easily ignored. The YOLOv3 model introduced the FPN network structure (Lin et al., 2017), emphasizing the down-sampling process of semantic information extraction. Based on this, the YOLOv5 incorporates PANet (Liu et al., 2018) to aggregate image features by incorporating secondary bottom-up fusion, as shown in Figure 6A. This approach integrates accurate low-level localization signals to enrich the entire feature hierarchy and facilitate the flow of information. On the other hand, PANet is characterized by simple two-way fusion, and their contributions to the output features often remain unequal because of the varying input resolutions. Furthermore, feature fusion of PANet involves a direct addition of distinct input features, leading to unbalanced output features and complicating computational processes.

The BiFPN, introduced by Tan et al. (2020), is an object detection model module. Its main strength lies in effectively fusing information within a deep learning network, ensuring efficiency and accuracy. The problem of correctly combining multi-scale features from multiple layers of a convolutional neural network are solved to improve the detection accuracy of objects at various scales. The bottom-up and top-down paths are used to construct a feature pyramid that captures fine-grained features. The BiFPN combines the feature maps from the bottom-up and top-down paths. Furthermore, to avoid all feature maps contributing equally, the BiFPN provides learnable weights for each input feature map, allowing the network to assign varied priorities to different scales and resolutions. The notable enhancement brought by BiFPN is the introduction of a bi-directional connection between neighboring levels of the network. This augmentation substantially improves the flow of information and gradient propagation during the training process. It also improves to tomato localization, helping the model to capture details at different scales and make more accurate predictions of tomato locations. In addition, BiFPN is designed to be computationally efficient, making it well suited for real-time detection. Figure 6B shows the BiFPN architecture. (Equation 5) shows the fast normalized fusion between the feature maps from the bottom-up and top-down paths.

$$\begin{cases} P_6^{td} = Conv\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize(P_7^{in})}{w_1 + w_2 + \varepsilon}\right) \\ P_6^{out} = Conv\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot Resize(P_5^{out})}{w_1' + w_2' + w_3' + \varepsilon}\right) \end{cases} \tag{5}$$

where the intermediate feature situated at Level 6 along the top-down pathway is P_6^{td}, while the resulting feature at Level 6 stemming from the bottom-up pathway is P_6^{out}, *Conv* and *Resize* correspond to convolution and sampling operations, respectively. w and ϵ represent the weight and a small pre-set value to avoid numerical instability, respectively. Usually, this value was set to 0.0001.

BiFPN improves the detection accuracy compared to the PANet used in the YOLOv5s model. Nevertheless, the BiFPN employs a nearest neighbor interpolation method for the up-sampling of feature maps. Using this approach could lead to a small receptive field and make the network focus only on sub-pixel spaces, resulting in the loss of rich semantic information. In this study, the CARAFE module was introduced to the BiFPN to tackle this problem. This integration improved feature maps with rich information and high resolutions. Section 2.2 describes the CARAFE module in detail.

3.3.3 Soft-NMS (non-maximum suppression) algorithm

The soft-NMS algorithm (Bodla et al., 2017) is a modified version of the conventional NMS algorithm (Hosang et al., 2017) used by the YOLOv5 framework. The fundamental principle behind the NMS algorithm involves selecting the bounding box with the highest confidence score. It suppresses the other bounding boxes with significant overlap with the selected box, leading to the missed detection of overlapping fruits. Moreover, the NMS algorithm does not perform optimally when dealing with different scales. Equation 6 shows the NMS algorithm:

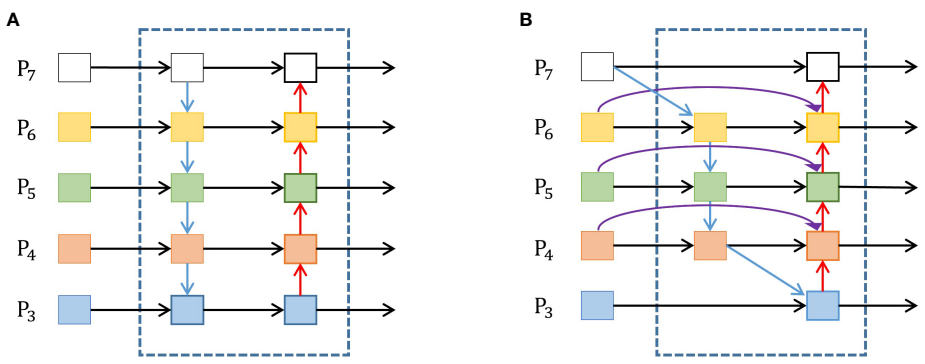


FIGURE 6
Architectures of PANet and BiFPN. **(A)** PANet architecture, **(B)** BiFPN architecture.

$$\hat{S}_i = \begin{cases} \hat{S}_i, & IoU\ (M, \hat{b}_i) < N_t \\ 0, & IoU\ (M, \hat{b}_i) \geq N_t \end{cases} \tag{6}$$

where $\hat{b}_i$ and $\hat{S}_i$ denote the ith predictor and its score, respectively. N_t is the pre-set threshold; M denotes the candidate box having the highest score; $IoU\ (M, \hat{b}_i)$ is the overlap region between M and $\hat{b}_i$.

The objective of the Soft-NMS algorithm is to solve the limitations of the traditional NMS algorithm approach. It is also designed to be more tolerant to overlapping objects. This is achieved using a softening function that progressively decreases the scores of bounding boxes overlapping with the one possessing the highest score. The primary goal is to reduce the severe suppression of surrounding boxes that might be slightly less confident but still contain useful information. This modification seeks to enhance the detection accuracy and improve the handling of cases involving overlapping fruits within the final detection results. And it helps maintain a consistent ranking of bounding box scores, even when there is overlap. The Soft-NMS algorithm is outlined in (Equation 7):

$$\hat{S}_i = \begin{cases} \hat{S}_i, & IoU\ (M, \hat{b}_i) < N_t \\ \hat{S}_i e^{-\frac{IoU(M,\hat{b}_i)^2}{\sigma}}, & IoU\ (M, \hat{b}_i) \geq N_t \end{cases} \tag{7}$$

where σ represents the hyperparameter of the penalty function. When the $IoU\ (M, \hat{b}_i)$ exceeds the pre-defined threshold, the prediction frame confidence score is reduced systematically instead of being set to zero. As a result, the detection accuracy of overlapping and occluded fruits can be improved.

3.3.4 Loss function

The loss function used in this study is expressed as (Equation 8), which encompasses the regression error of bounding coordinates, the confidence error of the bounding box, and the classification error of object category.

$$L = Loss_{reg} + Loss_{conf} + Loss_{cls} \tag{8}$$

In this study, the bounding box regression loss incorporatesthe use of *CIoU* (Complete IoU) as in (Equation 8.a). It could accurately represent the gap between the prediction and annotation frames, enhancing the network model during training. It also considers crucial factors, such as the overlapping area (expressed in Equation 8.b), central point distance, and aspect ratio (expressed in Equation 8.c) between b and b_{gt}.

$$CIoU = 1 - IoU + \frac{d^2\ (\hat{b}, b_{gt})}{c^2} + \alpha v \tag{8.a}$$

with

$$IoU = \frac{\hat{b} \cap b_{gt}}{\hat{b} \cup b_{gt}} \tag{8.b}$$

and

$$v = \frac{4}{\pi^2}(tan^{-1}\frac{w_{gt}}{h_{gt}} - tan^{-1}\frac{w}{h})^2, \qquad \alpha = \frac{v}{(1 - IoU) + v} \tag{8.c}$$

where b and b_{gt} represent the predicted and ground truth bounding boxes, respectively. d signifies the distance between the predicted center point and the true center point; c is the diagonal length of the enclosing box covering b and b_{gt}; α and v are the positive trade-off and aspect ratio parameters, respectively.

Object classification loss is expressed as (Equation 8.e), wherein the process is initiated by calculating the confidence C of the cell grid as in Equation 8.d):

$$C = P(object) \times IoU(b, b_{gt}) \tag{8.d}$$

then,

$$Loss_{conf} = \sum_{i=1}^{s\times sNB}\sum_{j=1}\lambda_{i,j}[C_i \cdot log(\tilde{C}_i)log(1 - C_i)] - \sum_{i=i}^{s\times sNB}\sum_{j=1}(1 - \lambda_{i,j})[C_i \cdot log\tilde{C}_i + (1 - C_i)log(1 - \tilde{C}_i) \tag{8.e}$$

with λi,j expresses in (Equation 8.f):

$$\lambda_{i,j} = \begin{cases} 1, & if\ part\ of\ j-th\ bounding\ box\ is\ in\ the\ i-th\ grid\ cell \\ 0, & otherwise \end{cases} \tag{8.f}$$

where s×s denotes the size of the grid cell; *NB* stands for the number of bounding boxes; $\tilde{C}_i$ represents the confidence obtained from the prediction box; C_i signifies the confidence threshold.

3.4 Experimental setup

The experiments of this research were conducted using an Intel i5 64-bit quad-core CPUs operating at a frequency of 3.30 GHz (Santa Clara, CA, USA). The system had 16 GB of RAM and an NVIDIA GeForce GTX 1070Ti GPU with 8 GB memory. The chosen model framework was PyTorch, with CUDA 11.1 and Python 3.8.10 for implementation. Table 1 lists some hyper parameters used in the experiments.

TABLE 1 Configuration of certain hyper-parameters.

Parameters	Value
Number of epochs	400
Learning rate	0.001
Optimizer weight decay	94.75
STD momentum	96.3
Warm-up initial momentum	0.8
Batch size	8
Box loss gain	0.05
Cls (classification loss gain)	0.5
Cls_pw (cls BCE loss positive weight)	1.0
Obj (object loss gain)	1.0
Anchor_t (anchor multiple threshold)	4.0

The criteria used for assessing the performance of fruit detection encompassed precision, recall, mean average precision (mAP), and F_1 score (Padilla et al., 2020). The metrics are defined in (Equations 9–12):

$$R = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Negatiive} \tag{9}$$

$$P = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Positive} \tag{10}$$

where R and P are the recall and precision, respectively. Using mAP is a valuable approach to assess the model performance across different confidence levels.

$$mAP = \frac{1}{N_{cls}} \sum_{a=1}^{N_{cls}} AP_a \tag{11}$$

with AP expresses in (Equation 11.a):

$$AP = \sum_{q}^{Q} (r_{q+1} - r_q) \max_{\tilde{r} \geq r_{q+1}} p(\tilde{r}) \tag{11.a}$$

where $p(\tilde{r})$ represents the calculated precision at a given recall value $(\tilde{r})$, whileN_{cls} is the total number of classes.

$$F_1 = \frac{2 \times R \times P}{R + P} \tag{12}$$

4 Results and discussions

4.1 Ablation study

The first step in this study was to determine which attention mechanism (CBAM (Woo et al., 2018), ECA (Wang et al., 2020b), CA (Hou et al., 2021), SE (Hu et al., 2018)) works better on the tomato datasets after fusing the original C3 module network. From Table 2, we can see that integrating the SE attention module with the C3 module led to a notable outcome. The mean average precision with an IoU of 0.5 to 0.95 reached 85.1%, which is the best result.

Since the SE attention module relies on modeling channel-wise relationships and adaptive re-calibration of feature maps to capture important information, it helps to improve feature extraction of the model. The fusion of the SE attention module with the C3 module was implemented within the backbone network. Furthermore, the integration of BiFPN, CARAFE, and Soft-NMS was used in the neck to improve the detection capabilities of YOLOv5s. An ablation study was carried out to evaluate the effectiveness of each component.

Integrating the SE attention module with the C3 module resulted in a 0.9% increase in the mean average precision with an IoU of 0.5 to 0.95, as shown in Table 3. This enhancement underscores the efficacy of the SE attention module to channel the model towards useful information. Subsequently, a further increase of 0.6% in mAP was achieved by replacing PANet with BiFPN. This is because the BiFPN assists the model in determining useful weights for comprehensive fusion of high-level and low-level features, thereby improving detection performance. Discernible performance improvements became evident after incorporating the CARAFE module as an up-sampling operator within PANet and BiFPN. This is due to the fact that CARAFE enhances spatial details and improves feature map resolution better than the original up-sampling method. On the other hand, the most remarkable results emerged when the Soft-NMS algorithm was applied to the BiFPN+CARAFE configuration, showcasing 3.5% advancement over the original YOLOv5s model. This proves the advantage of the continuous weighting scheme of Soft-NMS. This sequence of observations indicates a substantial enhancement in detection performance through different modifications.

TABLE 2 Ablation analysis of different attention mechanisms.

	C3	CBAM	ECA	CA	SE	mAP (0.5:0.95) (%)
Modifications	✓					84.2
	✓	✓				83.7
	✓		✓			84.9
	✓			✓		84.4
	✓				✓	85.1

TABLE 3 Ablation analysis of different components.

	C3SE	PANet	BiFPN	CARAFE	Soft_NMS	mAP (0.5:0.95) (%)
Modifications		✓				84.2
	✓	✓				85.1
	✓		✓			85.7
	✓	✓		✓		86.7
	✓		✓	✓		87.2
	✓		✓	✓	✓	87.7

4.2 Feature map visualization

Visualizations were performed to compare the improved model variants with the original YOLOv5s. Figure 7A presents an input image with tomatoes annotated for improved visibility. Figures 7B, C show the difference between the C3 and C3SE modules, respectively. In particular, Figure 7C highlights finer details that are more discernible. This observation underscores the role of the SE module in steering the backbone network towards useful information. Figures 7D, E represent the original neck of YOLOv5s and the modified neck used in SBCS-YOLov5s, respectively. Figure 7E shows an improved feature map with heightened resolution after incorporating the BiFPN and CARAFE modules. These enhancements facilitate efficient context information aggregation and seamless fusion within the network.

Every modification produced superior features with high resolution compared to those in the original model (Figure 7). This visual evidence substantiates that SBCS-YOLOv5s excels in accuracy, resilience, and efficiency when compared to the original model.

4.3 Comparison of the SBCS-YOLOv5s with other detection algorithms

The performance of SBCS-YOLOv5s was compared with several other object detection models. These models included Faster-RCNN (Ren et al., 2015), Dynamic-RCNN (Zhang et al., 2020), YOLOv3 (Redmon and Farhadi, 2018), YOLOv3-tiny (Redmon and Farhadi, 2018), YOLOv4 (Boschkovskiy et al., 2020), YOLOv4-tiny (Boschkovskiy et al., 2020), YOLOv7-tiny (Wang et al., 2022), and YOLOv5s (Jocher et al., 2020).

The mean average precision with IoU of 0.5 to 0.95 was 3.8%, 9.7%, 5.8%, 16.4%, 4.6%, 9.3%, 4.5%, and 3.5% higher than those of Faster-RCNN, ynamic RCNN, YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny, YOLOv7-tiny, and YOLOv5s, respectively (Table 4). Furthermore, the detection time achieved 2.6 ms per image, fulfilling the real-time detection criteria. Moreover, the precision of the proposed model improved by 0.3%, 1.5%, 2.5%, 1.7%, 1.4%, 1.5%, 0.6%, and 1.4% compared to the Faster RCNN, Dynamic RCNN, YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny, YOLOv7-tiny, and YOLOv5s,

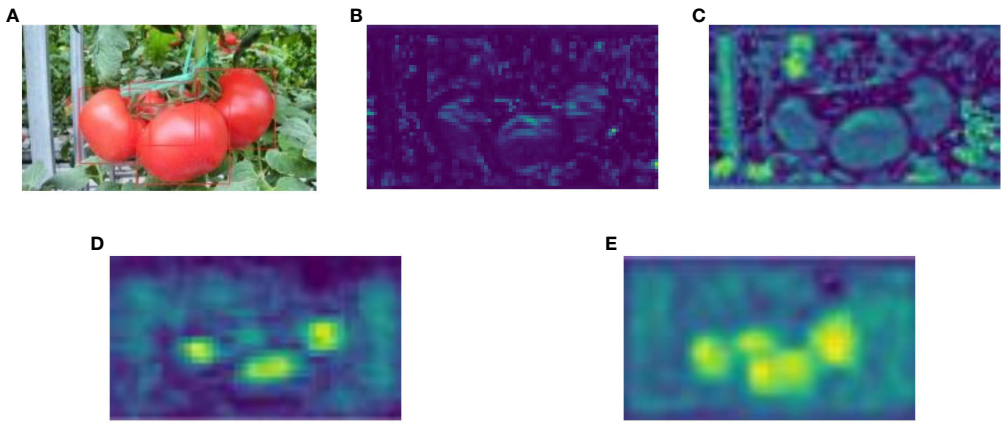


FIGURE 7
(A) Annotated input image, **(B)** the feature of the C3 module of YOLOv5s, **(C)** feature of the C3SE module of SBCS-YOLOv5s, **(D)** feature of the original neck of YOLOv5s, and **(E)** feature of neck of SBCS-YOLOv5s.

TABLE 4 Comparison of the different models.

Model	Precision (%)	Recall (%)	F1 (%)	mAP (0.5) (%)	mAP (0.5:0.95) (%)	Time (ms)
Faster-RCNN (VGG-16)	96.5	94.8	95.6	97.8	83.9	3.9
Dynamic RCNN	95.3	93.2	94.2	96.6	78.0	2.4
YOLOv3	94.3	92.4	93.4	97.1	81.8	4.8
YOLOv3-tiny	95.1	91.9	93.4	97.4	71.3	3.8
YOLOv4	95.4	95.3	95.3	97.5	83.1	4.3
YOLOv4-tiny	95.3	94.0	94.6	98.0	78.4	3.5
YOLOv7-tiny	96.2	94.2	95.1	98.2	83.2	4.3
YOLOv5s*	95.4	94.5	95.4	98.2	84.2	4.1
SBCS-YOLOv5s	96.8	97.3	97.04	98.7	87.7	2.6

*YOLOv5s v6. 1 version is used in this study.

respectively. The F1 score and mAP with an IoU of 0.5 increased by 1.64% and 0.5%, respectively, compared to the original YOLOv5s model. Hence, the performance of SCBS-YOLOv5s was improved compared to other object detection networks. Importantly, the experimental results revealed the efficient real-time detection capability of SCBS-YOLOv5s in accurately identifying tomatoes within their natural environmental context.

The detection performance of the improved YOLOv5s surpassed that of alternative models while demonstrating greater efficiency (Figure 8). The mean average precision with an IoU of 0.5 to 0.95 exhibited a notable 3.5% improvement compared to the original YOLOv5s model. Furthermore, the processing time for detecting each image was decreased by 1.5ms. These results collectively signify the improved model prowess in achieving improved accuracy, compactness, and efficiency when tasked with fruit detection in a natural environment.

4.4 Performance of the improved model under different conditions

In a natural environment, tomatoes are exposed to different lighting conditions because of the uneven illumination. Moreover, they can become obscured by leaves or branches and might overlap. The performance of the improved model was assessed across diverse scenarios. Table 5 shows how the tomatoes were classified into sunshine and shade cases regarding illumination. Within the test dataset, there were 425 tomatoes under shaded conditions and 487 tomatoes under sunlight conditions. In terms of obscured and overlapped severity, the tomatoes were classified as mild and significant occlusion, as delineated in Table 5. The latter pertains to situations where tomatoes are obstructed by leaves, branches, or other tomatoes by over 50%.

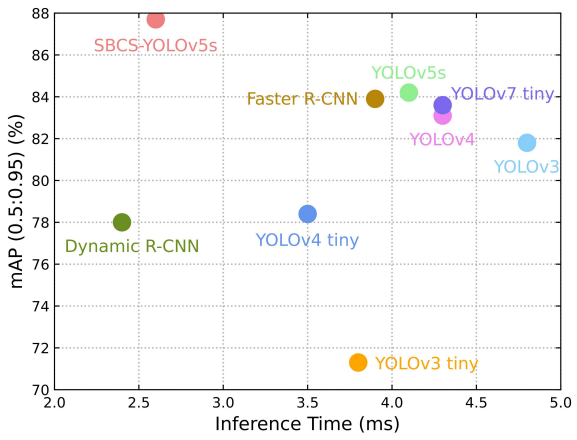


FIGURE 8
Detection performance of different models (accuracy vs. inference time).

The correct detection rate for tomatoes under sunlight conditions was 97.2%, while the rate was 97.4% when tomatoes were in shaded conditions (Table 5). False identification was 3.1% for sunlight and 3.3% for shade, respectively. Approximately 97.7% of the tomatoes were detected accurately when they exhibited mild occlusion, with a correctness rate of 96.4% in the case of severe occlusion (Table 5). The rates of missed identification were 2.3% and 3.6% for mild and severe occlusions, respectively. Figure 9 presents some examples of detection outcome instances under various conditions. The results revealed the robustness of the improved model in effectively managing variations in illumination and situations involving overlapping fruits.

4.5 Qualitative analysis between SBCS-YOLOv5s and the original YOLOv5s model

Figure 10 shows some prediction results from the SBCS-YOLOv5s and the original YOLOv5s model.

As shown in Figure 10, the detection performance of SBCS-YOLOv5s was superior to the original YOLOv5s model. In particular, Figure 10G visually demonstrates the improved model focus on more useful information and retain the features for small tomatoes. Moreover, the original YOLOv5s model returned some false negatives and false positives, as shown in Figures 10E, F.

5 Conclusions and future work

This paper introduced an accurate and efficient tomato detection solution named SBCS-YOLOv5s, which builds upon the YOLOv5s framework. The accuracy and efficiency of the model were improved by substituting the original C3 module within YOLOv5s with a C3SE module, combining the SE attention module with the C3 module. This change amplified the feature extraction capabilities. Furthermore, the PANet in the neck of the original model was replaced with a weighted Bi-directional Feature Pyramid Network (BiFPN), enhancing the adaptability of the detector to objects of varying scales by fusing high-level and

TABLE 5 Performance of the improved model under different conditions.

Conditions	Tomato Count	Correctly Identified		Falsely Identified		Missed	
		Amount	Rate (%)	Amount	Rate (%)	Amount	Rate (%)
Sunlight	487	473	97.2	15	3.1	14	2.8
Shading	425	414	97.4	14	3.3	11	2.6
Mild occlusion	609	595	97.7	17	2.8	14	2.3
Severe occlusion	303	292	96.4	12	3.9	11	3.6

FIGURE 9
Some examples of tomato detection results under different conditions. **(A–C)** sunlight cases, and **(D–F)** shade cases, **(G–I)** slight occlusions, and **(J–L)** severe occlusions.

FIGURE 10
Some detection results from the two models. **(A–C)** ground Truth, **(D–F)** prediction images from the YOLOv5s model, and **(G–I)** prediction images from SBCS-YOLOv5s.

bottom-level features at high resolution. Furthermore, the traditional up-sampling operator within the BiFPN structure was substituted with the CARAFE module to yield more refined semantic information. Finally, the conventional NMS algorithm was replaced with the Soft-NMS algorithm to improve the detection accuracy of overlapped and occluded fruits.

A thorough experimentation was carried out to validate the performance of SBCS-YOLOv5s. An ablation study was instrumental in substantiating the efficacy of each modification. The findings of the experiment showed that the mAP with an IoU of 0.5 to 0.95 had 3.8%, 9.7%, 5.8%, 16.4%, 4.6%, 9.3%, 4.5%, and 3.5% improvements compared to other object detection algorithms, reaching 2.6ms per image in terms of detection time.

Furthermore, the experiments underscored the robustness of SBCS-YOLOv5s because it effectively detected tomatoes across diverse scenarios involving varying lighting and occlusion conditions.

Despite the excellent performance of the improved model, there is room for enhancing the detection performance. In the future study, the explicit incorporation of contextual information will be explored to refine the detection accuracy. Moreover, we will consider incorporating information about tomato maturity to enable differentiation among tomatoes at distinct growth stages based on SBCS-YOLOv5s presented in this study.

Data availability statement

The original contributions presented in the study are included in the article/supplementary aterial. Further inquiries can be directed to the corresponding authors.

Author contributions

PT: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. GL: Data curation, Formal analysis, Funding acquisition, Software, Validation, Writing – review & editing. SP: Funding acquisition, Investigation, Project administration, Supervision, Validation, Writing – review & editing. JK: Methodology, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by BK21PLUS, Creative Human Resource Development Program for IT Convergence, Shandong Provincial Natural Science Foundation (ZR2023QC116, ZR2022ME155), Weifang Science and Technology Development Plan (2021GX054, 2021GX056), and the Doctoral Research Foundation of Weifang University (2022BS70).

Conflict of interest

Author GL was employed by the company Univalsoft Joint Stock Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Azarmdel, H., Jahanbakhshi, A., Mohtasebi, S. S., and Munoz, A. R. (2020). Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM). *Posthavest Biol. Technol.* 166, 111201. doi: 10.1016/j.postharvbio.2020.111201

Bodla, N., Singh, B., Chellappa, R., and Davis, L. S. (2017). "Soft-NMS–improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (Venice, Italy: IEEE),5561–5569. doi: 10.48550/arXiv.1704.04503

Boschkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv* preprint arXiv:2004.10934. doi: 10.48550/arXiv.2004.10934

Bresilla, K., Perulli, G. D., Boini, A., Morandi, B., Corelli Grappadelli, L., and Manfrini, L. (2019). Single-shot convolution neural networks for real-time fruit detection within the tree. *Front. Plant Sci.* 10. doi: 10.3389/fpls.2019.00611

Cao, Z., Mei, F., Zhang, D., Liu, B., Wang, Y., and Hou, W. (2023). Recognition and detection of persimmon in a natural environment based on an improved YOLOv5 model. *Electronics* 12, 785. doi: 10.33990/electronics12040785

Chen, W., Lu, S., Liu, B., Chen, M., Li, G., and Qian, T. (2022). CitrusYOLO: a lagorithm for citrus detection under orchard environment based on YOLOv4. *Multimed. Tools Appl.* 81, 31363–31389. doi: 10.1007/s11042-022-12687-5

Elfwing, S., Uchibe, E., and Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* 107, 3–11. doi: 10.1016/j.neunet.201712.012

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Columbus, OH, USA: IEEE),580–587. doi: 10.48550/arXiv.1311.2524

Goel, N., and Sehgal, P. (2015). Fuzzy classification of pre-harvest tomatoes for ripeness estimation-an approach based on automatic rule learning using decision tree. *Appl. Soft Comput.* 36, 45–56. doi: 10.1016/j.asoc.2015.07.009

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., and Lewis, K. (2015). Sensors and systems for fruit detection and localization: a review. *Compt Electr. Agric.* 116, 8–19. doi: 10.1016/j.compag.2015.05.021

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969. doi: 10.48550/arXiv.1703.06870

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Las Vegas, NV, USA: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Hosang, J., Benenson, R., and Schiele, B. (2017). "Learning non-maximum suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Honolulu, HI, USA: IEEE), pp. 4507–4515. doi: 10.1109/CVPR.2017.685

Hou, Q., Zhou, D., and Feng, J. (2021). "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Nashville, TN, USA: IEEE), 13713–13722. doi: 10.48550/arXiv.2103.02907

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Lake city, UT, USA: IEEE), 7132–7141. doi: 10.1109/CVPR.2018.00745

Ioffe, S., and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. (Lille, France: PMLR), 448–456. doi: 10.48550/arXiv.1502.03167

Jana, S., and Pareskh, R. (2017). "Shape-based fruit recognition and classification," in *Proceedings of International Conference on Computational Intelligence, Communications, and Business Analytic (CIBA)*. (Kolkata, India: Springer). doi: 10.1007/978-981-10-6430-2_15

Jiao, Y., Luo, R., Li, Q., Deng, X., Yin, X., Ruan, C., et al. (2020). Detection and localization of overlapped fruits application in an apple harvesting robot. *Electronics* 9, 1023. doi: 10.3390/electronics9061023

Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., et al. (2020). ultralytics/yolov5: v3. 0. *Zenodo*.

Kurtulmus, F., Lee, W. S., and Vardar, A. (2011). Green citrus detection using "eigenfruit" color and circular Gabor texture features under natural outdoor conditions. *Comput. Electr. Agric.* 78, 140–149. doi: 10.1016/compag.2011.07.001

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Honolulu, HI, USA: IEEE), 2117–2125. doi: 10.1109/CVPR.2017.106

Liu, G., Mao, S., and Kim, J. H. (2019). A mature-tomato detection algorithm using machine learning and color analysis. *Sensors* 19, 2023. doi: 10.3390/s19092023

Liu, G., Nouaze, J. C., Touko, M. P. L., and Kim, J. H. (2020). YOLO-Tomato: a robust algorithm for tomato detection based on YOLOv3. *Sensors* 20, 2145. doi: 10.3390/s20072145

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Lake City, UT, USA: IEEE), 8759–8768. doi: 10.48550/arXiv.1803.01534

Mbouembe, P. L. T., Liu, G., Sikati, J., Kim, S. C., and Kim, J. H. (2023). An efficient tomato-detection method based on improved YOLOv4-tiny model in complex environment. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1150958

Padilla, R., Netto, S. L., and Da Silva, E. A. (2020). "A survey on performance metrics for object detection algorithms," in *International Conference on Systems, Signals, and Image Processing (IWSSIP)*. (Niteroi, Brazil: IEEE), 237–242. doi: 10.1109/IWSSIP48289.2020.9145130

Payne, A., Walsh, K., Subedi, P., and Jarvis, D. (2014). Estimating mango crop yield analysis using fruit at 'stone hardening' stage and night time imaging. *Comput. Electr. Agric.* 100, 160–167. doi: 10.1016/j.compag.213.11.011

Peixoto, J. V. M., Neto, C. M., Campos, L. F., Dourado, W. D. S., Nogueira, A. P., and Nascimento, A. D. (2017). Industrial tomato lines: morphological properties and productivity. *Genet. Mol. Res.* 16, 1–15. doi: 10.4238/gmr16029540

Rahnemoofar, M., and Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors* 17, 905. doi: 10.3390/s17040905

Rakun, J., Stajnko, D., and Zazula, D. (2011). Detection fruits in natural scenes using spatial-frequency based texture analysis and multi-view geometry. *Comput. Electr. Agric.* 76, 80–88. doi: 10.1016/j.compag.2011.01.007

Redmon, J., and Farhadi, A. (2017). "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (Honolulu, HI, USA: IEEE), 7263–7271. doi: 10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv* preprint arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767

Redmon, J., Farhadi, A., Divvala, S., and Girshick, R. (2016). You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (Las Vegas, NV, USA: IEEE), 779–788. doi: 10.48550/arXiv.1506.0240

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: towards real-time object detection with region proposal networks. In *Proceedings of the International Conference on Neural Information Processing Systems 39*. (Montreal, QC, Canada: IEEE) 91–99. doi: 10.1109/TPAMI.2016.2577031

Santo, T. T., de Souza, L. L., dos Santos, A. A., and Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Elects. Agric.* 170, 105247. doi: 10.1016/compag.2020.105247

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). "Inception-v4, Inception-Resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence* (San Francisco, CA, USA: AAAI Press). doi: 10.1609/aaai.v31i1.11231

Tan, M., Pang, R., and Le, Q. V. (2020). "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (WA, USA: IEEE), 10781–10790. doi: 10.1109/CVPR42600.2020.01079

Wang, C. Y., Boschkovskiy, A., and Liao, H. Y. M. (2022). "Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Vancouver, CANADA: IEEE), 7464–7475. doi: 10.48550/arXiv.2207.02696

Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., and Yeh, I. H. (2020a). "CSPNet: a new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*. (Seattle, WA, USA: IEEE), 390–391. doi: 10.1109/CVPRW50498.2020.00203

Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C. C., and Lin, D. (2019). "CARAFE: Content-aware reassembly of features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (Seoul, Korea: IEEE), 3007–3016. doi: 10.48550/arXiv.1905.02188

Wang, X., Liu, J., and Liu, G. (2021). Diseases detection of occlusion and overlapping tomato leaves based on deep learning. *Front. Plant Sci.* 12, 792244. doi: 10.3309/fpls.2021.792244

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020b). "ECA-Net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11534–11542. doi: 10.1109/CVPR42600.2020.01155

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). "Cbam: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*. (Munich, Germany: Spinger), 3–19. doi: 10.1007/978-3-030-01234-2_1

Yang, Q., Chen, C., Dai, J., Xun, Y., and Bao, G. (2020). Tracking and recognition algorithm for robot harvesting oscillating apples. *Int. J. Agric. Biol. Eng.* 13, 163–170. doi: 10.25165/j.ijabe.20201305.5520

Zhang, H., Chang, H., Ma, B., Wang, N., and Chen, X. (2020). "Dynamic r-cnn: towards high quality object detection via dynamic training," in *European Conference on Computer Vision (ECCV)*. (Glasgow, UK: Springer, Cham, Computer Vision-ECCV), 260–275. doi: 10.1007/978-3-030-58555-6_16

Zhao, Y., Gong, L., Zhou, B., Huang, Y., and Liu, C. (2016b). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and color analysis. *Biosyst. Eng.* 148, 127–137. doi: 10.1016/j.biosystemseng.2016.05.001

Zhao, C., Lee, W. S., and He, D. (2016a). Immature green citrus detection based on color feature and sum of absolute transformed difference (SATD) using color images in the citrus grove. *Comput. Electr. Agric.* 124, 243–253. doi: 10.1016/j.compag.2016.04.009

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 12 January 2024
DOI 10.3389/fpls.2023.1328952

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Wei Meng,
Beijing Forestry University, China
Yuwan Gu,
Changzhou University, China
Xiaoyang Liu,
Huaiyin Institute of Technology, China

*CORRESPONDENCE
Chunhua Guo
gch58@163.com

RECEIVED 27 October 2023
ACCEPTED 15 December 2023
PUBLISHED 12 January 2024

CITATION
Han D and Guo C (2024) Automatic classification of ligneous leaf diseases via hierarchical vision transformer and transfer learning.
Front. Plant Sci. 14:1328952.
doi: 10.3389/fpls.2023.1328952

Automatic classification of ligneous leaf diseases via hierarchical vision transformer and transfer learning

Dianyuan Han and Chunhua Guo*

Media and Communications College of Weifang University, Weifang, Shandong, China

Background: Identification of leaf diseases plays an important role in the growing process of different types of plants. Current studies focusing on the detection and categorization of leaf diseases have achieved promising outcomes. However, there is still a need to enhance the performance of leaf disease categorization for practical applications within the field of Precision Agriculture.

Methods: To bridge this gap, this study presents a novel approach to classifying leaf diseases in ligneous plants by offering an improved vision transformer model. The proposed approach involves utilizing a multi-head attention module to effectively capture contextual information about the images and their classes. In addition, the multi-layer perceptron module has also been employed. To train the proposed deep model, a public dataset of leaf disease is exploited, which consists of 22 distinct kinds of images depicting ligneous leaf diseases. Furthermore, the strategy of transfer learning is employed to decrease the training duration of the proposed model.

Results: The experimental findings indicate that the presented approach for classifying ligneous leaf diseases can achieve an accuracy of 85.0% above.

Discussion: In summary, the proposed methodology has the potential to serve as a beneficial algorithm for automated detection of leaf diseases in ligneous plants.

KEYWORDS

precision agriculture, transformer, neural networks, machine vision, transfer learning

1 Introduction

The occurrence of leaf diseases in plants holds significant relevance in the field of plant pathology. Severe leaf disease can have detrimental effects on plants, including leaf drying and hindered bud formation. It can weaken the health of the plant and worsen the susceptibility to other diseases Kai et al. (2011); Bo et al. (2019); Xu et al. (2020); Wang et al. (2021; 2022). In addition, the occurrence of fruit leaf disease can lead to a decline in both

the quantity and quality of fruits, as well as increase the vulnerability of nearby plants to infection. Given the strong reliance of the economy on agricultural productivity, the impact of the leaf disease on the environment becomes particularly significant if preventive measures are not implemented in a timely manner. Therefore, the prompt identification of diseases affecting fruit leaves is crucial for human well-being Patil et al. (2017); Afzaal et al. (2021); Mahum et al. (2022). In general, the identification and categorization of leaf diseases have predominantly depended on the human visual system, which is error prone, is time-consuming and labor-intensive. Hence, the implementation of automated leaf disease classification is imperative in the context of fruit production for mitigating both the production and economic losses Sneha and Bagal (2019); JencyRubia and BabithaLincy (2021); Y et al. (2022).

In recent decades, there has been a significant surge in the utilization of machine learning-based algorithms for addressing leaf disease categorization problems. Numerous machine vision algorithms have been proposed to classify illnesses affecting plant leaves. In the study conducted by Singh and Misra (2017), the authors proposed an image segmentation method for the automatic identification and classification of plant leaf diseases, specifically focusing on the minor leaf disease common in pine trees within the United States. The researchers investigated the utilization of several classifier algorithms for the purpose of identifying plant leaf disease. A system for automatic detection of plant disease using image processing techniques was proposed by the authors Mounika and Bharathi (2020). The approach was used for calculation of textural data pertaining to illnesses affecting plant leaves. In their work, Kulkarni and Sapariya (2021) proposed a method to automatically detect and classify leaf illnesses, which encompasses many stages, including image gathering, image pre-processing, segmentation, and classification. In their study, Reddy et al. (2021) employed Support Vector Machine (SVM) and Random Forest algorithms for the purpose of detecting illnesses in leaves. This study compared assessment measures, such as Root Mean Square Error (RMSE), Peak Signal Noise Ratio (PSNR), for the diseaseaffected regions of the leaves to assess their potential impact on agricultural output.

In recent years, deep learning has gained significant interest due to its remarkable achievements in many domains, such as natural language processing (NLP) and machine vision. Consequently, there have been additional advancements in the field of plant leaf disease categorization by the utilization of deep learning models. Liu et al. (2017) introduced a methodology for detecting apple leaf diseases utilizing deep convolutional neural networks (CNNs). The model reported in this study is capable of generating an ample number of diseased images with a deep learning model, AlexNet. The study utilized a dataset including 13,689 images depicting various apple leaf illnesses. The CNN model developed in this research was trained to accurately classify four types of apple leaf diseases. In the study conducted by Anagnostis et al. (2020), a resilient CNN model was developed to address the timely identification of anthracnose, a prevalent fungal disease that affects numerous tree species globally. This model was to use to classify images of plant leaves as either infected or uninfected by anthracnose. The researchers acquired a dataset consisting of grayscale and RGB images. Then, they utilized a rapid Fourier transform to extract characteristics from the images. Finally, to implement the classification task, they employed a CNN model. To effectively identify olive leaf disease, Ksibi et al. (2022) proposed the utilization of ResNet50 and MobileNet models for image feature extraction, employing the technique of feature concatenation. To train the deep learning models employed in this investigation, a dataset including 5,400 images of olive leaves was utilized. These images were acquired from an olive grove using an unmanned aerial vehicle (UAV) equipped with a camera. In their study, Devi et al. (2022) proposed a methodology for the prediction and classification of corn leaf disease. The authors employed transfer learning and the Alexnet model, leveraging the Adaptive Moment Estimation (ADAM) optimizer and the Stochastic Gradient Descent with momentum (SGDM) mechanism. The model was trained and evaluated using a dataset consisting of 5,300 images, which were categorized into four different types: healthy, blight, common rust, and gray leaf spot. Yao et al. (2022) conducted a study focusing on the identification of kiwifruit leaf disease. They developed a publicly available dataset while using the YOLOX target detection algorithm to mitigate the influence of environmental elements. The study of Yu et al. (2022) introduced a method for efficiently detecting soybean illnesses. It leverages a residual attention network (RANet) model. This study included the incorporation of three types of soybean leaf spot diseases, namely soybean brown leaf spot, soybean frog eye leaf spot, and soybean phyllosticta leaf spot, into the dataset. The OTSU algorithm was utilized to pre-process the initial images for eliminating the surrounding features. Additionally, the image dataset was augmented by the application of image enhancement algorithms. Additionally, the residual attention layer was constructed by integrating attention processes into a ResNet18 model.

The majority of the preceding approaches in the field of leaf disease classification have predominantly employed CNN architectures. Regrettably, the CNN-based models have limitations due to the local receptive field inside the convolutional modules. This characteristic directs attention towards the surrounding region in an image, perhaps overlooking the connections between distant pixels. In contrast, the transformer is renowned for its utilization of an attention mechanism to effectively capture and represent the extensive inter-dependencies within the data samples. The successful performance of transformer in NLP tasks has resulted in its integration and use in the field of computer vision Liu et al. (2021). For instance, the work conducted by Qian et al. (2022) introduced a novel strategy for classifying maize leaf diseases using a vision transformer-based method. The authors of the study also gathered RGB images from publicly available databases and experimental fields, classifying them into four distinct categories: southern corn leaf blight, gray leaf spot, southern corn rust, and healthy specimens. Nevertheless, the vision transformer model proposed in this study might provide challenges when used to high-resolution images due to the quadratic computational complexity of the self-attention mechanism in relation to image resolution. Furthermore, the original vision transformer necessitates a substantial allocation of memory capacity and processing resources.

Taking the aforementioned research into consideration, we propose a hierarchical vision transformerbased approach by employing transfer learning strategy, for classifying leaf diseases of

ligneous plants. The hierarchical design in the proposed vision transformer yields notable reductions in computational resource requirements and the number of weighting parameters for the vision transformer. Furthermore, this work utilizes the weighting factors that were pre-trained on the dataset ImageNet Russakovsky et al. (2014). To assess the effectiveness of the suggested methodology, a subset of a publicly available dataset was utilized. This subset comprises a total of 22 types of ligneous leaf images. Furthermore, a series of comparative tests were carried out to evaluate the performance of the suggested methodology as well as the state-of-the-art methods. The experimental findings provide evidence that the suggested methodology outperforms the state-of-the-art techniques in terms of accuracy, precision, recall, and, F1 score.

In general, the contributions of this study include:

- A leaf disease classification pipeline is proposed. The proposed model primarily consists of a hierarchical vision transformer.
- The presented vision transformer model comprises of two channels, which are used to extract the features from the original leaf images and the edges in the corresponding images, respectively.
- The experimental findings prove the superiority of the proposed methodology over the state-of-the-art algorithms.

The subsequent sections of this article are structured in the following manner. Section 2 presents an elaborate exposition of the suggested transformer concept. Section 3 provides a detailed account of the experimental methodology employed in this study, as well as the subsequent findings and their analysis. Finally, The study concludes at Section 4.

2 Methodology

2.1 Dataset collection and pre-processing

The dataset utilized in this research is sourced from the publicly accessible plant dataset of AI Challenger 2018 Wu et al. (2017), which has a total of 10 plant specimens, each classified into one of 27 categories representing either leaf diseases or healthy conditions. In a systematic manner, a total of 61 image classes have been categorized into distinct groups based on species, pest species, and severity levels. The objective of this work is to categorize diseases affecting ligneous fruit leaves. Therefore, only the leaves that were affected by diseases were selected from the dataset for the purposes of training and validation. In this study, a total of 22 categories of images depicting leaf diseases were included in the dataset. These categories encompassed both sick leaves and healthy leaves.

As seen in Figure 1, the training set comprises 11,603 images, whereas the testing set consists of 1,668 images. These images are categorized into 22 distinct classes. Furthermore, the dataset includes a collection of example images, as seen in Figure 1. These images encompass both healthy and sick leaves.

In this study, the utilization of transfer learning is employed to improve the performance of the proposed approach, taking inspiration from the work of Chen et al. (2020). To achieve this, the proposed model is initially trained on the ImageNet dataset Russakovsky et al. (2014), considering the relatively small size of the presented image dataset. In addition, the images are resized into a uniform dimension of 224×224 to minimize the computing resources needed during the training phase. Moreover, the present study employs a set of data augmentation techniques to increase the number of image samples, which can further enhance the generalization of the proposed model and mitigate the risk of over-fitting during the training process. These techniques include RandomFlip, Color Jitter, Cutmix Yun et al. (2019), and Mixup Zhang et al. (2017).

2.2 Overall framework

The proposed vision transformer model is provided in Figure 2, which is a typical two-channel swin vision transformer Liu et al. (2021) model, and there is no weighting parameter sharing between these two channels.

As seen in Figure 3, the input of the lower channel is achieved by the utilization of the Sobel operator Liu and Wang (2022) and the continuous image fusion operation. The edge Sobel operator is employed on the original image in order to provide input for the suggested methodology. Initially, the gray-scale equivalent is derived from each original image. Next, the original image undergoes convolution with the Sobel operators of size 3×3 in both the horizontal and vertical axes. The specific characteristics of the horizontal and vertical Sobel operators, denoted as G_x and G_y respectively, are outlined below in Equations 1 and 2.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \times I, \tag{1}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \times I, \tag{2}$$

where the original image is taken as I, and let G_x be equal to the transpose of G_y. It is worth noting that the elements in the operators G_x and G_y are differentiable. The starting values of the convolutional layer, also known as the Sobel operator layer, are determined by the elements in the G_x and G_y operators. These values may be optimized by a back-propagation approach during the training phase of the proposed transformer. In addition to combining the output of these two channels through concatenation, the classification process involves the utilization of a softmax classifier, an average pooling layer, and a fully-connected layer.

2.2.1 Details of the backbone

As seen in Figure 2, the configuration of blocks in each channel and the size of tokens may be adjusted to accommodate diverse scales of machine vision applications. In accordance with the present investigation, the quantity of blocks in each channel is

FIGURE 1
A collection of sample images depicting various types of leaf diseases. The leaves in the top row exhibit signs of good health. The leaves exhibiting signs of illness are seen in the bottom row.

multiplied by a factor of 2, 2, 6, and 2, respectively. Following the input technique, the input image is initially partitioned into non-overlapping patches of size 4×4. Hence, the feature dimension of a single patch may be calculated as the product of its width, height, and number of color channels, resulting in a value of 48 (where 3 represents the number of RGB channels). In a manner akin to the vision transformer proposed by Dosovitskiy et al. (2020), the approach involves treating each patch as a token, where the feature representation of a token is obtained by concatenating the pixel values inside the associated patch. Different from the original vision transformer, the proposed transformer model leverages the swin transformer block and the shift-window self-attention mechanism.

In the initial stage, a linear embedding layer is employed to project the original feature into a dimension of arbitrary size (C=96 in the context of this work). Next, a series of swin transformer

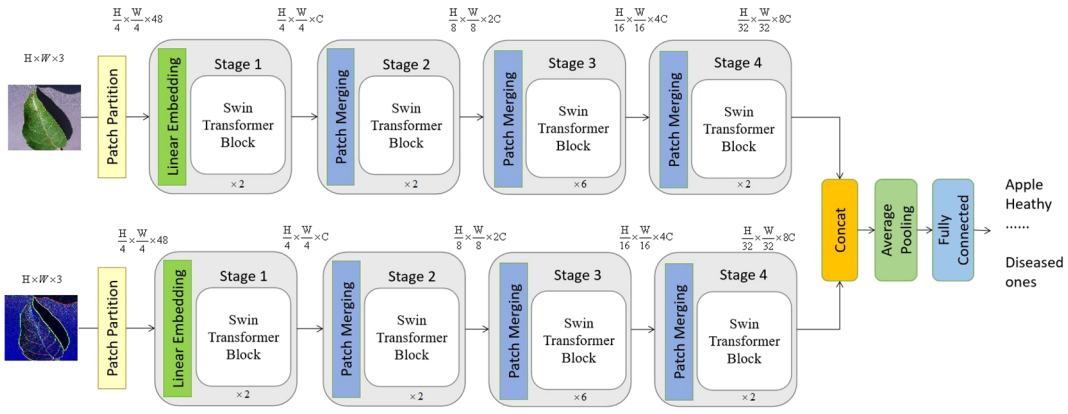


FIGURE 2
The suggested model consists of a two-channel swin vision transformer, which exhibits a certain overall structure. The top channel of the proposed model receives an initial image as its input, while the lower channel gets the edge information of the original image as its input. It is worth noting that the value of C, which is equal to 96, might vary depending on the architecture of the model.

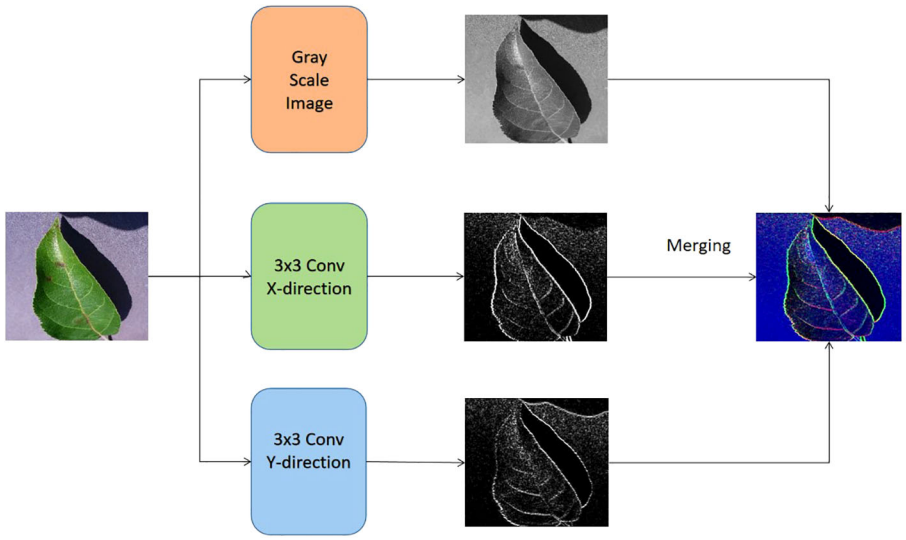


FIGURE 3
The formation of the input for the bottom channel of the proposed vision transformer model.

blocks are utilized on the tokens, incorporating two distinct forms of self-attention modules. Furthermore, it should be noted that the number of tokens in the swin transformer blocks stays consistent with the linear embedding unit, which is calculated as $\frac{H}{4} \times \frac{W}{4}$.

The hierarchical representation is generated by the provided model through the utilization of patch merging modules, which effectively down-sample the feature resolutions by a factor of 2. The first merger module and feature modification are denoted as Stage 2, which are then repeated as Stage 3 and Stage 4. Furthermore, the dimensions of the output features progress from Stage 1 to Stage 4 as $\frac{H}{4} \times \frac{H}{4} \times C$, $\frac{H}{8} \times \frac{H}{8} \times C$, $\frac{H}{16} \times \frac{H}{16} \times C$, and $\frac{H}{32} \times \frac{H}{32} \times C$, respectively. The hierarchical representation is primarily distinguished between the swin vision transformer Liu et al. (2021) and the original vision transformer Dosovitskiy et al. (2020) by the inclusion of Stage 2, Stage 3, and Stage 3 together. The given methodology does not include the utilization of any class taken. In this approach, the output vector of dimensions $N = \frac{H}{32} \times \frac{W}{32}$ is generated by using global average pooling followed by a fully-connected layer. The linear classifier then takes into account the first C components of this output vector.

2.2.2 Swin transformer block

Each stage of the proposed model consists of the swin transformer blocks. And each swin transformer block consists of consecutive modules, as shown in Figure 4. In this architecture, there are two important modules W-MSA and SW-MSA, which represent the multi-head self-attention (MSA) with a standard window and the MSA with a shifted window, respectively.

The mathematical representation of the consecutive swing transformer modules can be articulated in Equations 3–6:

$$\hat{z}^{l} = W - MSA(LN(z^{l-1})) + z^{l-1}, \tag{3}$$

$$z^{l} = MLP(LN(\hat{z}^{l})) + \hat{z}, \tag{4}$$

$$\hat{z}^{l+1} = SW - MSA(LN(z^{l})) + z^{l}, \tag{5}$$

$$z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1}, \tag{6}$$

where the notation W-MSA refers to window-based MSA, MLP stands for multiple layer perception Tolstikhin et al. (2021), SW-MSA represents shifted-window MSA, and LN signifies layer normalization Ba et al. (2016).

2.2.3 Shifted window-based self-attention mechanism

In contrast to the initial vision transformer that heavily relies on global self-attention, which necessitates calculating the relationships between a token and all other tokens, the window-based MSA module employs a window of size $M \times M$ (with a default value of M=7) to restrict the extent of calculation. Hence, the computational complexity becomes more manageable with the incorporation of the window-based self-attention mechanism, as opposed to the quadratic complexity of the vision transformer Dosovitskiy et al. (2020), which is dependent on the image resolution $h \times w$ (as shown in Equations 7, 8).

$$\Omega(MSA) = 4hwC2 + 2(hw)^2C, \tag{7}$$

$$\Omega(W - MSA) = 4hwC^2 + 2M^2hwC, \tag{8}$$

where h and w denote the height and width of an image, C=96, and M=7 in the following settings.

Furthermore, the SW-MSA strategy is intended to enhance the encoding of global relationships among the pixels in multiple windows. The use of the relationship across many windows may be maximized with the introduction of SW-MSA. As seen in Figure 5, the partitioning method of the regular window is employed in layer l, where self-attention is computed within each window. In the subsequent layer, denoted as l + 1, the partitioning of the window is adjusted both

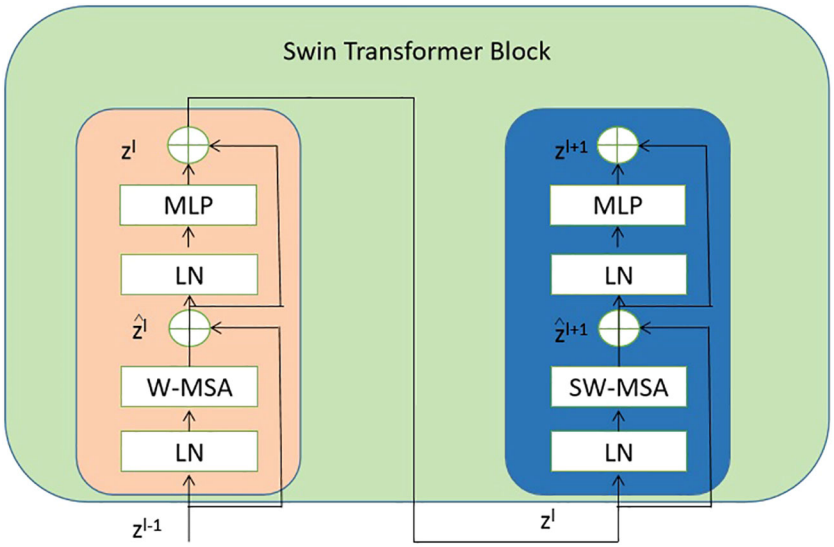


FIGURE 4
The detailed components inside the Swin Transformer model. The abbreviation LN is used to refer to layer normalization. The normal and shifted-windows multi-head self-attention modules are denoted as W-MSA and SW-MSA, respectively. The acronym MLP stands for multiple-layer perception. .

horizontally and vertically, resulting in the creation of a greater number of distinct windows. Thus, the self-attention calculation in Layer l+1 traverses the initial windows in Layer l.

It should be noted that the loss function employed in the proposed model is a cross-entropy loss. This loss is computed by comparing the ground truth category of the image with the classification output given by the suggested model, as seen in Figure 2.

3 Experiments

3.1 Implementation details

The tests were done employing four NVIDIA RTX 3080 GPUs, the PyTorch deep learning framework Paszke et al. (2019) version 2.0.1, and the Python programming language version 3.8.3. The backbone of the suggested model consists of the Swin-T vision transformer, which is employed for each channel. The dimensions of the input images are standardized to 224×224. Furthermore, the suggested swin vision transformer was initialized using the pre-trained weighting parameters of ImageNet Russakovsky et al. (2014). Typically, the hyper-parameters employed in the experiments encompass the subsequent elements, as shown in Table 1 To note that the experiments by using the proposed approach were conducted in a 10-fold cross-validation scheme. Meanwhile, the hyper-parameters were determined by using a trial-and-error strategy.

In order to assess the effectiveness of the suggested model and the comparison methodologies, the experiments contained several assessment measures, including accuracy, precision, recall, and F1 score (as shown in Equations 9–12).

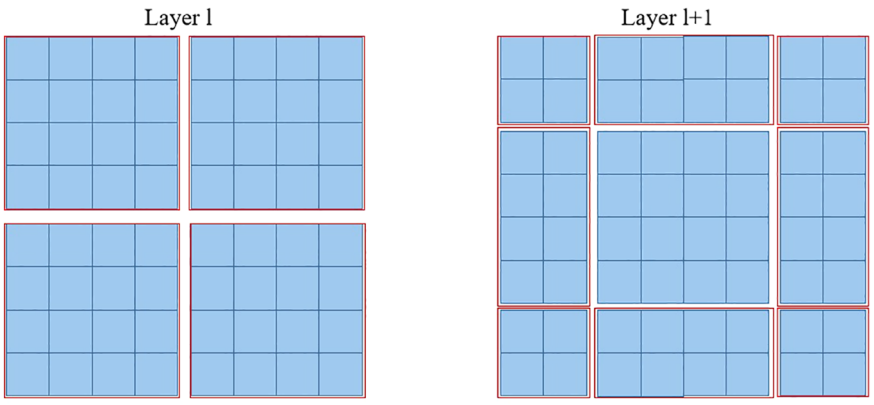


FIGURE 5
The diagram depicting the SW-MSA mechanism employed in the proposed methodology. The red boxes are used to indicate the local window, which serves the purpose of constraining the scope of self-attention calculation.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}, \quad (9)$$

$$Precision = \frac{TP}{TP + FP}, \quad (10)$$

$$Recall = \frac{TP}{TP + FN}, \quad (11)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall,} \quad (12)$$

where TP, TN, FP, and FN denote number of true positive, true negative, false positive, and false negative, respectively.

3.2 Ablation study

The proposed vision transformer model incorporates two distinct topologies for swin vision transformers. To evaluate the efficacy of the introduced swin vision transformer, a series of ablation experiments were conducted on a publicly available dataset. These experiments involved varying the settings of the introduced models, which were used to replace the original settings of the proposed approach. The original approach consisted of a vision transformer Dosovitskiy et al. (2020) and the Sobel operator with fixed 3×3 values.

As seen in Figure 6, it is evident that the accuracy of the suggested methodology surpasses that of the model utilizing the original vision transformer or the fixed Sobel operator. The transformer model under consideration has demonstrated a performance improvement of 2.2% and 1.4% compared to the vision transformer version and the fixed Sobel operator version, respectively, when evaluated on a subset comprising 25% of the utilized dataset. Furthermore, the transformer model under consideration has demonstrated a performance improvement of 2.22% and 1.40% compared to the vision transformer version and the fixed Sobel operator version, respectively, when evaluated on 50% of the identical dataset. Hence, the selected model was deemed suitable as the foundational framework for the subsequent investigations.

3.3 Experimental results

To evaluate the performance of the proposed approach in a fair manner, the comparison experiments were conducted between the state-of-the-art methods, including, and ours on the same dataset as provided in Table 2.

In order to objectively assess the performance of the proposed approach, a series of comparative experiments were conducted. These experiments involved benchmarking the proposed approach against several state-of-the-art methods, namely AlexNet Krizhevsky et al. (2012), GoogleNet Szegedy et al. (2014), VGG Abas et al. (2018), ResNet101 Zhang (2021), EfficientNetB3 Singh et al. (2022), Inception V3 Jenipher and Radhika (2022), MobileNet V2140 Elfatimi et al. (2022), and vision transformer Dosovitskiy et al. (2020). In the experiments, these state-of-the-art methods adopted their original settings in the literature. To note that the former seven state-of-the-art algorithms are CNN models. And the proposed approach was inspired by the work of the last model vision transformer. Meanwhile, the evaluation was carried out on the dataset specified in Table 2.

As seen in Table 3, the suggested strategy exhibits superior accuracy, precision, recall, and F1 score compared to existing state-of-the-art approaches. To provide specific results, our method demonstrates an increase in overall accuracy of 2.1% when compared to MobileNet V2140. Additionally, our proposed approach exhibits improvements in Precision, Recall, and F1 score

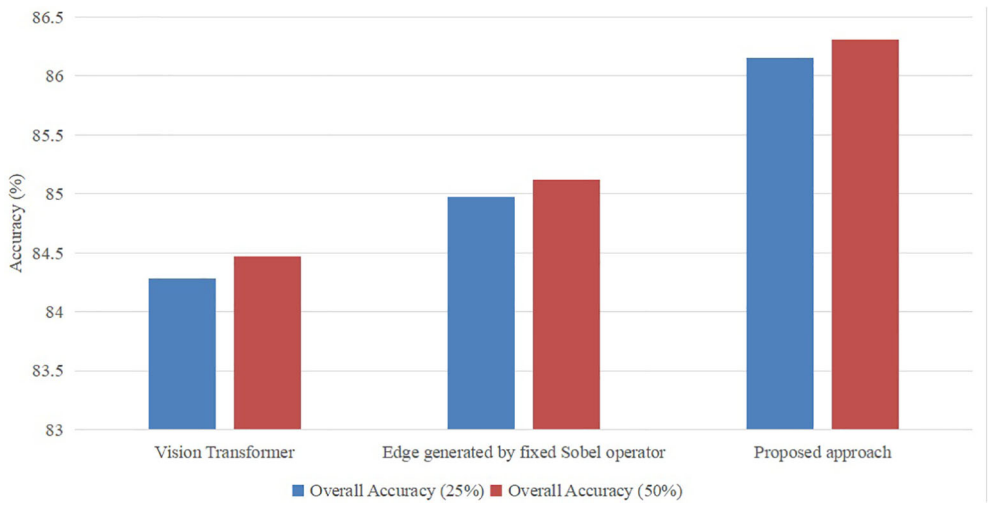


FIGURE 6
Ablation study with different settings with ratios (25% and 50%) of the training set in the publicly available dataset.

TABLE 1 Hyper-parameters used in the experiments.

Item	Value
Batch_size	8
optimizer	Adam
learning rate	1e-4
depth	12
epochs	100

TABLE 2 Distribution of the images in the dataset of this study.

Class	Label Name	No. of training images	No. of testing images
1	Apple healthy	1,185	169
2	Apple_Scab general	211	30
3	Apple_Scab serious	152	22
4	Apple Frogeye Spot	427	61
5	Cedar Apple Rust general	142	20
6	Cedar Apple Rust serious	40	6
7	Cherry healthy	598	85
8	Cherry_Powdery Mildew general	116	12
9	Cherry_Powdery Mildew serious	110	18
10	Grape healthy	294	42
11	Grape Black Rot Fungus general	381	54
12	Grape Black Rot Fungus serious	462	66
13	Grape Black Measles Fungus general	503	74
14	Grape Black Measles Fungus serious	419	59
15	Grape Leaf Blight Fungus general	61	9
16	Grape Leaf Blight Fungus serious	630	90
17	Citrus healthy	367	52
18	Citrus Greening June general	1,828	269
19	Citrus Greening June serious	1,799	262
20	Peach healthy	251	36
21	Peach_Bacterial Spot general	857	122
22	Peach_Bacterial Spot serious	770	110
–	Total	11,603	1,668

by 2.6%, 2.7%, and 2.7% respectively, when compared to MobileNet V2140. Furthermore, even when compared to the original vision transformer, our approach showcases enhancements in accuracy, Precision, Recall, and F1 score by 1.1%, 1.5%, 0.59%, and 1.1% respectively. In summary, the suggested methodology demonstrates higher performance compared to both CNN-based and vision transformer-based algorithms. This provides evidence of the prospective capability of the proposed technique in feature extraction.

In order to assess the effectiveness of the suggested methodology on various image categories within the leveraging dataset, we have included the accuracy-based confusion matrix (as seen in Figure 7) for the proposed technique. This matrix pertains to the 22 categories of leaf disease images inside the public dataset. The majority of the categories have demonstrated encouraging outcomes. The leaf disease images that exhibit inadequate classification pertain to the plant species "Apple" and "Citrus." The category labeled as "Citrus healthy" can sometimes be mistaken with the category known as "Citrus Greening June general." The attribution of the resemblance between various forms of leaf diseases is warranted. Another challenging classification assignment involves distinguishing between "Apple_Scab general" and "Apple_Scab serious." This phenomenon may be ascribed to the existence of two distinct variants of an image falling under the overarching classification of "Apple_Scab."

In addition, the T-distributed stochastic neighbor embedding (t-SNE) was implemented using the suggested methodology, as seen in Figure 8, van der Maaten and Hinton (2008). It should be noted that t-SNE is a computational approach employed for the purpose of visualizing the multidimensional feature space of the 22 categories of sick leaves in a two-dimensional (2D) format. Figure 8 presents a summary of the t-SNE clustering outcomes for both the output produced by the suggested technique and the ground truth. Figure 8 exhibits a notable clustering pattern as classes 16 and 17 are closely packed together on the right side. It should be noted that the distinct attributes of these leaf images can only be ascribed to a limited number of locations that are outside the clusters.

3.4 Discussion

The utilization of CNN models in deep learning has become prevalent. These models possess the capability to extract feature maps from images. Furthermore, the effectiveness of feature extraction may be enhanced by employing a network structure with increased depth. Nevertheless, the efficacy of CNNs may be limited due to the inherent constraint of the convolutional module, which primarily emphasizes the analysis of small receptive fields inside the images. This phenomenon rapidly results in the disregard of the interconnections among distant pixels within an image. In addition, the process of enhancing the performance of deeper convolutional neural network models necessitates a greater allocation of processing resources.

TABLE 3 Comparison results between the state-of-the-arts and the proposed method.

Method	Accuracy	Precision	Recall	F1 score
AlexNet	78.51	77.63	80.16	78.87
GoogleNet	81.23	80.59	82.05	81.31
VGG	82.35	82.19	82.94	82.56
ResNet101	83.18	82.56	83.29	82.92
EfficientNetB3	83.25	83.03	83.48	83.25
Inception V3	84.01	83.23	84.33	83.78
MobileNet V2140	84.69	83.52	84.92	84.21
Vision Transformer	85.47	84.38	86.71	85.53
Our method	**86.43**	**85.73**	**87.22**	**86.47**

Bold values denote the best performance.

In the context of leaf disease images, it is observed that the affected regions are frequently dispersed over the whole image, rather than being confined to a specific localized location. This characteristic is exemplified in Figure 9. Given the limitations of the local receptive field mechanism in addressing the specific leaf disease image, the mere addition of extra layers to the CNN models does not always ensure improved performance in image classification. This study presents the introduction of a vision transformer-based model for image classification, which leverages the relationships among distant pixels inside the images. The suggested dual channel model employs the technique of MSA to continually extract the correlation between image patches. This approach effectively preserves the information that is advantageous for classification purposes. In contrast to the original vision transformer model, the swin vision transformer model is capable of extracting valuable information from images while concurrently mitigating its computing resource requirements. Nevertheless, this research endeavor is subject to many constraints: The dataset utilized in the experiments suffers from unbalanced image samples, hence limiting the effectiveness

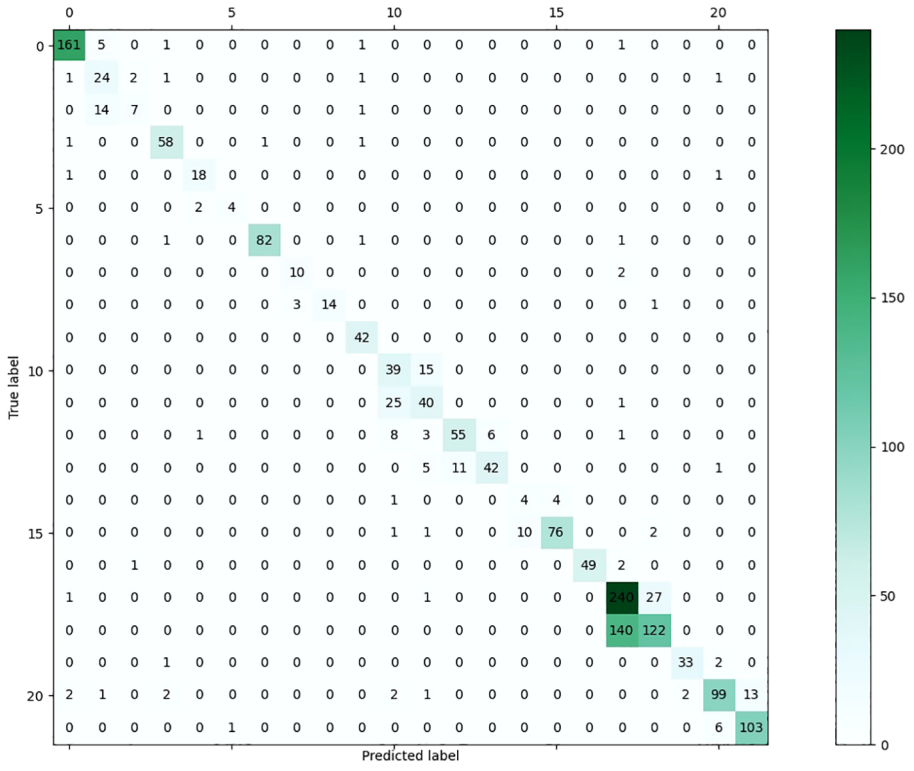


FIGURE 7
The confusion matrix of the proposed approach on the presented dataset.

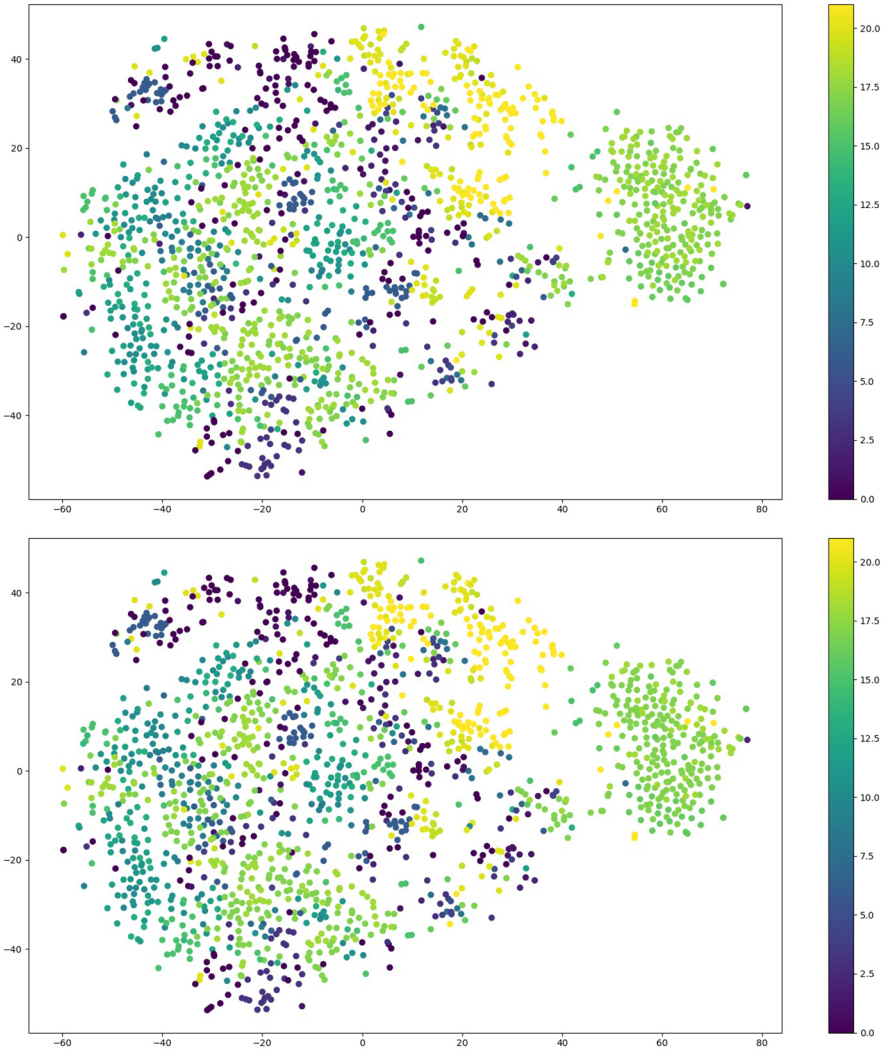


FIGURE 8
The outcome of performing t-SNE on outcome generated from the proposed approach (Top) and on the ground truth samples (Bottom).

of the presented method. Meanwhile, number of the image samples contained in the leveraged dataset is still limited, which constrains the accuracy of the proposed approach at relatively low level. In addition, there exists duplication between the edge information included in the lower channel of the proposed model and the upper channel.

4 Conclusion

The present study introduces a novel network architecture for leaf disease image classification, utilizing a two-channel swin transformer-based approach. The system consists of a dedicated channel for the original image and an additional channel specifically intended to capture the edges in the merged image. In addition, the Sobel operator has been utilized to extract the edge information from the images of leaf diseases. The utilization of the two-channel swin vision transformer model has resulted in the attainment of improved performance compared to the current state-of-the-art methods. The efficacy of the suggested model is demonstrated by experimental findings conducted on the publically accessible dataset. The experimental results of the proposed approach have proved the superior performance of the proposed approach in leaf disease classification. It can be concluded that the proposed approach could be a valuable algorithm for leaf classification and Precision Agriculture.

Recently, there has been encouraging performance demonstrated by vision transformer-based models in challenges

FIGURE 9
A sample image contains the diseased areas across the entire image.

related to multi-modal machine vision. Henceforth, we shall further explore the intricacies of multi-model-based deep learning models in the context of leaf disease categorization and prediction. In addition, more samples need to be collected to eliminate the class imbalance issue in the dataset used in this study.

Data availability statement

Publicly available datasets were analyzed in this study. The datasets for this study can be found in the AI challenger 2018 at https://aistudio.baidu.com/datasetdetail/76075.

Author contributions

DH: Conceptualization, Data curation, Formal Analysis, Investigation, Visualization, Writing – original draft. CG: Funding acquisition, Methodology, Project administration, Supervision, Validation, Writing – original draft.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abas, M. A. H., Ismail, N., Yassin, A. I. M., and Taib, M. N. (2018). Vgg16 for plant image classification with transfer learning and data augmentation. *Int. J. Eng. Technol.* 7 (4), 90–94. doi: 10.14419/IJET.V7I4.11.20781

Afzaal, H., Farooque, A. A., Schumann, A. W., Hussain, N., McKenzie-Gopsill, A., Esau, T. J., et al. (2021). Detection of a potato disease (early blight) using artificial intelligence. *Remote. Sens.* 13, 411. doi: 10.3390/rs13030411

Anagnostis, A., Asiminari, G., Papageorgiou, E. I., and Bochtis, D. D. (2020). A convolutional neural networks based method for anthracnose infected walnut tree leaves identification. *Appl. Sci.* 10 (2), 469. doi: 10.3390/app10020469

Ba, J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *ArXiv*. doi: 10.48550/arXiv.1607.0645

Bo, G., Leilei, D., Wei, L., and Bo, L. (2019). Research on maize disease image recognition method based on grabcut algorithms. *J. Chin. Agric. Mechanization*. doi: 10.1155/2021/9110866

Chen, J., Chen, J., fu Zhang, D., Sun, Y., and Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* 173, 105393. doi: 10.1016/j.compag.2020.105393

Devi, K. S. G., Balasubramanian, K., Senthilkumar, C., and Ramya, K. (2022). Accurate prediction and classification of corn leaf disease using adaptive moment estimation optimizer in deep learning networks. *J. Electrical Eng. Technol.* 18, 637–649. doi: 10.1007/s42835-022-01205-0

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*. doi: 10.48550/arXiv.2010.11929

Elfatimi, E., Eryigit, R., and Elfatimi, L. (2022). Beans leaf diseases classification using mobilenet models. *IEEE Access* 10, 1–1. doi: 10.1109/ACCESS.2022.3142817

JencyRubia, J., and BabithaLincy, R. (2021). Detection of plant leaf diseases using recent progress in deep learning-based identification techniques. *J. Eng. Technol. Ind. Appl.* 7, 29–36. doi: 10.5935/jetia.v7i30.768

Jenipher, V. N., and Radhika, S. (2022). "An automated system for detecting rice crop disease using cnn inception v3 transfer learning algorithm," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. (Piscataway, NJ: IEEE) 88–94.

Kai, S., Zhikun, L., Hang, S., and Chunhong, G. (2011). "A research of maize disease image recognition of corn based on bp networks," in *Third International Conference on Measuring Technology & Mechatronics Automation*. (Piscataway, NJ: IEEE).

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 6084–90. doi: 10.1145/3065386

Ksibi, A., Ayadi, M., Soufiene, B., Jamjoom, M. M., and Ullah, Z. (2022). Mobires-net: A hybrid deep learning model for detecting and classifying olive leaf diseases. *Appl. Sci.* 12 (20), 10278. doi: 10.3390/app122010278

Kulkarni, R. R., and Sapariya, A. D. (2021). Detection of plant leaf diseases using machine learning. *10th International Conference on Computing, Communication and Networking Technologies*, (Piscataway, NJ: IEEE). doi: 10.1109/ICCCNT45670.2019.8944556

Liu, B., Zhang, Y., He, D., and Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* 10, 11. doi: 10.3390/sym10010011

Liu, W., and Wang, L. (2022). Quantum image edge detection based on eight-direction sobel operator for neqr. *Quantum Inf. Process.* 21, 1–27. doi: 10.1007/s11128-022-03527-4

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Piscataway, NJ: IEEE). 9992–10002.

Mahum, R., Munir, H., un-nisa Mughal, Z., Awais, M., Khan, F. S., Saqlain, M., et al. (2022). A novel framework for potato leaf disease detection using an efficient deep learning model. *Hum. Ecol. Risk Assessment: Int. J.* 29, 303–326. doi: 10.1080/10807039.2022.2064814

Mounika, R., and Bharathi, P. S. (2020). Detection of plant leaf diseases using image processing. *Agricultural Food Sci.*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *ArXiv*. doi: 10.48550/arXiv.1912.01703

Patil, P., Yaligar, N., and M, M. (2017). "Comparision of performance of classifiers - svm, rf and ann in potato blight disease detection using leaf images," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. (Piscataway, NJ: IEEE). 1–5.

Qian, X., Zhang, C., Chen, L., and Li, K. (2022). Deep learning-based identification of maize leaf diseases is improved by an attention mechanism: Self-attention. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.864486

Reddy, P. C., Chandra, R. M. S., Vadiraj, P., Reddy, M. A., Mahesh, T. R., and Madhuri, G. S. (2021). "Detection of plant leaf-based diseases using machine learning approach," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. (Piscataway, NJ: IEEE). 1–4.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2014). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Singh, V. B., and Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* 4, 41–49. doi: 10.1016/j.inpa.2016.10.005

Singh, R., Sharma, A., Anand, V., and Gupta, R. (2022). "Impact of efficientnetb3 for stratification of tomato leaves disease," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*. (Piscataway, NJ: IEEE). 1373–1378.

Sneha, H., and Bagal, S. (2019). Detection of plant leaf diseases using image processing. *Int. J. Advance Res. Innovative Ideas Educ.* 5, 168–170. doi: 10.31838/jcr.07.06.310

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., et al. (2014). "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Piscataway, NJ: IEEE). 1–9.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., et al. (2021). "Mlp-mixer: An all-mlp architecture for vision," in *Neural Information Processing Systems*. (Cambridge, USA: MIT Press).

van der Maaten, L., and Hinton, G. E. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605.

Wang, G., Wang, J., Yu, H., and Sui, Y. (2022). Research on identification of corn disease occurrence degree based on improved resnext network. *Int. J. Pattern recognition Artif. Intell.* 36. doi: 10.1142/S0218001422500057

Wang, G., Yu, H., and Sui, Y. (2021). Research on maize disease recognition method based on improved resnet50. *Mobile Inf. Syst.* doi: 10.1155/2021/9110866

Wu, J., Zheng, H., Zhao, B., Li, Y., Yan, B., Liang, R., et al. (2017). Ai challenger: A large-scale dataset for going deeper in image understanding. *ArXiv*. doi: 10.48550/arXiv.1711.06475

Xu, Y., Hao, Q., Qiao, L., and Wang, S. (2020). "Study on classification of maize disease image based on fast k-nearest neighbor support," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. (Piscataway, NJ: IEEE).

Y, V., Billakanti, N., Veeravalli, K. ,. N., A. D., R., and Kota, L. (2022). "Early detection of casava plant leaf diseases using efficientnet-b0," in *2022 IEEE Delhi Section Conference (DELCON)*. (Piscataway, NJ: IEEE).1–5.

Yao, J., Wang, Y., Xiang, Y., Yang, J., Zhu, Y., Li, X., et al. (2022). Two-stage detection algorithm for kiwifruit leaf diseases based on deep learning. *Plants* 11. doi: 10.3390/plants11060768

Yu, M., Ma, X., Guan, H., Liu, M., and Tao, Z. (2022). A recognition method of soybean leaf diseases based on an improved deep learning model. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.878834

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. J. (2019). "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Piscataway, NJ: IEEE). 6022–6031.

Zhang, Q. (2021). A novel resnet101 model based on dense dilated convolution for image classification. *SN Appl. Sci.* 4. doi: 10.1007/s42452-021-04897-7

Zhang, H., Cissé, M., Dauphin, Y., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *ArXiv*. doi: 10.48550/arXiv.1710.09412

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 17 January 2024
DOI 10.3389/fpls.2023.1324491

OPEN ACCESS

EDITED BY
Jian Lian,
Shandong Management University, China

REVIEWED BY
Jana Shafi,
Prince Sattam Bin Abdulaziz University, Saudi Arabia
Ning Lu,
Southwest Forestry University, China

*CORRESPONDENCE
Lina Yang
lnyang@gxu.edu.cn

RECEIVED 19 October 2023
ACCEPTED 20 December 2023
PUBLISHED 17 January 2024

CITATION
Yuan Y, Yang L, Chang K, Huang Y, Yang H and Wang J (2024) DSCA-PSPNet: Dynamic spatial-channel attention pyramid scene parsing network for sugarcane field segmentation in satellite imagery. *Front. Plant Sci.* 14:1324491. doi: 10.3389/fpls.2023.1324491

DSCA-PSPNet: Dynamic spatial-channel attention pyramid scene parsing network for sugarcane field segmentation in satellite imagery

Yujian Yuan[1,2], Lina Yang[1,2]*, Kan Chang[1], Youju Huang[3], Haoyan Yang[1] and Jiale Wang[1]

[1]School of Computer, Electronics, and Information, Guangxi University, Nanning, China, [2]Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer, Electronics, and Information, Guangxi University, Nanning, China, [3]Guangxi Institute of Remote Sensing of Natural Resources, Nanning, China

Sugarcane plays a vital role in many global economies, and its efficient cultivation is critical for sustainable development. A central challenge in sugarcane yield prediction and cultivation management is the precise segmentation of sugarcane fields from satellite imagery. This task is complicated by numerous factors, including varying environmental conditions, scale variability, and spectral similarities between crops and non-crop elements. To address these segmentation challenges, we introduce DSCA-PSPNet, a novel deep learning model with a unique architecture that combines a modified ResNet34 backbone, the Pyramid Scene Parsing Network (PSPNet), and newly proposed Dynamic Squeeze-and-Excitation Context (D-scSE) blocks. Our model effectively adapts to discern the importance of both spatial and channel-wise information, providing superior feature representation for sugarcane fields. We have also created a comprehensive high-resolution satellite imagery dataset from Guangxi's Fusui County, captured on December 17, 2017, which encompasses a broad spectrum of sugarcane field characteristics and environmental conditions. In comparative studies, DSCA-PSPNet outperforms other state-of-the-art models, achieving an Intersection over Union (IoU) of 87.58%, an accuracy of 92.34%, a precision of 93.80%, a recall of 93.21%, and an F1-Score of 92.38%. Application tests on an RTX 3090 GPU, with input image resolutions of 512 × 512, yielded a prediction time of 4.57ms, a parameter size of 22.57MB, GFLOPs of 11.41, and a memory size of 84.47MB. An ablation study emphasized the vital role of the D-scSE module in enhancing DSCA-PSPNet's performance. Our contributions in dataset generation and model development open new avenues for tackling the complexities of sugarcane field segmentation, thus contributing to advances in precision agriculture. The source code and dataset will be available on the GitHub repository https://github.com/JulioYuan/DSCA-PSPNet/tree/main.

KEYWORDS

deep learning, precision agriculture, remote sensing, D-scSE, PSPNet, satellite imagery, sugarcane field segmentation

1 Introduction

Sugarcane, accounting for approximately 70% of the world's sugar production (Shield, 2016) and serving as a substantial source of biofuel, is a crop with considerable economic and environmental consequences (Cardona et al., 2010; Sindhu et al., 2016). The crop's relevance extends beyond its nutritional and energy contributions, playing an integral part in global energy security and economic stability (Moraes et al., 2015; Shield, 2016; Som-Ard et al., 2021). The escalating global population and concurrent amplification of energy demands necessitate the enhancement of sugarcane cultivation efficiency and yield optimization (Li and Yang, 2015; Som-Ard et al., 2021; Tabriz et al., 2021).

In recent years, remote sensing technology has emerged as a potent game-changer in agriculture. Its ability to provide comprehensive, accurate, and timely data is significantly altering traditional agricultural practices (Khanal et al., 2020; Weiss et al., 2020; Omia et al., 2023). This technology is particularly influential in major sugarcane-producing countries like Brazil, India, and China, where it has been instrumental in economic development and energy security (dos Santos Luciano et al., 2018; Jiang et al., 2019; Som-Ard et al., 2021). One of the key applications of remote sensing in agriculture is crop field segmentation (Sun et al., 2022; Ji et al., 2023), a process critical to various agricultural management strategies, including crop health monitoring, yield estimation, and resource allocation (Huan et al., 2021; Wang et al., 2022; Ji et al., 2023). Given its substantial downstream impacts on agricultural decision-making, achieving high accuracy levels in this operation is crucial.

To address this critical need, multiple techniques have been implemented in crop field segmentation and mapping using remote sensing data. For instance, one notable approach used a boundary-semantic-fusion deep convolution network (BSNet) to delineate farmland parcels from high-resolution satellite images, enhancing the F1 and Intersection over Union (IoU) scores (Shunying et al., 2023). An innovative open-source tool, HS-FRAG, has demonstrated its robustness by using an object-based hybrid segmentation algorithm for delineating agricultural fields, particularly in fragmented landscapes (Duvvuri and Kambhammettu, 2023). An edge detection model premised on a connectivity attention-based approach and a high-resolution structure network has been designed for farmland parcel extraction. The model introduces a post-processing method to connect disconnected boundaries, thereby enabling the generation of more refined farmland parcels (Xie et al., 2023). Similarly, a technique called the Multiple Attention Encoder-Decoder Network (MAENet) was proposed for farmland segmentation, yielding an impressive IoU score of 93.74% and a Kappa coefficient of 96.74% (Huan et al., 2021). (Bian et al., 2023) proposed CACPU-Net, linked crop type mapping with 2D semantic segmentation based on single-source and single-temporal autumn.

Sentinel-2 satellite images, achieving excellent classification accuracy on the parcel boundary. (Lu et al., 2023) proposed a multi-scale feature fusion semantic segmentation model for crop classification in high-resolution remote sensing images, providing a good reference for high-precision crop mapping and field plot extraction, while avoiding excessive data acquisition and processing.

Advancements in crop field segmentation have closely paralleled innovations in the broader arena of semantic segmentation techniques. Initially, pioneering work like the Fully Convolutional Network (FCN) introduced by (Long et al., 2015) broke new ground by replacing the conventional fully connected layer in CNNs with a convolutional layer for image segmentation. This led to alternative frameworks such as SegNet, developed by (Badrinarayanan et al., 2017), which further refined the architecture by eliminating the fully connected layer of VGGNet (Simonyan and Zisserman, 2014) and obviating the need for training during the up-sampling process. However, these early models were hampered by limitations, notably in contextual image comprehension and small object recognition, which gave rise to classification errors. Addressing these issues, the Unet model proposed by (Ronneberger et al., 2015) improved segmentation through multi-scale down-sampling and up-sampling fusion channels. To enhance global context information coherence, the Pyramid Scene Parsing Network (PSPNet) model was introduced by (Zhao et al., 2017), featuring a pyramid pooling module. Meanwhile, (Yu and Koltun, 2015) innovated by introducing dilated convolution into the traditional convolution kernel. Yet, the stacking of dilated convolutions with the same dilation rate led to information loss. The hybrid dilated convolution was proposed to address this, combining the benefits of hole convolution while reducing information loss (Wang et al., 2018). In the same vein, the DeepLab series, including V1, V2, V3, and V3+, focused on the study of dilated convolution (Chen et al., 2017a; Chen et al., 2017b). A notable advancement is the Feature Pyramid Network (FPN), which uses a top-down architecture with lateral connections to build high-level semantic feature maps at all scales (Lin et al., 2017). Recently, there has been a growing concern regarding the computational burden posed by the extensive parameters inherent in traditional semantic segmentation models. This burgeoning challenge has not only increased the demand for computational resources but has also hindered the scalability and real-time deployment of these models in resource-constrained environments. To address these limitations, the research community has directed its focus toward the development of efficient and fast semantic segmentation models (Zhang et al., 2023). One pioneering effort in this direction is the introduction of the "squeeze & excitation" mechanism in fully convolutional networks, which emphasizes channel-wise feature recalibration to adaptively emphasize informative features while suppressing less useful ones (Roy et al., 2018). This approach has been further enhanced by the Convolutional Block Attention Module (CBAM), a flexible and lightweight module that can be seamlessly integrated into any CNN architecture. CBAM refines feature maps spatially and channel-wise, ensuring that the model pays selective attention to vital regions in the input data (Woo et al., 2018). Similarly, the Squeeze-and-Excitation Networks propose a novel architectural unit that dynamically adjusts channel-wise feature responses based on the interdependencies between channels, leading to a substantial boost in model performance without considerable computational overhead (Hu et al., 2018). Collectively, these advancements reflect the broader trend in the field to optimize model efficiency without compromising accuracy, ensuring that

semantic segmentation models remain applicable and effective in diverse real-world scenarios.

While semantic segmentation models have made impressive strides, their application to farmland segmentation, particularly in the case of complex crops like sugarcane, still faces a host of challenges. The quest for consistent precision in farmland segmentation, particularly for complex crops such as sugarcane, is fraught with significant challenges (Som-Ard et al., 2021). Factors including fluctuating light conditions, variations in agricultural landscapes, disparities in field sizes, and evolving crop phenology add layers of complexity to these tasks (Khanal et al., 2020; Weiss et al., 2020; Omia et al., 2023). Therefore, it is imperative to develop robust, advanced techniques that can overcome these obstacles and deliver accurate sugarcane field segmentation.

To this end, the present study introduces an innovative deep learning architecture for the segmentation of sugarcane fields, incorporating a modified ResNet34 backbone with the PSPNet and the proposed Dynamic Squeeze-and-Excitation Context (D-scSE) blocks. This proposed model efficiently addresses the complex challenges inherent in sugarcane field segmentation, outperforming traditional techniques and standard deep learning architectures. Moreover, given the importance of high-quality training data in deep learning applications, our research also contributes a novel dataset derived from high-resolution satellite imagery of Guangxi's Fusui County in December. This dataset presents a comprehensive spectrum of environmental conditions and sugarcane field features, representing a realistic testing ground for our model and future similar applications.

The remainder of this paper is organized as follows: Section 2 details the study area, dataset characteristics, and the methodological framework underpinning our research, including the development and refinement of the DSCA-PSPNet architecture. Section 3 presents the findings from our extensive experiments, offering both qualitative and quantitative analyses of the model's performance. In Section 4 we explore the implications of our findings, address the limitations of the current study, and outline potential avenues for future research. Finally, Section 5 synthesizes the key contributions of our work, highlighting its significance in the context of precision agriculture and its broader impact on sustainable farming practices.

In essence, the contributions of this study are threefold:

1) The study introduces an innovative deep learning model specifically engineered for sugarcane field segmentation. Utilizing a unique combination of a modified ResNet34 backbone with PSPNet and proposed novel D-scSE blocks, our model is equipped to effectively navigate through the complexities of remote sensing in agricultural landscapes.
2) The utilization and contribution of a distinctive dataset, comprised of satellite imagery from Guangxi's Fusui County in December, stands as a valuable asset. The data capture the rich diversity of environmental conditions in the region, thus presenting a robust testing bed for our model and a valuable resource for the wider research community.
3) Our model stands apart in its performance, outperforming existing state-of-the-art segmentation techniques. Tested rigorously against established models, our approach demonstrates superior accuracy and robustness, establishing a new benchmark in sugarcane field segmentation.

2 Materials and methods

2.1 Study sites and data

2.1.1 Study area

The study area is in Fusui County (As shown in Figure 1), Guangxi Zhuang Autonomous Region, China, which is situated between latitudes 22°30′N and 22°47′N and longitudes 107°62′E and 107°96′E. This region is known for its extensive sugarcane production, accounting for a significant portion of the country's sugarcane output. The climate in Fusui County is classified as a subtropical monsoon climate, characterized by hot and humid summers, mild winters, and abundant rainfall, which provides suitable conditions for sugarcane cultivation.

The landscape in this area consists of diverse terrain, including flatlands, riverbanks, and karst hills, which pose challenges for accurate sugarcane field segmentation. The complex terrain may lead to variations in the spectral signature of sugarcane fields, as well as the presence of shadows, mixed pixels, and other occlusions. Furthermore, the study area includes a range of land cover types, such as cropland, forests, water bodies, and urban areas, which can create difficulties in distinguishing sugarcane fields from other land cover types.

2.1.2 Datasets

High-resolution RGB satellite images were acquired from the BJ-2 satellite on December 18th, 2017 for the study area. The images have a spatial resolution of 0.8 meters, which is suitable for identifying and segmenting individual sugarcane fields at a fine scale. Twenty remote sensing images of size 4096×4096 pixels^2 were selected for this study. The selected images provide a comprehensive representation of the landscape diversity and phenological stages of sugarcane fields in the region. The exact locations of these selected images are marked in Figure 1.

As shown in Figure 2, the images were acquired during cloud-free conditions, with minimal atmospheric haze, to ensure optimal image quality for the analysis. Additionally, the images were chosen to represent various landscape features and land cover types present in the study area, including diverse terrain, riverbanks, agricultural lands, and urban areas. This selection strategy aimed to provide a robust dataset that could effectively capture the challenges associated with accurate sugarcane field segmentation in a complex and dynamic environment.

2.1.3 Data quality and preprocessing

To uphold data integrity and uniformity in this study, we embarked on a rigorous preprocessing regimen for the satellite imagery acquired from the Guangxi Institute of Natural Resources Remote Sensing (GXINRRS). These high-resolution images,

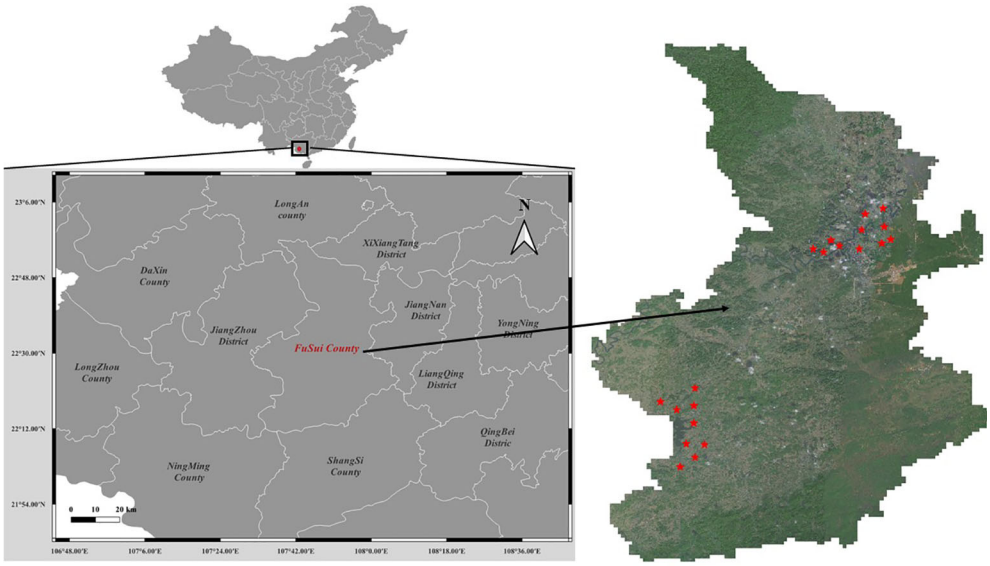


FIGURE 1
Study area.

captured by the BJ-2 satellite, underwent a comprehensive preprocessing protocol, including atmospheric correction, radiometric calibration, and geometric correction, using ENVI software.

The atmospheric correction stage involved adjusting specific parameters in the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes module, accounting for aerosol optical thickness, precipitable water vapor, and atmospheric pressure. This step ensured the minimization of atmospheric distortions, thereby enhancing the representation of the ground reflectance. During the radiometric calibration phase, the sensor's radiometric response function and the incident solar irradiance at the time of acquisition were factored in. This calibration converted the raw digital numbers in the images into standardized reflectance values, ensuring their consistent representation across different scenes. Lastly, geometric correction rectified any image distortions due to sensor geometry, Earth's curvature, and terrain relief, utilizing the satellite's ephemeris data, Earth's ellipsoid and datum information, and a digital elevation model for terrain correction. This step facilitated the accurate portrayal of spatial relationships among features in the images.

2.1.4 Ground truth data collection

The collection and verification of ground truth data for this study was an intricate and meticulous process involving collaboration with local agricultural experts, geography workers,

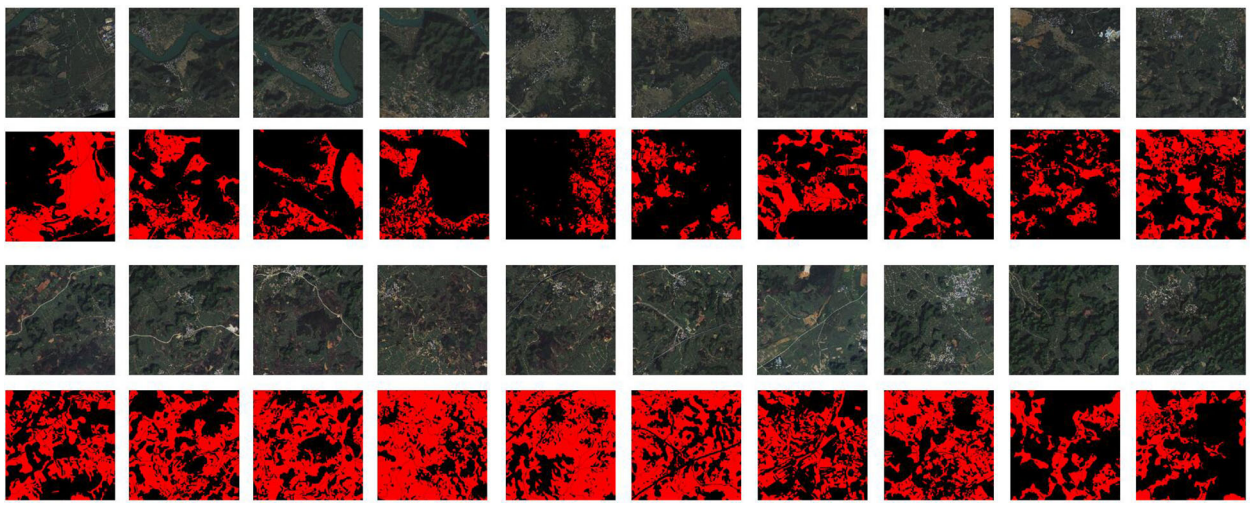


FIGURE 2
Selected images.

and sugarcane experts. The methodology was designed to ensure accurate segmentation of sugarcane fields and robust training data for the deep learning model.

The following steps were taken in the process of ground truth data collection:

1) Image Acquisition and Preprocessing: We obtained BJ-2 satellite images of Fusui County, Guangxi, from GXINRRS and performed the preprocessing techniques mentioned in section 2.3. These images captured a diverse range of environmental conditions.
2) Expert Annotation: Agricultural and sugarcane experts and geography workers manually annotated the acquired images using ArcGIS software. They drew polygons around the sugarcane fields and delineated them by hand-drawing, utilizing their deep knowledge of local agriculture to identify these regions accurately.
3) Cross-Verification: After the initial annotation, the annotated images were cross-checked by a separate team of geography workers. They scrutinized the annotations, ensuring the masks accurately represented sugarcane fields.
4) Review and Revision: Any images that were flagged during cross-verification underwent a review and revision process. The original experts and the verification team collaborated to resolve discrepancies, resulting in a final, agreed-upon annotation.
5) Final Dataset Formation: Once all images had been annotated and verified, they were compiled into the final dataset. With its carefully validated ground truth labels, this dataset was then used for training, validating, and evaluating the proposed deep learning model.

This rigorous process, while time-consuming, was necessary to ensure the high quality and reliability of our ground truth data. This process's collaborative and iterative nature also served to minimize human error and bias.

2.1.5 Closer look at selected images and annotated masks

To provide a comprehensive understanding of the study area and the inherent complexities it presents for sugarcane field segmentation, we examine specific images from our dataset, displayed collectively in Figure 3.

Figure 3 presents a comprehensive view of three different landscapes and their corresponding segmentation maps, identified as (A), (B), and (C). In column (A), a river area is captured with features including a riverbank, karst hills, and sugarcane fields. This image presents the challenge of segmenting sugarcane fields that are intertwined with riverbanks, where water and vegetation boundaries are often indistinct. The corresponding ground truth for this area serves as the benchmark for our segmentation task. Column (B) depicts a living area with buildings, karst hills, and sugarcane fields. This scenario emphasizes the intricacy of segmenting sugarcane fields near urban structures, where the line between built and natural environments can be ambiguous. The corresponding ground truth, excluding the small roads, trees, bushes, and reaped sugarcane fields, helps in accurately distinguishing between the urban structures and natural

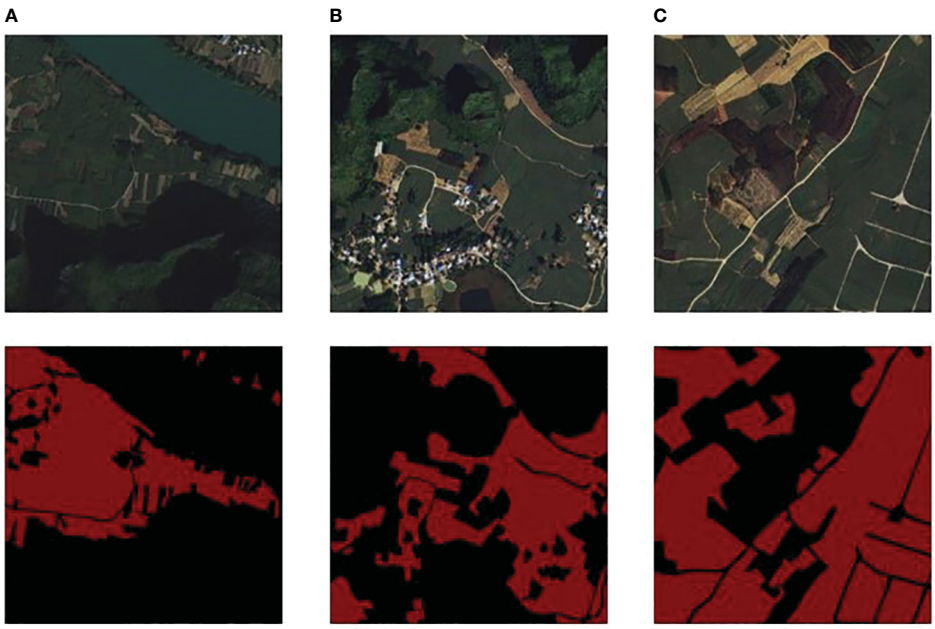


FIGURE 3
(A) River area and ground truth; **(B)** Resident area and label; **(C)** Farmland area and label.

vegetation. Lastly, column (C) portrays a farmland teeming with mixed crops and sugarcane fields. This scene highlights the difficulty of distinguishing sugarcane fields from other crop types and non-crop vegetation, which often share overlapping spectral characteristics, making the task of segmentation more complex. The corresponding ground truth excluded small roads, reaped sugarcane fields, and other non-sugarcane vegetation, which aids in deciphering the diverse crops present in the image.

Together, these images underscore the diverse challenges encountered during sugarcane field segmentation in our study area. They highlight the necessity for an advanced deep learning approach, one that is capable of grappling with these complexities and delivering precise and reliable segmentation outcomes. The source code and dataset will be available on the GitHub repository https://github.com/JulioYuan/DSCA-PSPNet/tree/main.

2.2 DSCA-PSPNet

2.2.1 Backbone comparison

In the domain of semantic segmentation tasks, particularly for complex applications like sugarcane field segmentation from satellite images, the choice of backbone architecture substantially influences the overall model performance. For this study, we exclusively used PSPNet as the segmentation decoder, with the focus of our experimentation being on selecting the most efficient and accurate backbone. We considered six popular architectures, namely ResNet34, ResNet50 (He et al., 2016), VGG16 (Simonyan and Zisserman, 2014), EfficientNet-B5 (Tan and Le, 2019), MobileNet-V3Large (Howard et al., 2019), and ViT-B/16 (Vision Transformer) (Dosovitskiy et al., 2020), to serve as the backbone.

Experiments were carried out using the dataset and experiment settings elaborated in sections 3.3.1 and 3.3.2. The backbone architectures were compared based on metrics such as IoU, F1 scores, prediction time for a single 512x512 RGB image, number of parameters, and memory footprint. The results are concisely tabulated in Table 1:

Based on our comprehensive evaluation, ResNet34 emerges as the most suitable backbone architecture for sugarcane field segmentation when paired with the PSPNet decoder. With a prediction time of 3.98ms, it not only facilitates real-time inference but also operates with a manageable number of parameters (21.44M), thereby making it amenable to deployment in resource-constrained environments. Furthermore, its memory requirement is 81.78 MB, while maintaining high IoU and F1 scores, indicative of its accuracy and reliability. Consequently, for the specialized task of semantic segmentation in agricultural settings, the balanced and robust performance of ResNet34 substantiates its selection as the backbone architecture.

2.2.2 Modified ResNet34 backbone

ResNet (He et al., 2016) is a family of deep residual networks that effectively addresses the degradation problem in deep neural networks by introducing residual connections. In this study, we utilize the ResNet34 architecture as our model's backbone, with specific modifications tailored to the task of agricultural crop field segmentation.

As illustrated in Figure 4A, the modified ResNet34 backbone consists of several components. It begins with an input layer, followed by a stem composed of a convolutional layer, batch normalization, and a ReLU activation function. The stem is succeeded by two residual layers, each containing a series of standard residual blocks, as depicted in Figure 4B. These residual layers capture local features in the input images.

The latter part of the backbone includes two dilated layers, with dilated blocks that incorporate dilated convolutions (Yu and Koltun, 2015), as shown in Figure 4C. The dilated blocks allow for a larger receptive field without increasing the number of parameters or computational complexity. The final output layer generates high-level feature maps for the input images.

The modified ResNet34 backbone integrates the advanced D-scSE attention mechanism after each residual layer (layer1, layer2, layer3, and layer4), enhancing channel and spatial dependencies and refining feature representation. The inclusion of the D-scSE mechanism improves the model's ability to capture essential contextual information, leading to more precise segmentation results. A detailed examination of the D-scSE mechanism's design and its role in augmenting the modified ResNet34 backbone will be provided in section 3.3.

The architecture's larger receptive field, achieved by incorporating dilated convolutions in the later stages, is especially beneficial for capturing contextual information in high-resolution agricultural imagery with objects spanning various spatial scales. By incorporating these modifications, the backbone design provides an

TABLE 1 Metrics comparison for different backbones

Methods	IoU	F1-Score	Prediction Time (ms)	Parameters (Million)	Memory Size (MB)
ResNet34	83.18	89.49	3.98	21.44	81.78
ResNet50	81.46	89.31	4.16	24.30	92.70
VGG16	78.85	88.09	4.98	39.34	150.09
EfficientNet-B5	81.17	89.42	7.97	28.41	108.40
MobileNet-V3Large	77.09	86.97	2.98	3.02	11.52
ViT-B/16	81.66	89.76	12.95	24.35	92.89

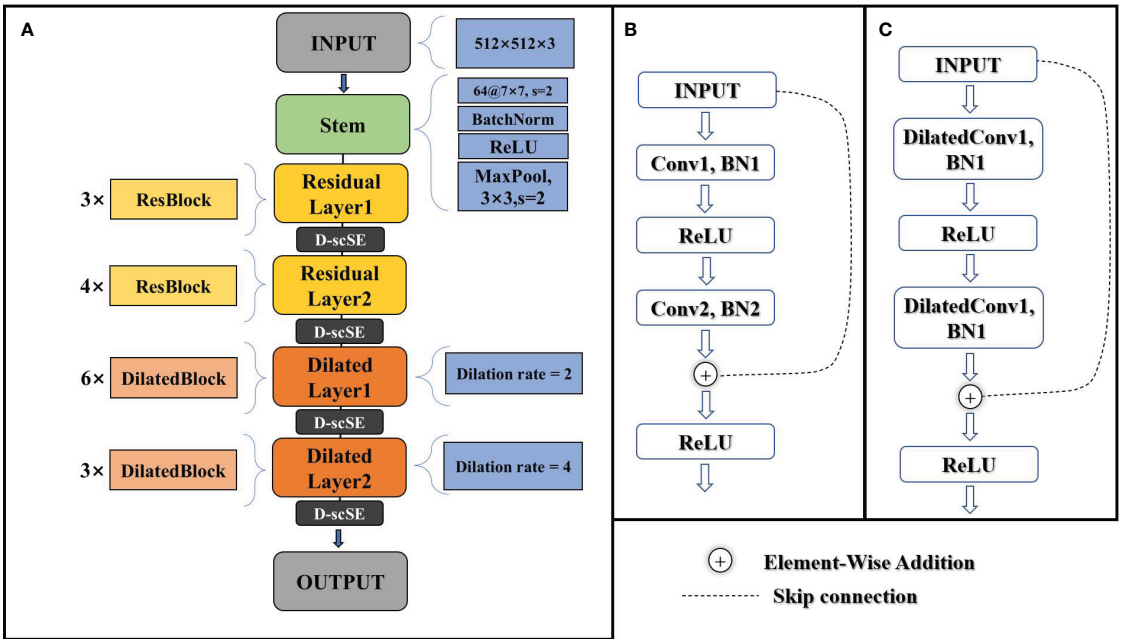


FIGURE 4
(A) schematic of the modified ResNet34 architecture. **(B)** schematic of the standard residual block. **(C)** schematic of the dilated residual block.

effective foundation for the decoder to generate accurate agricultural crop field segmentation maps, addressing the specific challenges of the task and leveraging the power of residual networks, dilated convolutions, and the D-scSE mechanism.

2.2.3 D-scSE block

The D-scSE mechanism, where "D" stands for "Dynamic," is an advanced attention mechanism inspired by the original scSE (Roy et al., 2018). While the original scSE effectively encodes channel and spatial dependencies, it doesn't account for the varying importance of these aspects across different input data or stages of network depth. The importance of spatial and channel-wise features may dynamically vary based on the contextual information in the scene, or the intricacy of the features being learned at different network layers. This limitation could potentially restrict the learning capacity and performance of the original scSE.

To overcome this, the D-scSE mechanism introduces dynamic weights, providing a more adaptive balancing between the significance of spatial and channel-wise information. These weights are learned during the training process, offering the flexibility to modulate the degree of attention applied to the spatial and channel dimensions based on the input's inherent characteristics.

In this section, we will delve into the specifics of the D-scSE's design, its components, and the way it refines feature representation. We'll discuss how this dynamic weighting scheme leads to enhanced feature learning and contributes to the overall efficacy of our proposed model architecture.

1) Channel Squeeze and Spatial Excitation Block (sSE): This block focuses on spatial information, as shown in Figure 5A. The input feature map $U \in R^{C\times H\times W}$ is first channel-wise squeezed using a. 1×1 convolution (Equation 1):

$$\boldsymbol{U} = [\boldsymbol{u}^{1,1}, \boldsymbol{u}^{1,2}, \ldots, \boldsymbol{u}^{i,j}, \ldots, \boldsymbol{u}^{H,W}] \cdot \text{ with } \boldsymbol{u}^{i,j} \in \boldsymbol{R}^{C\times 1\times 1} \quad (1)$$

The spatial squeeze operation computes the output matrix $k \in R^{H\times W}$ (Equation 2):

$$\boldsymbol{k} = \boldsymbol{W}_k \star \boldsymbol{U} \quad (2)$$

where $W_k \in R^{C\times 1\times 1}$ and $\star$ denotes the convolution operation. The spatial information weight is added to the feature map U by applying the sigmoid activation function $(\cdot)$ to each element in k (Equation 3):

$$\begin{aligned}\hat{\boldsymbol{U}}_{sSE} &= \boldsymbol{F}_{scale}(\boldsymbol{U}, \boldsymbol{k}) \\ &= [\sigma(\boldsymbol{k}_{1,1})\boldsymbol{u}^{1,1}, \ldots, \sigma(\boldsymbol{k}_{i,j})\boldsymbol{u}^{i,j}, \ldots, \sigma(\boldsymbol{k}_{H,W})\boldsymbol{u}^{H,W}]\end{aligned} \quad (3)$$

2) Spatial Squeeze and Channel Excitation Block (cSE): This block focuses on channel-wise dependencies, as shown in Figure 5B. The input feature map U is first spatially squeezed using global average pooling and global max pooling (concatenated) before passing them through the convolutional layers (Equation 4):

$$\boldsymbol{x} = \text{Concat}\left(\frac{1}{H\times W}\sum_{i}^{H}\sum_{j}^{W}\boldsymbol{U}(:,\boldsymbol{i},\boldsymbol{j}), \max_{i=1,\ldots,H;j=1,\ldots,W}\boldsymbol{U}(:,\boldsymbol{i},\boldsymbol{j})\right) \quad (4)$$

To discern the dependency information between channels, a single fully connected layer is employed, with weights $W \in R^{C\times 2C}$. Activation of this layer is achieved through the application of the ReLU function $(\cdot)$ and the sigmoid function $\sigma(\cdot)$ (Equation 5):

$$\boldsymbol{s} = \sigma(\boldsymbol{W}\delta(\boldsymbol{x})) \quad (5)$$

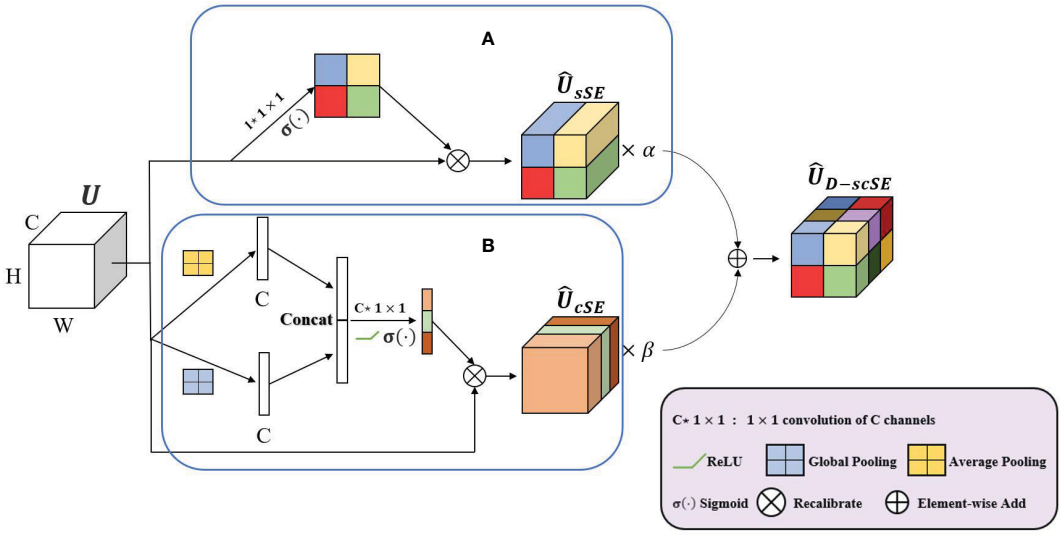


FIGURE 5
A schematic representation of the D-scSE mechanism. **(A)** sSE module and **(B)** cSE module.

The final output is obtained by re-scaling the transformation U (Equation 6):

$$\hat{U}_{cSE} = \mathbf{F}_{\text{scale}}(U, s) = s \star U \tag{6}$$

We introduce dynamic weighting to balance the contributions of the sSE and cSE branches to the final output. The outputs of the sSE and cSE branches are combined (Equation 7):

$$\hat{U}_{D-scSE} = \alpha \hat{U}_{sSE} + \beta \hat{U}_{cSE} \tag{7}$$

where α and β are learnable parameters initialized by sampling from a uniform distribution $U(-\sqrt{6/n}, \sqrt{6/n})$ where n is the number of input units in the weight tensor. These dynamic weights are updated during the training process, allowing the D-scSE module to adaptively balance the importance of spatial and channel information based on the input data.

D-scSE module enhances the original scSE mechanism by integrating dynamic weighting and diversified pooling strategies, as shown in Figure 5. With the sSE branch concentrating on spatial information and the cSE branch addressing channel-wise dependencies, the module effectively recalibrates both dimensions of the feature map. By employing learnable weights, the D-scSE module adeptly balances spatial and channel information, ultimately delivering a robust feature extraction mechanism for the segmentation task.

2.2.4 Pyramid scene parsing network decoder

In our proposed architecture, we utilize the PSPNet decoder, originally introduced by (Zhao et al., 2017), to generate high-quality segmentation results. The decoder effectively captures contextual information from the output feature map of the encoder by leveraging pyramid parsing and fusing multi-scale features. Additionally, the decoder is integrated with the D-scSE mechanism to further refine the feature representation.

The decoder comprises the following components:

1) Pyramid Pooling Module: This module is designed to extract contextual information from the input feature map by applying multiple pooling operations with varying kernel sizes. This approach enables the capture of both local and global context at different scales. The pyramid pooling module consists of four parallel branches, each employing an average pooling layer with a unique kernel size. Subsequently, a 1x1 convolution is used to reduce the number of channels to a predefined number (e.g., C/4). The resulting feature maps are then upsampled to their original spatial dimensions using bilinear interpolation.
2) Feature Concatenation: The upsampled feature maps originating from the pyramid pooling module are concatenated with the initial input feature map, facilitating the fusion of multi-scale contextual information.
3) D-scSE Mechanism: As detailed in Section 3.3, the D-scSE mechanism is incorporated following the feature concatenation step to adaptively recalibrate the spatial and channel-wise information. The inclusion of the D-scSE mechanism within the decoder further refines the feature representation, enabling the model to better manage varying object scales and shapes.
4) Final Convolution Layers: After implementing the D-scSE mechanism, the feature map is processed through a series of convolutional layers to generate the ultimate output segmentation map. This typically consists of one or more

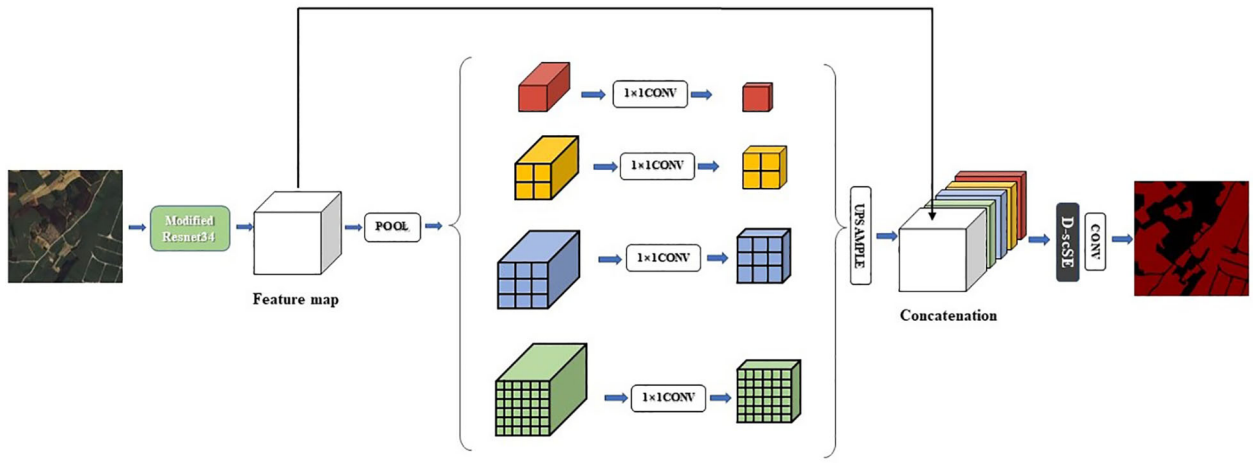


FIGURE 6
Structure of DSCA-PSPNet.

3x3 convolutions, followed by a 1x1 convolution to project the feature map onto the desired number of output classes. The final segmentation map is then upsampled to match the original input image size using bilinear interpolation.

By integrating the PSPNet decoder with the D-scSE mechanism, DSCA-PSPNet (as shown in Figure 6) effectively captures and exploits multi-scale contextual information, thereby enhancing segmentation performance. This decoder design contributes to the generation of more accurate and finer-grained segmentation maps, ultimately improving the overall efficacy of the architecture.

2.3 Experiments

2.3.1 Data preparation and augmentation

To create a diverse and representative dataset for model validation, twenty remote sensing images of size 4096x4096 pixels^2 were selected from the remote sensing images of Fusui County in Guangxi Zhuang Autonomous Area. The locations of the data samples were selected based on the presence of different land features, such as river areas, farmland areas, and living areas.

Each of the twenty original 4096×4096 images was cropped into sixty-four 512×512 images, resulting in a total of 1280 images. This cropping is a standard practice in semantic segmentation tasks, especially when handling high-resolution imagery, to manage GPU memory constraints and optimize computational efficiency. While this approach divides larger sugarcane plots into smaller segments, it does not significantly impact the segmentation task. Our model is designed to accurately classify each pixel within these segments, ensuring effective and reliable segmentation across the cropped images. To ensure a balanced dataset for model training and evaluation, 70% of the cropped images from each original image were allocated to the training set, 15% were assigned to the validation set, and the remaining 15% were assigned to the test set. This partitioning strategy ensured that the training, validation, and test sets contained a diverse range of features and challenges associated with sugarcane field segmentation.

Data augmentation techniques were applied to increase the diversity of the training dataset, making the model more robust and capable of handling real-world scenarios. The augmentation techniques applied to the dataset include rotation, horizontal and vertical flipping, random scaling, random brightness and contrast adjustment, addition of Gaussian noise, Gaussian blur, and hue, saturation, and value adjustment. These augmentations were performed using the Albumentations Python library. For each original training sample, 5 augmented samples were generated by applying all the aforementioned augmentation techniques simultaneously. This resulted in an augmented dataset of 4480 samples. Hence, the distribution of samples among the training, validation, and test sets as shown in Table 2.

TABLE 2 Sample distribution across training, validation, and test sets .

Dataset	Original Images	Augmented Images	Total Images
Training set	896	4480	5376
Validation set	192	0	192
Test set	192	0	192

2.3.2 Experimental design

The experiments conducted in this study, which encompassed the training, validation, and testing of the proposed model, were performed on a system equipped with the Windows 10 System. The experimental runtime environment was set up using Anaconda3, Python 3.10.5, CUDA 11.7, and OpenCV 4.6. The hardware used for the experiments included 64 GB RAM, Intel (R) Core i9-10980XE@3.00GHz processor, and a NVIDIA RTX 3090 GPU. Pytorch was chosen as the deep learning framework for implementing the proposed model.

The purpose of the experiments in this study was to verify the effectiveness of the proposed model, in the recognition of sugarcane field. The fed images were 512×512. The AdamW optimizer, an improvement over traditional Adam by decoupling weight decay from the optimization steps, was utilized to prevent overfitting and achieve faster convergence. The learning rate was controlled using a cyclical learning rate strategy. The base learning rate was set to 0.0001, and it cyclically varied between this value and a maximum of 0.001, facilitating optimal convergence. Other hyperparameters included an epoch count of 100 and a training batch size of 16.

2.3.3 Evaluation metrics

The accuracy, precision, IoU, F1 score, and Recall were calculated (Equations 8–12) and used as the accuracy evaluation indexes of the experimental results in this study, that is,

$$\mathbf{IoU} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP} + \mathbf{FN}} \tag{8}$$

$$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}} \tag{9}$$

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{10}$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{11}$$

$$\boldsymbol{F1} = \mathbf{2} \times \frac{\mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}} \tag{12}$$

where TP denotes positive samples correctly classified by the model, FN denotes positive samples incorrectly classified by the model, FP denotes negative samples incorrectly classified by the model, TN denotes negative samples correctly classified by the model.

Accuracy is depicted as the fraction of pixels that were accurately predicted, in contrast to the total sum of pixels. Precision constitutes an evaluative metric to gauge the accuracy of predictions within a specific category. IoU is a statistical measure that identifies the degree of overlap between the predicted and the original annotated regions within an image. The F1 score is the harmonic mean of precision and recall, serving as a balanced estimator of the classifier's performance. In addition, recall, also known as sensitivity or true positive rate, quantifies the proportion of actual positives that are correctly classified. It is an integral part of the evaluation schema, examining the classifier's proficiency in identifying all the pertinent instances within the dataset.

3 Results

3.1 Contrast experiments

In this revised section, we will first delve into the qualitative analysis through the visual examination of segmentation results and subsequently provide a quantitative examination through the rigorous metric evaluations. Our objective remains to present a coherent and comprehensive comparison of the proposed DSCA-PSPNet with the benchmark models: Unet, DeepLabV3+, FPN, and PSPnet.

Figure 7 displays the segmentation results for a landscape marked by reaped land, sugarcane fields, and river banks. The original images (Figure 7A) elucidate a complex environment where sugarcane fields fringe the river banks, interspersed with fragments of reaped land. The ground truth (Figure 7B) meticulously captures the distinct boundaries between these zones. DSCA-PSPNet (Figure 7G) demonstrates a remarkable alignment with the ground truth, adeptly segment the sugarcane fields from adjacent reaped land and preserving the nuanced contours of the karst hills. In contrast, Unet (Figure 7C) falsely recognizes the karst hills green vegetation as the sugarcane field, blurring the transition between karst hills and sugarcane fields. Deeplabv3+ (Figure 7D) provides a robust segmentation of sugarcane fields, but the delineation of reaped land seems slightly generalized. FPN (Figure 7E) exhibits a slightly better results but the miss segmentations are still existing. PSPnet (Figure 7F) offers balanced performance, although minor miss segmentations are evident, especially in regions where sugarcane fields are situated in the narrow land between river and hills. Collectively, the comparative analysis underscores DSCA-PSPNet's superior capability in effectively segmenting complex riverine landscapes.

Figure 8 offers a detailed segmentation analysis of a landscape primarily characterized by sugarcane fields, reaped land, other vegetation, and minor road networks. The ground truth (Figure 8B) accurately maps out these features, showcasing the stark boundaries between cultivated sugarcane fields, reaped areas, other vegetation, and the intricate web of roads. DSCA-PSPNet (Figure 8G) mirrors this ground truth with impressive precision, successfully delineating the sugarcane fields from reaped patches and capturing the delicate intricacies of the minor roads and other vegetation patches. In comparison, Unet (Figure 8C) occasionally confuses the reaped land with lighter patches of sugarcane fields, leading to minor segmentation inconsistencies. Deeplabv3+ (Figure 8D) effectively segments the larger sugarcane plots but sometimes overlooks the subtle distinction between reaped land and lighter sugarcane fields. FPN (Figure 8E) provides a commendable segmentation but faces challenges in accurately mapping the other vegetations. PSPnet (Figure 8F) produces a balanced segmentation but has minor discrepancies in areas where roads intersect with reaped land and other vegetations. Collectively, the comparative evaluation emphasizes DSCA-PSPNet's robust capability in accurately segmenting a multifaceted farmland environment, highlighting its promise for precision agriculture applications.

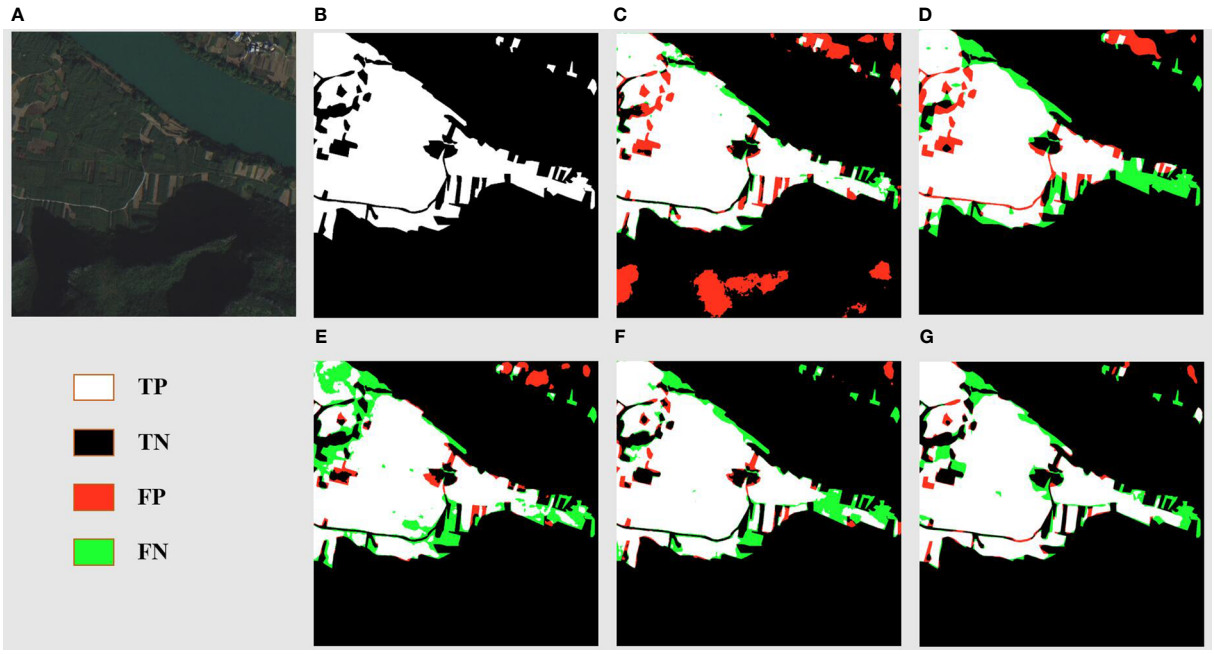


FIGURE 7
River area prediction results of models. **(A)** Original Images. **(B)** Ground Truths. **(C)** Unet. **(D)** Deeplabv3+. **(E)** FPN. **(F)** PSPnet. **(G)** DSCA-PSPNet).

Figure 9 delves into the segmentation of a landscape in the residential zones with sprawling farmland areas. DSCA-PSPNet (Figure 9G) emerges as a standout, replicating the ground truth with exceptional accuracy. It captures the structured layout of residential zones and small roads, and has the minimal miss segmentations in water pond area. In contrast, Unet (Figure 9C) exhibits challenges in accurately segmenting the water pond region. Deeplabv3+ (Figure 9D) adeptly identifies the larger residential blocks but seems to slightly oversimplify the segmentation of smaller farmland patches situated between residential clusters. FPN (Figure 9E) offers a respectable segmentation but shows major miss segmentation in water pond region too. PSPnet (Figure 9F) provides a consistent segmentation but faces minor deviations in areas where dense vegetation in farmlands is

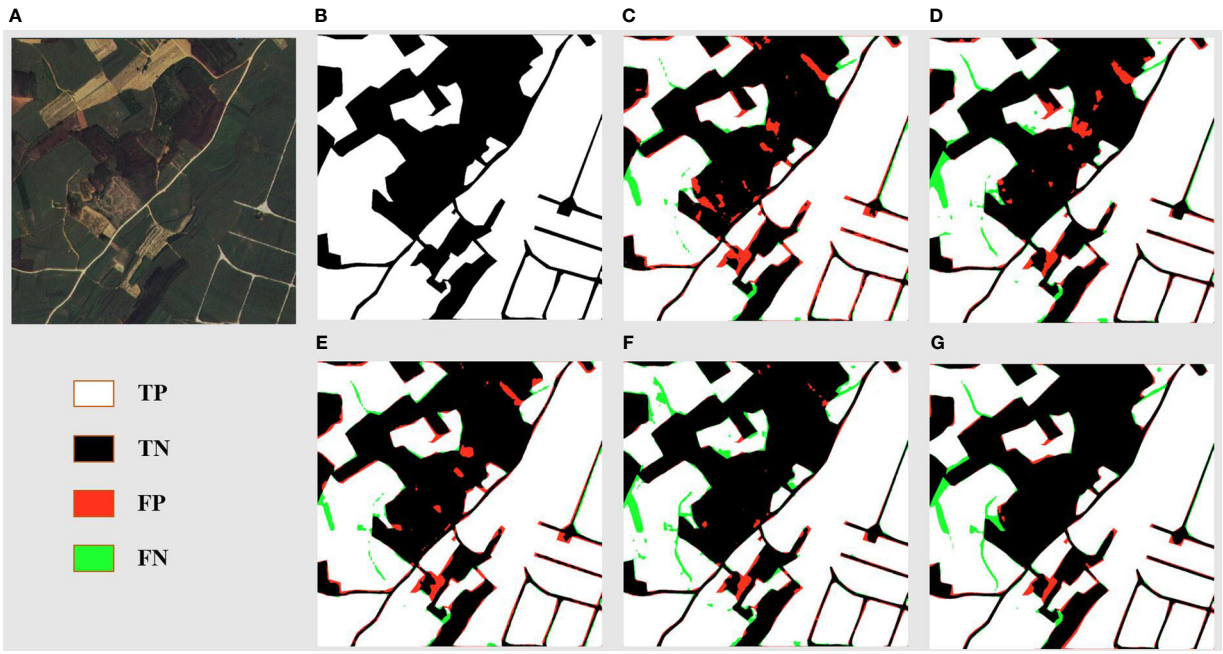


FIGURE 8
Farmland area prediction results of models. **(A)** Original Images. **(B)** Ground Truths. **(C)** Unet. **(D)** Deeplabv3+. **(E)** FPN. **(F)** PSPnet. **(G)** DSCA-PSPNet).

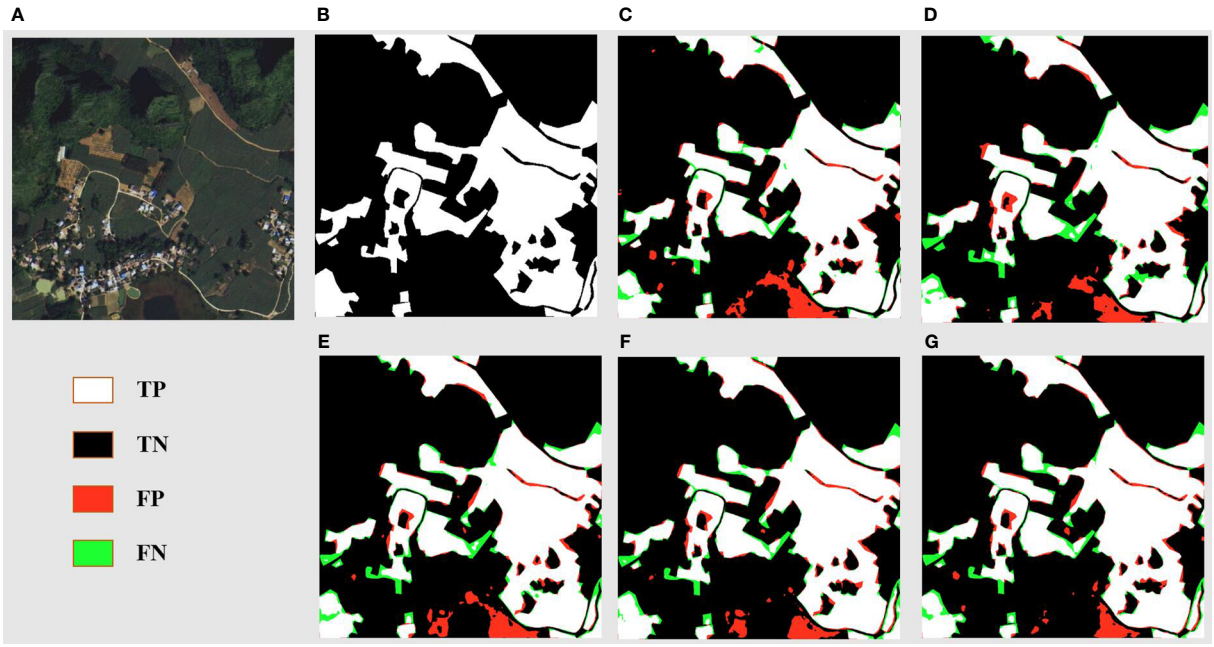


FIGURE 9
Resident and farmland area prediction results of models. **(A)** Original Images. **(B)** Ground Truths. **(C)** Unet. **(D)** Deeplabv3+. **(E)** FPN. **(F)** PSPnet. **(G)** DSCA-PSPNet).

proximal to residential zones. In summation, the analysis underscores DSCA-PSPNet's superior ability in segment the residential and farmland landscapes, showing its power in mixed-use land segmentation tasks.

To complement the qualitative insights underscoring the enhanced performance of DSCA-PSPNet, we shall now transition to a quantitative analysis that empirically substantiates these observations.

Evidently, DSCA-PSPNet stands out across all evaluation metrics, reinforcing its potency as affirmed by the visual outcomes. Specifically, DSCA-PSPNet records an IoU of 87.58%, indicative of its exceptional overlap prediction ability, leading the second-best performer, PSPnet-resnet34, by a significant margin of 4.4%. Its accuracy score of 92.34% is the highest among all models, reflecting the model's impressive capability in classifying each pixel correctly. In terms of precision, DSCA-PSPNet's score of 93.8% further cements its supremacy, signaling its strength in minimizing false positives, outperforming the runner-up, FPN-resnet34, by approximately 0.77%. Additionally, DSCA-PSPNet records a recall of 93.21% and an F1 score of 92.38%. These metrics respectively highlight DSCA-PSPNet's competence in accurately identifying true positives and maintaining a balanced performance between precision and recall.

Shifting focus to computational efficiency and resource consumption in Table 3, DSCA-PSPNet continues to shine. Although its prediction time of 4.57 ms for a single 512 × 512 image on RTX 3090 GPU is slightly slower than PSPnet-resnet34, it outperforms Unet, DeeplabV3+ and FPN considerably. Importantly, with 22.57M parameters, DSCA-PSPNet's model complexity is on par with other models, showcasing that superior performance does not necessitate excessive complexity. Further, DSCA-PSPNet's GFLOPs and memory usage affirm its efficiency, making it apt for deployment in resource-constrained scenarios.

TABLE 3 Accuracy metrics comparison for different segmentation methods.

Methods	IoU	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
Unet (resnet34)	78.44	88.65	84.69	91.90	87.06
DeepLabV3+(resnet34)	81.83	90.62	86.65	<u>92.40</u>	<u>90.31</u>
FPN (resnet34)	79.84	89.79	<u>93.13</u>	84.88	88.59
PSPnet (resnet34)	<u>83.18</u>	<u>92.25</u>	91.64	91.52	89.49
DSCA-PSPNet	**87.58**	**92.34**	**93.80**	**93.21**	**92.38**

Bold value is the highest value.
Underline value is the second highest value.

In conclusion, the comprehensive evaluation presented in this section, through both qualitative and quantitative perspectives, cements the superiority of DSCA-PSPNet in sugarcane field segmentation. Its consistent lead across a variety of performance metrics, the demonstrated visual prowess, and efficient resource utilization collectively mark DSCA-PSPNet as a promising tool in the domain of sugarcane field segmentation and beyond. This underscores the applicability and potential of DSCA-PSPNet for real-world implementation, thus appealing to the academic community and sugarcane practitioners alike.

3.2 Ablation study

The ablation study aims to examine the progression of performance improvements that our proposed DSCA-PSPNet offers, starting from the baseline PSPNet(resnet34), and its variants augmented with sSE and cSE mechanisms, and finally to DSCA-PSPNet. Using sSE and cSE in the same position as the D-scSE in the models, ensures an unbiased and consistent basis for comparison.

A valuable tool in our analysis is the use of attention maps, generated from the output of the final layer of the backbone. This layer, rich with high-level semantic information, provides a detailed visual guide to how different models prioritize areas within an image.

The attention maps in Figure 10, column (A) presents the original images, and columns (B) to (E) show the attention maps for PSPNet, PSPNet+sSE, PSPNet+cSE, and DSCA-PSPNet, respectively. The difference in focus and detail becomes quite evident upon comparison. The baseline PSPNet exhibits less distinct segmentation, while the addition of sSE and cSE mechanisms enhances the model's ability to distinguish different landforms more clearly. Yet, it is with DSCA-PSPNet that we observe the most significant concentration of attention on intricate agricultural details, such as edges and sugarcane fields. This confirms the superior capability of our D-scSE mechanism in capturing both local and global contextual details, enhancing the model's understanding of the image.

Along with visual observations from attention maps, we perform a quantitative analysis on key performance metrics for each model variant, as represented in the tables below:

Tables 4 and 5 shows that DSCA-PSPNet surpasses PSPnet and its sSE and cSE variants in all performance metrics. For example, in terms of IoU, DSCA-PSPNet outperforms the next best model, PSPnet+cSE, by 2.4 percentage points. This pattern continues with Accuracy%(2.09 percentage points higher), Precision% (0.16 percentage points higher), Recall% (1.69 percentage points higher), and F1-Score % (0.89 percentage points higher). These results confirm the effectiveness of the D-scSE module in improving DSCA-PSPNet's performance.

In summary, our ablation study systematically evaluates the performance improvements of DSCA-PSPNet, beginning with the baseline PSPNet (ResNet34) and progressing through its variants augmented with sSE and cSE mechanisms, to the final DSCA-PSPNet model. This study not only quantitatively demonstrates DSCA-PSPNet's superiority over its predecessors but also qualitatively underlines the effectiveness of our design choices, particularly the inclusion of the D-scSE module. By analyzing attention maps generated from the model's final layer, we observed a significantly enhanced focus on critical sugarcane field details, such as field edges and textures, in DSCA-PSPNet compared to the baseline and other

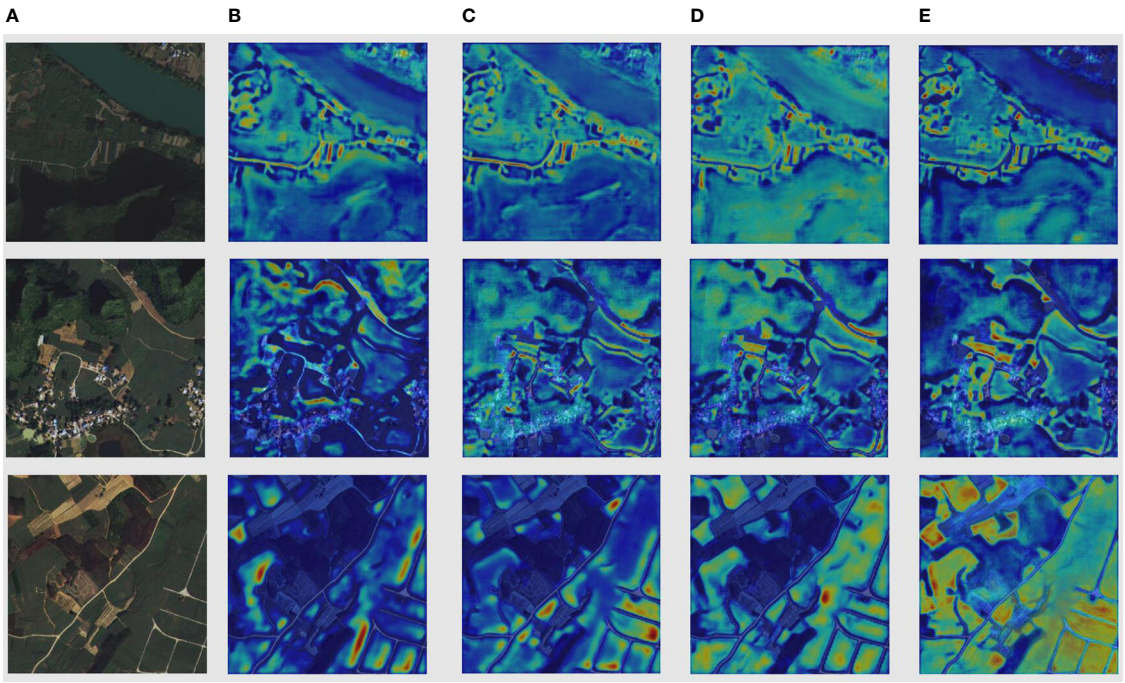


FIGURE 10
Columnnn **(A)** Original images. Columnn **(B)** Attention map of PSPnet. Columnn **(C)** Attention map of PSPnet+sSE. **(D)** Attention map of PSPnet+cSE. **(E)** Attention map of DSCAPSPNet.

TABLE 4 Performance metrics comparison for different segmentation methods.

Methods	Prediction Time (ms)	Parameters (Million)	GFLOPs	Memory Size (MB)
Unet(resnet34)	6.97	24.44	31.36	93.21
DeepLabV3+(resnet34)	5.98	22.44	31.62	85.60
FPN (resnet34)	7.24	23.16	27.49	88.33
PSPnet (resnet34)	**3.98**	**21.44**	**9.41**	**81.78**
DSCA-PSPNet	4.57	22.57	11.41	84.47

Bold value is the highest value.
Underline value is the second highest value.

variants. Quantitative analysis reveals that DSCA-PSPNet surpasses other models in key performance metrics, including IoU, accuracy, precision, recall, and F1-score, confirming the D-scSE module's pivotal role in improving segmentation capabilities. These results collectively highlight the D-scSE module's contribution to the model's overall efficacy in accurately segmenting complex sugarcane cultivation scenes, thereby validating the module's integration as a critical enhancement in our deep learning architecture for precision agriculture applications.

4 Discussion

The primary limitation of the DSCA-PSPNet study is its reliance on a dataset exclusively from Guangxi's Fusui County, captured on a single date. This limitation, while providing high accuracy within its narrow scope, raises concerns about the model's robustness and adaptability to different sugarcane cultivation environments. The challenges in acquiring diverse, high-resolution satellite data, often restricted due to censorship and stringent data-sharing policies, combined with the intensive requirements of accurately labeling such imagery, have led to a lack of dataset diversity (Sing et al., 2021). Consequently, the model's current iteration, although advanced, might not fully account for the variances in sugarcane fields across different geographical locations with varying environmental conditions and agricultural practices. A critical aspect yet to be verified is the model's ability to accurately segment sugarcane fields in different stages of growth, under varying weather conditions, or in regions with distinct soil types (Lin et al., 2009). Addressing these challenges is imperative for future research. Efforts will be concentrated on expanding the model's application to a broader range of sugarcane-producing regions worldwide. For instance, testing DSCA-PSPNet in countries like Brazil and India, which are major sugarcane producers but have different climatic conditions and cultivation practices compared to southern China and south east Asia, would be crucial. This would help assess the model's adaptability and performance in diverse sugarcane farming contexts. Additionally, the examination of the model's performance using multi-temporal satellite imagery is essential. This would offer insights into its capability to consistently recognize sugarcane fields throughout different growth stages and under varying seasonal weather patterns, such as the monsoon impact in South Asia or the dry season in Brazil. Collaborations with international agricultural research institutes, satellite imagery providers, and experts in global sugarcane cultivation could facilitate access to a more varied range of data, overcoming the limitations in data acquisition and labeling. Such collaborative efforts are vital in refining DSCA-PSPNet to address the unique challenges of sugarcane field segmentation in different parts of the world. Enhancing the model's accuracy and versatility in this manner is not only crucial for advancing precision agriculture in the context of sugarcane farming but also has broader implications for sustainable agricultural practices and food security globally.

5 Conclusion

In the pursuit of sustainable agricultural practices, precise and accurate crop field segmentation remains a critical concern. Addressing this need, this study introduces the DSCA-PSPNet, a deep learning model specifically designed for sugarcane field segmentation. The integration of a modified ResNet34 backbone with PSPNet and D-scSE blocks is pivotal to the model's success. The modified ResNet34 backbone, enhanced with dilated blocks, serves as a robust foundation for feature extraction, capitalizing on its deep residual learning framework to circumvent issues like vanishing gradients in deeper networks. These dilated blocks significantly

TABLE 5 Accuracy metrics comparison in ablation study.

Methods	IoU	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
PSPnet(resnet34)	83.18	92.25	91.64	91.52	89.49
PSPnet+sSE	84.76	92.79	92.13	92.88	90.59
PSPnet+cSE	85.18	93.25	93.64	91.52	91.49
DSCA-PSPNet	**87.58**	**92.34**	**93.80**	**93.21**	**92.38**

Bold value is the highest value.

augment the network's capability for feature extraction, enabling the model to cover a wider field of view, thus capturing more contextual information without compromising resolution or incurring additional computational costs (Zhang and Zhang, 2021). The PSPNet component further assists in aggregating contextual information across various scales, crucial for differentiating sugarcane fields from other similar features in satellite imagery. The D-scSE blocks add a dynamic aspect to the model by recalibrating the channel-wise and spatial features in the network, fine-tuning the focus on relevant features for precise segmentation. Together, these elements enable DSCA-PSPNet to effectively navigate the spectral and spatial complexities inherent in agricultural landscapes. This design has enabled the model to achieve an IoU of 87.58%, an accuracy of 92.34%, a precision of 93.8%, a recall of 93.21%, and an F1-Score of 92.38%. These figures demonstrate its superior performance over established models. Moreover, DSCA-PSPNet proves to be computationally efficient, with a memory size of 84.47MB and a model size of 22.57MB.

In addition to developing the model, this study has compiled a comprehensive high-resolution satellite imagery dataset from Guangxi's Fusui County, encompassing a broad spectrum of environmental conditions and field characteristics. This dataset provides a challenging yet realistic testing ground for DSCA-PSPNet, contributing significantly to the validation and refinement of the model. Furthermore, it represents a valuable resource for future research and innovation in the field of agricultural segmentation. The insights gained from this study not only demonstrate the potential of DSCA-PSPNet in sugarcane field segmentation but also highlight the model's adaptability and potential applicability to other crop types. Future research could leverage this model and dataset to explore segmentation in different agricultural contexts, potentially expanding the scope of precision agriculture. By integrating these advances with ongoing research efforts, there is a strong potential for models like DSCA-PSPNet to play a pivotal role in enhancing sustainable farming practices, thereby contributing significantly to global food security and sustainable development goals.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/JulioYuan/DSCA-PSPNet/tree/main.

Author contributions

YY: Conceptualization, Methodology, Validation, Visualization, Writing – original draft. LY: Formal Analysis, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. KC: Formal Analysis, Methodology, Writing – review & editing. YH: Data curation, Resources, Validation, Writing – review & editing. HY: Investigation, Project administration, Writing – review & editing. JW: Data curation, Formal Analysis, Methodology, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China (62171145) (61966003)(62371144) and Natural Science Foundation of Guangxi Province (2020GXNSFAA159171).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoderdecoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Bian, Y., Li, L., and Jing, W. (2023). CACPU-Net: Channel attention U-net constrained by point features for crop type mapping. *Front. Plant Sci.* 13, 1030595. doi: 10.3389/fpls.2022.1030595

Cardona, C. A., Quintero, J. A., and Paz, I. C. (2010). Production of bioethanol from sugarcane bagasse: Status and perspectives. *Bioresour. Technol.* 101, 4754–4766. doi: 10.1016/J.BIORTECH.2009.10.097

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv. Prepr. arXiv1706.05587*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv. Prepr. arXiv2010.11929*.

dos Santos Luciano, A. C., Picoli, M. C. A., Rocha, J. V., Franco, H. C. J., Sanches, G. M., Leal, M. R. L. V., et al. (2018). Generalized space-time classifiers for monitoring sugarcane areas in Brazil. *Remote Sens. Environ.* 215, 438–451. doi: 10.1016/j.rse.2018.06.017

Duvvuri, S., and Kambhammettu, B. V. N. P. (2023). HS-FRAG: An open source hybrid segmentation tool to delineate agricultural fields in fragmented landscapes. *Comput. Electron. Agric.* 204, 107523. doi: 10.1016/j.compag.2022.107523

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019). "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 1314–1324.

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huan, H., Liu, Y., Xie, Y., Wang, C., Xu, D., and Zhang, Y. (2021). MAENet: multiple attention encoder–decoder network for farmland segmentation of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5.

Ji, Z., Wei, J., Chen, X., Yuan, W., Kong, Q., Gao, R., et al. (2023). SEDLNet: An unsupervised precise lightweight extraction method for farmland areas. *Comput. Electron. Agric.* 210, 107886. doi: 10.1016/j.compag.2023.107886

Jiang, H., Li, D., Jing, W., Xu, J., Huang, J., Yang, J., et al. (2019). Early season mapping of sugarcane by applying machine learning algorithms to Sentinel-1A/2 time series data: a case study in Zhanjiang City, China. *Remote Sens.* 11, 861. doi: 10.3390/rs11070861

Khanal, S., Kc, K., Fulton, J. P., Shearer, S., and Ozkan, E. (2020). Remote sensing in agriculture— accomplishments, limitations, and opportunities. *Remote Sens.* 12, 3783. doi: 10.3390/rs12223783

Li, Y.-R., and Yang, L.-T. (2015). Sugarcane agriculture and sugar industry in China. *Sugar. Tech.* 17, 1–8. doi: 10.1007/s12355-014-0342-1

Lin, H., Chen, J., Pei, Z., Zhang, S., and Hu, X. (2009). Monitoring sugarcane growth using ENVISAT ASAR data. *IEEE Transact. Geosci. Remote Sens.* 47 (8), 2572–2580. doi: 10.1109/TGRS.2009.2015769

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (IEEE), 936–944. doi: 10.1109/CVPR.2017.106

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lu, T., Gao, M., and Wang, L. (2023). Crop classification in high-resolution remote sensing images based on multi-scale feature fusion semantic segmentation model. *Front. Plant Sci.* 14, 1196634. doi: 10.3389/fpls.2023.1196634

Moraes, M. A. F. D., Oliveira, F. C. R., and Diaz-Chavez, R. A. (2015). Socio-economic impacts of Brazilian sugarcane industry. *Environ. Dev.* 16, 31–43. doi: 10.1016/j.envdev.2015.06.010

Omia, E., Bae, H., Park, E., Kim, M. S., Baek, I., Kabenge, I., et al. (2023). Remote sensing in field crop monitoring: A comprehensive review of sensor systems, data analyses and recent advances. *Remote Sens.* 15, 354. doi: 10.3390/rs15020354

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention– MICCAI 2015: 18th International Conference, Munich, Germany*, , October 5-9, 2015, Vol. 18. 234–241, *Proceedings, Part III*.

Roy, A. G., Navab, N., and Wachinger, C. (2018). "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference*, Granada, Spain, September 16-20, 2018. 421–429, Proceedings, Part I.

Shield, I. (2016). "Sugar and starch crop supply chains," in *Biomass supply chains for bioenergy and biorefining* (Elsevier), 249–269.

Shunying, W., Ya'nan, Z., Xianzeng, Y., Li, F., Tianjun, W., and Jiancheng, L. (2023). BSNet: Boundary-semantic-fusion network for farmland parcel mapping in high-resolution satellite images. *Comput. Electron. Agric.* 206, 107683. doi: 10.1016/j.compag.2023.107683

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv. Prepr. arXiv1409.1556*.

Sindhu, R., Gnansounou, E., Binod, P., and Pandey, A. (2016). Bioconversion of sugarcane crop residue for value added products–An overview. *Renew. Energy* 98, 203–215. doi: 10.1016/j.renene.2016.02.057

Singh, P., Diwakar, M., Shankar, A., Shree, R., and Kumar, M. (2021). A review on SAR image and its despeckling. *Arch. Computat. Methods Eng.* 28, 4633–4653.

Som-Ard, J., Atzberger, C., Izquierdo-Verdiguier, E., Vuolo, F., and Immitzer, M. (2021). Remote sensing applications in sugarcane cultivation: A review. *Remote Sens.* 13, 4040. doi: 10.3390/rs13204040

Sun, W., Sheng, W., Zhou, R., Zhu, Y., Chen, A., Zhao, S., et al. (2022). Deep edge enhancementbased semantic segmentation network for farmland segmentation with satellite imagery. *Comput. Electron. Agric.* 202, 107273. doi: 10.1016/j.compag.2022.107273

Tabriz, S. S., Kader, M. A., Rokonuzzaman, M., Hossen, M. S., and Awal, M. A. (2021). Prospects and challenges of conservation agriculture in Bangladesh for sustainable sugarcane cultivation. *Environ. Dev. Sustain.* 23, 15667–15694. doi: 10.1007/s10668-021-01330-2

Tan, M., and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 6105–6114.

Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., et al. (2018). "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, 1451–1460.

Wang, H., Chen, X., Zhang, T., Xu, Z., and Li, J. (2022). CCTNet: Coupled CNN and transformer network for crop segmentation of remote sensing images. *Remote Sens.* 14, 1956. doi: 10.3390/rs14091956

Weiss, M., Jacob, F., and Duveiller, G. (2020). Remote sensing for agricultural applications: A metareview. *Remote Sens. Environ.* 236, 111402. doi: 10.1016/j.rse.2019.111402

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Xie, Y., Zheng, S., Wang, H., Qiu, Y., Lin, X., and Shi, Q. (2023). Edge detection with direction guided postprocessing for farmland parcel extraction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* doi: 10.1109/JSTARS.2023.3253779

Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv. Prepr. arXiv1511.07122*.

Zhang, Z., and Zhang, S. (2021). Towards understanding residual and dilated dense neural networks via convolutional sparse coding. *Nat. Sci. Rev.* 3, nwaa159.

Zhang, X., Li, W., Gao, C., Yang, Y., and Chang, K. (2023). Hyperspectral pathology image classification using dimension-driven multi-path attention residual network. *Expert Syst. Appl.* 230, 120615. doi: 10.1016/j.eswa.2023.120615

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 04 March 2024
DOI 10.3389/fpls.2024.1289783

OPEN ACCESS

EDITED BY
José Dias Pereira,
Instituto Politecnico de Setubal (IPS), Portugal

REVIEWED BY
Carlos Banha,
Institute of Telecommunications (IT), Portugal
Haiou Wang,
Nanjing Xiaozhuang University, China

*CORRESPONDENCE
Xia Zheng
zx_mac@shzu.edu.cn

RECEIVED 06 September 2023
ACCEPTED 15 February 2024
PUBLISHED 04 March 2024

CITATION
Yang T, Zheng X, Xiao H, Shan C and Zhang J (2024) Moisture content online detection system based on multi-sensor fusion and convolutional neural network.
Front. Plant Sci. 15:1289783.
doi: 10.3389/fpls.2024.1289783

Moisture content online detection system based on multi-sensor fusion and convolutional neural network

Taoqing Yang[1,2,3], Xia Zheng[1,2,3]*, Hongwei Xiao[4], Chunhui Shan[5] and Jikai Zhang[1,2,3]

[1]College of Mechanical and Electrical Engineering, Shihezi University, Shihezi, China, [2]Key Laboratory of Northwest Agricultural Equipment, Ministry of Agriculture and Rural Affairs, Shihezi, China, [3]Key Laboratory of Modern Agricultural Machinery Corps, Shihezi, China, [4]College of Engineering, China Agricultural University, Beijing, China, [5]College of Food, Shihezi University, Shihezi, China

To monitor the moisture content of agricultural products in the drying process in real time, this study applied a model combining multi-sensor fusion and convolutional neural network (CNN) to moisture content online detection. This study built a multi-sensor data acquisition platform and established a CNN prediction model with the raw monitoring data of load sensor, air velocity sensor, temperature sensor, and the tray position as input and the weight of the material as output. The model's predictive performance was compared with that of the linear partial least squares regression (PLSR) and nonlinear support vector machine (SVM) models. A moisture content online detection system was established based on this model. Results of the model performance comparison showed that the CNN prediction model had the optimal prediction effect, with the determination coefficient (R^2) and root mean square error (RMSE) of 0.9989 and 6.9, respectively, which were significantly better than those of the other two models. Results of validation experiments showed that the detection system met the requirements of moisture content online detection in the drying process of agricultural products. The R^2 and RMSE were 0.9901 and 1.47, respectively, indicating the good performance of the model combining multi-sensor fusion and CNN in moisture content online detection for agricultural products in the drying process. The moisture content online detection system established in this study is of great significance for researching new drying processes and realizing the intelligent development of drying equipment. It also provides a reference for online detection of other indexes in the drying process of agricultural products.

KEYWORDS

convolutional neural network, prediction model, multi-sensor fusion, moisture content, online detection

1 Introduction

As an essential parameter in the drying processing of agricultural products, moisture content characterizes the drying rate and signals the end of drying (Yu et al., 2023). Achieving online detection of moisture content in the drying process is essential to optimize the drying process and realize the automation of drying. At present, material moisture content online detection methods include the dielectric properties method (Celik et al., 2022), model prediction method (Dalvi-Isfahan, 2020), spectral imaging method (Cho et al., 2020), and weighing method (Pongsuttiyakorn et al., 2019). The dielectric property method is a moisture content detection method based on the correlation between the dielectric properties of the material and the moisture content. The dielectric properties of the material are greatly affected by temperature, and performing accurate moisture content detection when the material is dried at different temperatures is not easy. The model prediction method is suitable for moisture content detection of specific materials under a specific drying environment. When the material or drying environment changes, the model needs to be re-established to detect moisture content. The spectral imaging method is expensive and requires computer vision technology, which is complicated to operate, not applicable to the agricultural product drying industry with low added value. The weighing method can detect the moisture content of different materials with high versatility, low cost, and simple operation, and is an essential method of moisture content online detection.

The weighing method is a method for real-time acquisition of the weight of the material during the drying process, according to the principle of constant dry matter, combined with the initial moisture content of the material to achieve moisture content online detection. The key to the weighing method is accurately acquiring the material weight by using the load sensor. The complex drying environment, the vibration of equipment, the impact and disturbance of airflow, and the variation of drying temperature will bring severe errors to the detection of the load sensor, which will affect the accuracy of the moisture content detection. Ju et al. (2023) stopped the blower to avoid airflow's influence on the load sensor's detection during moisture content detection but ignored the error caused by temperature variation. Yang et al. (2023a) similarly achieved moisture content detection by using the stop-air detection strategy and corrected the detection error caused by temperature change, improving moisture content detection accuracy. However, in different drying programs, the temperature variation range is far beyond the linear calibration interval of the load sensor, and achieving accurate measurement by simply compensating the error due to temperature change is difficult. Wang et al. (2014) while using a stop-air detection strategy at the same time, carried out linearization calibration of the detection results of the load sensor at different temperature sections and load ranges. The scheme effectively avoids the influence of temperature on the detection of the load sensor. Reyer et al. (2022) directly installed the load sensor in the drying chamber outside, more effectively eliminating the measurement error caused by temperature. However, this scheme destroyed the sealing of the drying chamber, which increased the difficulty of controlling the temperature and humidity in the drying chamber. The above moisture content online detection scheme was implemented under the stop-air detection strategy.

With the development of automation and intelligence in the drying industry, drying equipment needs to make real-time adjustments to the temperature and humidity in the drying chamber according to the drying rate, and it needs to detect the moisture content more frequently. In this context, stopping the blower to detect moisture content will undoubtedly break the continuity of drying and further increase energy consumption and drying time. Therefore, the existing moisture content online detection technology cannot meet the needs of the current drying process.

Multi-sensor fusion technology is an information processing method that uses computer technology to automatically analyze and synthesize information and data from multiple sensors or sources under specific guidelines to obtain the required decisions and estimates (Xie et al., 2022). Factors affecting load sensor detection, such as vibration of equipment, impact and disturbance of airflow, and temperature variation, can be detected by the sensors. Multi-sensor fusion technology can fuse the load sensor signal with other sensor signals, make regression prediction of the real weight of the material in the drying process, and further detect the moisture content of the material. Regression prediction based on multi-sensor fusion technology has been widely used in other industries, such as the remaining life prediction of aviation engines (Li et al., 2022b), tool wear prediction (Meng et al., 2021), air pollution level prediction (Ari and Alagoz, 2022), and wheel odometry prediction (Zhu et al., 2021). Kirsanov et al. (2021) used PLSR to relate the sensor signals to the values of different water quality parameters, which enabled the accurate detection of various water quality parameters. Li et al. (2019) has applied SVM in multi-sensor fusion to assess green tea quality accurately.

The complex dry environment causes all kinds of sensor signals to fluctuate and behave randomly. Raw sensor signals are difficult to transform into a stable output value after filtering. At the same time, the filtering process removes essential information hidden in the raw signals that are correlated with the output. Deep learning has been introduced into multi-sensor fusion prediction to obtain the correlation and causality hidden in raw monitoring data (Xu et al., 2020). Deep learning is a specific machine learning type consisting of a stack of multilayer nonlinear processing units (Samaras et al., 2019). Deep learning techniques have more powerful representational learning capabilities than traditional machine learning techniques. They can learn complex functions that map inputs to outputs directly from raw data (Wang et al., 2021). Convolutional neural network (CNN), a class of feed-forward neural networks that include convolutional computation and have a deep structure, are one of the representative algorithms for deep learning (Tong et al., 2023). CNN have also been widely used in solving regression prediction problems with multi-sensor fusion and have contributed to many tasks with state-of-the-art accuracy (Arvidsson et al., 2021; Zeng et al., 2021; Wan et al., 2022; Li et al., 2022a; Gao et al., 2023).

Given the air-impingement dryer's fast drying speed and high heat transfer coefficient, this study built a moisture content online detection system in the air-impingement dryer (Yang et al., 2023c). The tray

position needs to be added to the prediction model as an input variable because of the particularity of the structure of the air-impingement dryer. Overall, this study applied multi-sensor fusion technology to the moisture content online detection process and used the CNN prediction model to fuse the raw signals from the weight sensor, air velocity sensor, temperature sensor, and the tray position to accurately obtain the real weight of the material in the drying process. According to the initial moisture content of the material, the current moisture content was obtained, and finally, the regression prediction model of moisture content was established. A moisture content online detection system was built based on this model.

In summary, this study (1) completed the construction of a multi-sensor data acquisition platform; (2) carried out cantaloupe slice drying experiments to obtain the raw monitoring signals of multi-sensors used for CNN training; (3) established a material weight prediction model based on CNN and compared it with the traditional prediction model; and (4) established a moisture content online detection system based on the CNN prediction model. The technology roadmap is shown in Figure 1. This study built a moisture content online detection system and will provide new technical support for drying process optimization and promote the intelligent development of drying equipment.

2 Principles and methods

2.1 Principles

In this study, the online detection system was built in the air-impingement dryer and realized the online detection of the material moisture content based on the weighing method. The following sections show the operating principle of air-impingement dryer and the principle of moisture content detection based on the weighing method.

2.1.1 Operation principle of air-impingement dryer

The air-impingement dryer is a technology that realizes drying by impinging and heating the material with pressurized hot air (Zheng et al., 2023). Figure 2 shows the operation principle diagram of the air-impingement dryer. The air-impingement dryer is divided into the inner chamber and the outer chamber. Six infrared heating tubes are evenly installed on the top of the inner chamber, with a total power of 0–2 KW. The infrared heating tubes heat the materials placed on the tray with infrared radiation. The fan draws air from the inner chamber into the outer chamber. The air is cooled in the outer chamber, and the wet air is discharged from the outer chamber through a wet discharge valve. The fan blows the air into the inner chamber through the nozzle to realize internal circulation of the air in the equipment. When the air through the nozzle is squeezed, it forms a high-pressure airflow and impacts the material, removing the moisture on its surface. The material is dried under the double effect of infrared radiation heating and airflow impact.

The dryer regulates the air velocity of the fan through a frequency converter. The dryer is not equipped with an air velocity sensor, which cannot achieve closed-loop regulation of the air velocity, so there are large fluctuations in the airflow in the inner chamber. A temperature sensor is installed at the nozzle, which is used to detect the temperature of the air in the inner chamber. The equipment achieves closed-loop control of the air temperature in the inner chamber by adjusting the power of the infrared heating tube. The temperature of the outer chamber is significantly lower than that of the inner chamber due to the lack of heating by the infrared heater. The internal circulation of air increases the difficulty of temperature control in the inner chamber.

2.1.2 Principle of moisture content detection based on the weighing method

Moisture content detection based on the weighing method is a method to calculate the moisture content based on the initial weight and the real-time weight during the drying process under the default condition that the initial moisture content of the same batch of material is the same. The formula for calculating the moisture content based on the weighing method (wet basis) is shown in Equation 1 (Liu et al., 2021):

$$w_t = \frac{m_t - m(1 - w_i)}{m_t} \times 100\,\% \tag{1}$$

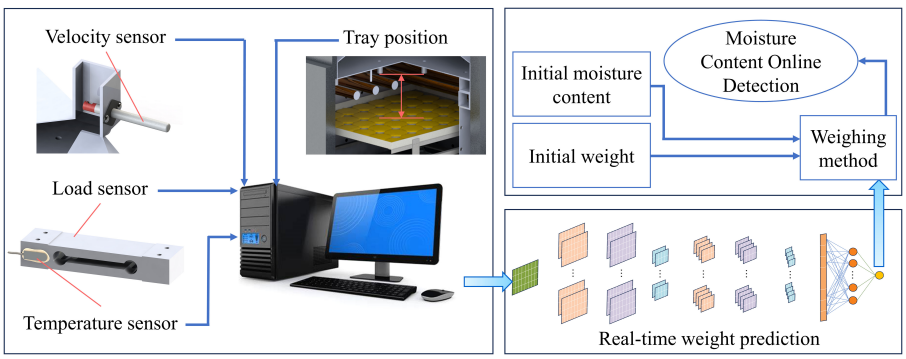


FIGURE 1
Technology Roadmap.

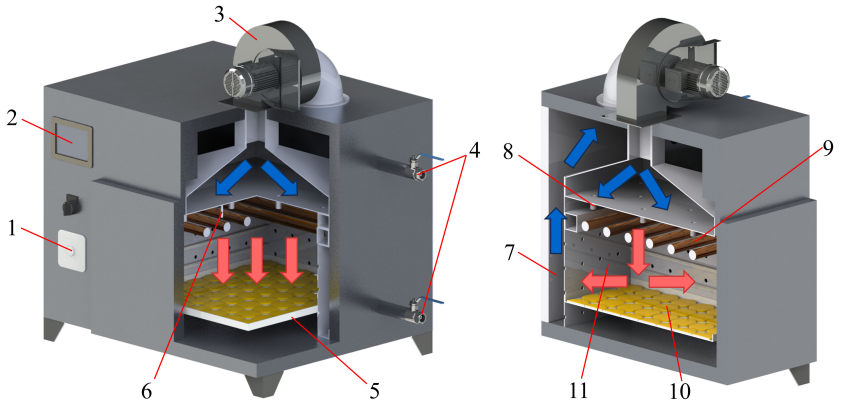


FIGURE 2
Operation principle diagram of air-impingement dryer. (1) Air velocity adjustment knob; (2) temperature control touch panel; (3) fan; (4) wet discharge valve; (5) tray; (6) temperature sensor; (7) outer chamber; (8) air nozzle; (9) infrared heating tube; (10) material; (11) inner chamber.

where w_t is the moisture content (wet base) of the material at time t, %; m_t is the weight of the material at time t, g; m is the initial weight of the material, g; and w_i is the initial moisture content (wet basis) of the material, %.

A load sensor usually needs to be installed at the bottom of the rack to obtain the weight of the material at time t. During the actual drying process, the load sensor has difficulty outputting a stable weight signal due to airflow disturbances and equipment vibration. The impact of airflow and temperature variation also causes measurement errors in the load sensor. During the air-impingement drying process, people often change the tray position on the rack to obtain different drying quality and drying rates of the material (Chang et al., 2022). Preliminary experiments found that the tray position also significantly affects the load sensor's measurement results. The tray position here indicates the distance between the tray and the nozzle.

The drying temperature, air velocity, tray position, and material weight set by the drying process of different materials vary greatly. Therefore, the error caused by the complex dry environment to the detection value of the load sensor needs to be eliminated. In addition to the tray position, other influencing factors can be detected by the sensor. The air velocity sensor can detect the airflow speed, and its raw signal can also reflect the airflow fluctuation. The temperature sensor can detect the temperature value that affects the measurement value of the load sensor. The device's vibration will also be reflected in the raw signal of the load sensor.

2.2 Multi-sensor data acquisition

The monitoring data from the three sensors during the drying process need to be collected for model training to establish a moisture content online detection model with the raw signals from load sensor, air velocity sensor, temperature sensor, and the tray position as inputs and the real weight of the material as outputs.

2.2.1 Multi-sensor data acquisition platform construction

The data acquisition system consists of an upper computer, a weight acquisition module, an air velocity acquisition module, a temperature acquisition module, and a 485 communication module as shown in Figure 3. The upper computer adopted the Legion Y7000P computer from Lenovo, which was responsible for human–computer interaction and data storage. The upper computer adopted the MODBUS communication protocol and connected with each slave unit via three RS485 buses to form a data acquisition network. In the weight acquisition module, the cantilever beam pressure sensor (HYPX017, Hengyuan Sensor Technology Co., Ltd., Bengbu, China) with a range of 3 kg was selected to collect the weight signal of the material in the drying process. In the air velocity acquisition module, a thermal air velocity sensor (WM4200, Chaozhi Reed Technology Co., Ltd., Changchun, China) with a range of 20 m/s was used to acquire the air velocity. The air velocity sensor was installed in the air duct of the outer chamber with a lower temperature to increase the service life of the air velocity sensor and to reduce the influence of temperature on the measurement results of the air velocity sensor. The dimensions of the air duct were 60mm × 50mm. The dimensions of the tray were 400mm × 350mm. Temperature variations in the elastic substrate of the load sensor are the leading cause of measurement errors. In the temperature acquisition module, a temperature sensor (PT100, Songdao Heating Sensor Co., Ltd., Shanghai, China) with a range of −45°C to 125°C was selected to collect the temperature signal of the load sensor elastic substrate. The temperature sensor was fixed to the elastic substrate by using thermally conductive silicone. A 485 communication module was used to communicate between the three sensors and the upper computer. The signals of each sensor were not filtered to obtain the correlation hidden in the raw monitoring data of the sensors.

2.2.2 Design of single-factor experiment

A single-factor experiment was carried out to investigate the effect of air velocity, temperature, and tray position on the measured

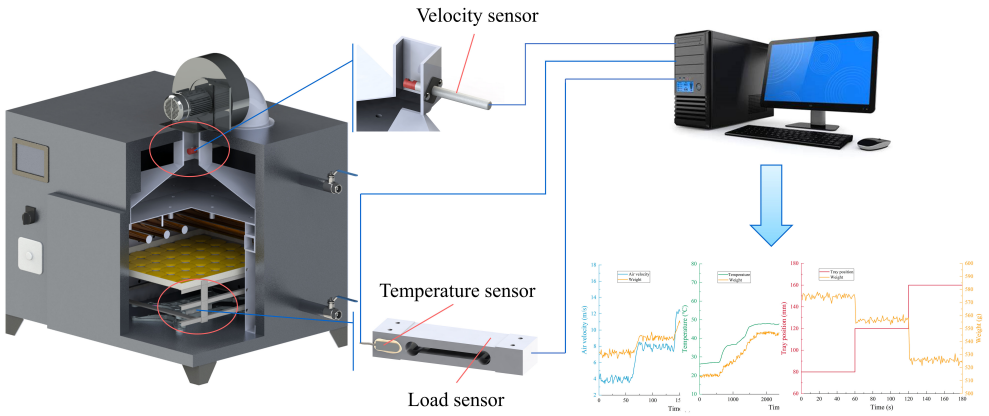


FIGURE 3
Multi-sensor data acquisition system.

data from the load sensor. First, air velocity was used as a single-factor variable for the experimental design. The load sensor measurement data were obtained continuously under a constant load of 500 g with a sampling interval of 1 s and duration of 300 s, load sensor substrate temperature of 25°C, tray position of 80 mm, and air velocity varying in the range of 4–16 m/s. The load sensor substrate temperature was set to a zero point temperature of 25°C. The zero point temperature of the load sensor refers to the temperature at which the output voltage of the load sensor is zero at no load, and the weighing value at this temperature is the standard value. In the experiment with load sensor substrate temperature as the single factor variable, the fan stopped running, the constant load was 500 g, the tray position was 80 mm, the temperature varied in the range of 25°C–70°C, the sampling interval was 10 s, and the sampling duration was 80 min. In the experiment with tray position as the single factor variable, the constant load was 500 g, the load sensor substrate temperature was 25°C, and the air velocity was set at 16 m/s. The load sensor data were collected at 80, 120, and 160 mm tray positions with a sampling interval of 1 s and a sampling duration of 180 s.

2.2.3 Experimental design for multi-sensor data acquisition

The data acquisition experiments were carried out under different drying environments to thoroughly investigate the correlation between the input variables and the weight of the material and improve the prediction model's accuracy. The temperature setting range in the drying of agricultural products is usually 40°C–70°C. The maximum air velocity of the outer air duct in the air-impingement dryer is 16 m/s. The tray position is determined by the structure of the rack, which has three layers in total, and the distances between the tray and the nozzle are 80, 120, and 160 mm, respectively. In summary, the data acquisition experiments were carried out at different temperatures (40°C, 50°C, 60°C, and 70°C), different air velocities (4, 8, 12, and 16 m/s), and different tray positions (80, 120, and 160 mm). Each group drying experiment randomly obtained 10 groups of data, and each group of data sampling interval was greater than 5 minutes, thus obtaining a total of 480 groups of data ($4 \times 4 \times 3 \times 10$). Each sampling time lasted 8 s, and the sampling frequency was 8 Hz.

Data acquisition experiments were conducted during the cantaloupe slice drying experiment. Fresh, undamaged cantaloupe was peeled, deseeded, and sliced into $30 \times 50 \times 7$ mm slices. For each set of experiments, 1000 g of cantaloupe slices were weighed and placed on the tray. The cantaloupe slices were removed from the tray, weighed immediately after each data acquisition, and quickly returned to the tray. The weight of the cantaloupe slices was the output value of this dataset.

2.3 Prediction model of moisture content online detection

2.3.1 Convolutional neural network

CNN is a deep learning model or a multilayer perceptron similar to artificial neural network. In this study, the CNN was used for regression analysis to mine potential information in the raw monitoring data of load sensor, air velocity sensor, temperature sensor, and tray position to achieve weight prediction and complete the study of moisture content online detection.

The CNN used in this study consisted of the input layer, convolutional layer, batch normalization layer, average pooling layer, fully connected layer, and output layer, and its structure is shown in Figure 4. The function of the input layer was mainly to normalize the input data, which can improve the model's generalization ability and increase the training speed (Hu et al., 2023). The convolutional layer uses convolutional operations to filter out redundant information in the original data, enhance the information related to the output, and achieve automatic feature extraction (Wang et al., 2022). The convolution kernel size was set to 3×3, the convolution mode was set to "same," and the step size was set to 1. The number of convolution kernels needed to be adapted to the structure of the training data, which was determined by a trial-and-error method based on the performance evaluation index of the model (Ma et al., 2023). The activation function in the neural network structure can make a nonlinear mapping of the

output, which is particularly important for the accuracy of the prediction model (Guan et al., 2022).

The CNN model was trained using the rectified linear unit (ReLU) function, the hyperbolic tangent (tanh) function, and the sigmoid function to select the best activation function. The most appropriate activation function was selected based on the model performance evaluation index. The average pooling layer was located after the convolutional layer, and its function was to accomplish the parameter degradation and maintain translation invariant properties, which can be achieved to reduce the feature map while preserving the critical features in the input to some extent (Zhong et al., 2023). The size of the average pooling matrix was set to 2, and the step size was set to 2. Adding a batch normalization layer between the convolutional layer and the average pooling layer allowed the inputs of each neural network layer to maintain the same distribution during neural network training, thus reducing the internal covariate shift, improving the gradient mobility, and achieving the regularization effect (Tan et al., 2021). A dropout layer was set before the fully connected layer. In the dropout layer, some input elements were randomly changed to zero with a probability set to 0.05. The dropout layer randomly rendered 5% of the elements non-functional, thus avoiding overfitting (Zhao et al., 2023). The fully connected layer flattened the feature map into a one-dimensional vector for final feature integration and output prediction. The role of the output layer was to output the predicted result, which in this study was the real weight of the material.

A total of 480 sets of data were randomly sorted, and 70% of the data (336 sets) were used as the training set, 15% of the data (72 sets) were used as the validation set, and 15% of the data (72 sets) were used as the test set. During the training of the CNN, the network parameters were updated according to the loss function for each training batch, and the batch size was set to 16. The training set had 336 sets of data, and one iteration was completed for every 21 updates of the network parameters. The maximum number of iterations was set to 50. The root mean square error (RMSE) between the real values and the predicted values of the validation set was used as the loss function, which was calculated by Equation 2. Model training was performed in a Legion Y7000P computer from Lenovo with MATLAB R2021a software.

$$\text{Loss} = \text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{2}$$

Where n is the number of samples in the validation set, $\hat{y}_i$ and y_i are the predicted value and real value of the i^{th} sample.

2.3.2 Evaluation of model performance

The performance of the prediction model was evaluated in terms of the RMSE of the training set (RMSE_{Tr}), validation set (RMSE_{Ve}), and test set (RMSE_{Te}), and the coefficients of determination (R^2) of the training set (R^2_{Tr}), validation set (R^2_{Ve}), and test set (R^2_{Te}). RMSE and R^2 represent the deviation and degree of fitting between the real and predicted values, respectively. RMSE focuses on the magnitude of the error, with smaller values indicating greater accuracy of the model. R^2 focuses on the ability of the model to explain the variation in the data, with values closer to 1 indicating a better fit of the model. These evaluation parameters were calculated by Equation 3 and Equation 4 (Wang et al., 2023):

$$R^2_{Tr}, R^2_{Ve}, R^2_{Te} = 1 - \frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{\sum_{i=1}^{n}(y_i - y_m)^2} \tag{3}$$

$$\text{RMSE}_{\text{Tr}}, \text{RMSE}_{\text{Ve}}, \text{RMSE}_{\text{Te}} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{4}$$

Where n is the number of samples in the corresponding set (training set, validation set, and test set), $\hat{y}_i$ and y_i are the predicted value and real value of the i^{th} sample, and y_m is the mean value of all the samples.

2.4 System validation experiments

MATLAB software was used for data processing, model prediction, and real-time display of moisture content in the moisture content online detection system. First, the initial weight and the initial moisture content of the material were set, and the initial moisture content was measured by the oven method (Yang et al., 2023b). The sensor cannot detect the tray position and is a fixed value. Thus, this value also needs to be input into the software.

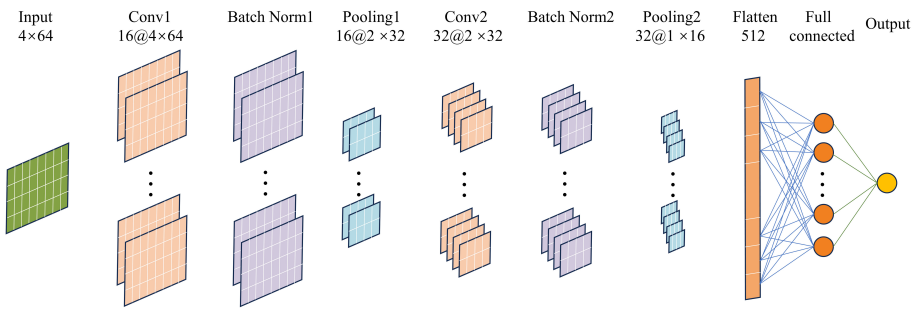


FIGURE 4
Structure of CNN.

MATLAB processed the data obtained from the sensors to meet the format requirements of the predictive model inputs. The processed data were then fed into a trained prediction model, which outputted the real weight of the material. The moisture content of the material was calculated according to the initial weight and the initial moisture content and displayed in real time.

The cantaloupe slice drying experiment in Section 2.2.3 was repeated. The initial weight of cantaloupe slices was 1000 g, and the initial moisture content was 90.19% (wet basis). The set values of air velocity, temperature and tray position were randomized into five experimental groups. The experimental design is shown in Table 1. Where temperature refers to the air temperature in the inner chamber, measured by the temperature sensor in Figure 2. Five sets of experiments were conducted sequentially under the same set of material conditions, with each set lasting 30 minutes. Three sensors, including a load sensor, an air velocity sensor, and a temperature sensor acquired data once at a random time during each set of test cycles.

3 Results and discussion

3.1 Results and analyses of single-factor experiment

Figure 5A shows the experimental results with air velocity as a single factor variable. The monitoring signal of the load sensor fluctuated greatly with more noise, which was due to the unstable impact force of the airflow on the tray caused by the inhomogeneity of the airflow. The vibration generated by the equipment operation made the load sensor unable to acquire the data in a stable state. The measured values of the load sensor in different air velocities had a significant difference, and the variation range of the measured values was from 519.34 g to 579.78 g, with a variation of 60.44 g. The wind direction was perpendicular to the tray's upper surface; thus, the load sensor's measured values showed a positive relationship with the air velocity and the relationship had a strong transient nature.

Figure 5B shows the experimental results with temperature as a single factor variable. The fluctuation range of the load sensor measurement value was 500.03–506.15 g with a fluctuation amplitude of 6.12 g in the temperature variation range of 26.26°C–65.76°C. The monitoring value of the load sensor and the temperature of the load sensor elastic substrate at a fixed load showed a positive relationship. The load sensor used for moisture content detection was a resistance strain gauge pressure sensor. Temperature can cause errors and noise in the measured values of the load sensor by affecting the resistance strain gauges' resistance value and the elastic substrate's elastic modulus (Burnos and Rys, 2017). The effect of temperature on the measured value of weight sensors also had significant relationships with the load. Wang et al. (2014) calibrated the measured values of weight sensors at different temperature sections and load ranges, effectively avoiding the influence of temperature on the detection of the load sensor.

Figure 5C shows the experiment results with the tray position as a single factor variable. Under a constant load, the tray position significantly affected the load sensor's measurement value. The fluctuation range of the load sensor measurements at the three tray positions was 521.44–578.01 g, with a fluctuation range of 56.57 g. The smaller the distance between the tray and the nozzles, the more concentrated the airflow from the nozzles, and the greater the force exerted on the tray, which in turn increased the load sensor measurements. The three layers of the tray were arranged vertically so that the tray position did not affect the measured value of the load sensor when the fan was stopped. Therefore, the tray position's influence on the load sensor's measured value was very much related to the air velocity.

3.2 Results and analyses of CNN training

3.2.1 Selection of activation function and number of convolution kernels

The activation function and the number of convolutional kernels in the CNN needed to be determined by trial-and-error method based on the model performance index. The model performance test with different activation functions and number of convolution kernels was performed with the same training, validation, and test sets. The test results are shown in Table 2. Table 2 shows that the model performance of the ReLU activation function was significantly better than that of the tanh and sigmoid functions. Li et al. (2022c) had similar findings when applying CNNs to predictive modeling. At the same activation function number (ReLU), a slight difference was found in the model performance for different numbers of convolutional kernels. The best model performance (R^2 closest to 1 and minimum RMSE) for the training, validation, and test sets occurred in the fourth, fifth, and fourth groups, respectively. The test set did not participate in the training process of the CNN, and its model performance was more reliable. The results of the test set of the fifth group were significantly

TABLE 1 The design of system validation experiments.

Factor	Group				
	1	2	3	4	5
Air velocity (m/s)	12.6	6.3	8.4	14.1	9.7
Tray position (mm)	140	100	140	100	60
Temperature (°C)	68	43	50	56	63

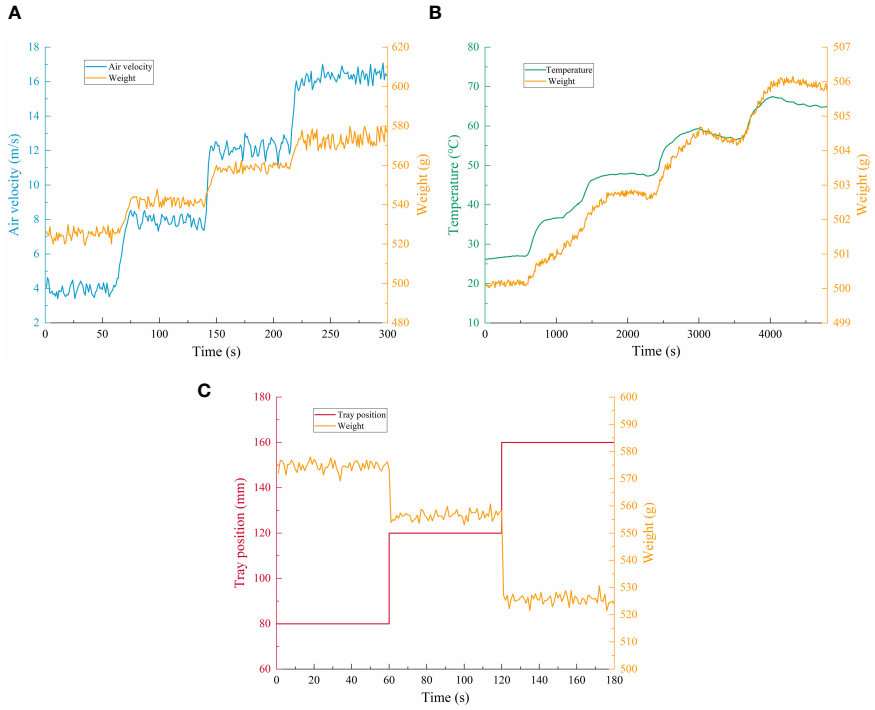


FIGURE 5
The results of single-factor experiment. **(A–C)** are the experimental results with air velocity, temperature and tray position as single factors, respectively.

worse than those of the fourth group. Taken together, the best structure of the CNN was the fourth group, the best activation function was ReLU, and the ideal number of convolutional kernels was (16, 32).

3.2.2 Variable learning rate optimization

After the optimal activation function and number of convolution kernels were determined, the RMSE of the model was still high. An observation of the loss function curve during the training process of the CNN showed that the loss function showed regular oscillations at the late stage of training, but no decreasing trend occurred. This is the phenomenon of gradient disappearance caused by a too-large learning rate in the late training period (Noppitak and Surinta, 2022). However, if the learning rate was reduced, then the training time would be much longer, and obtaining the global optimum would be difficult. This study set the learning rate schedule to piecewise mode, which can adopt different learning rates in different training stages. The initial learning rate was set to 0.0001, the learning rate drop period was set to 50, the learning rate drop factor was set to 0.25, and the maximum number of iterations was set to 200. The loss function curve with variable learning rate optimization was shown in Figure 6. The maximum number of iterations was 200, the learning rate decreased every 50 iterations, and the loss function was updated 21 times per iteration (number of training set samples/batch size). The loss function was updated a total of 4200 times. In Figure 6, the loss function oscillation amplitude decreased with each decrease in the learning rate, which was due to gradient reduction caused by the decrease in the iteration step size. Every time the learning rate decreased, the loss function decreased significantly, and the whole training process showed a decreasing trend, which indicates that the iterative gradient was restored and the training results were constantly approaching the optimal value. The R^2 and RMSE of the model test set were 0.9989 and 6.9, respectively, and the prediction results of the model test set are shown in Figure 7. Figures 7A, B show the fitting degree and prediction error of the predicted value to the real value, respectively. The maximum error was 0.042, and 85% of the test data had a prediction error of less than 0.02. In conclusion, the prediction model with variable learning rate optimization can more accurately predict the material weight.

3.3 Model performance comparison

The CNN model is more complex than the other two prediction models, and the combination with hardware is much more difficult. Therefore, the performance of the classical linear and nonlinear regression models PLSR and SSVM was tested to prevent model performance excess. Figures 8A–C show scatterplots of real and predicted values for the PLSR, SVR, and CNN models. The solid line is a regression line that aids in analyzing the degree of deviation of the predicted values relative to the real values. The closer the scatter is to the regression line, the better the fit of the model.

10.3389/fpls.2024.1289783

TABLE 2 Test results of different activation function and number of convolution kernels (shaded group is the optimal structure; bold font indicates the optimal solution).

Group	Activation function	Number of convolution kernels	Training set		Validation set		Test set	
			R^2	RMSE	R^2	RMSE	R^2	RMSE
1	Relu	8, 16	0.9931	17.3	0.9860	31.4	0.9734	24.7
2		8, 32	0.9910	19.8	0.9888	22.1	0.9823	25.6
3		8, 64	0.9911	19.7	0.9892	21.7	0.9829	25.1
4		16, 32	**0.9957**	**13.8**	0.9876	23.2	**0.9913**	**17.9**
5		16, 64	0.9948	15.1	**0.9908**	**20.0**	0.9800	27.2
6		32, 64	0.9926	18.0	0.9867	24.1	0.9769	29.2
7	Tanh	8, 16	0.9314	54.7	0.9096	57.8	0.8710	75.0
8		8, 32	0.8904	69.2	0.8770	73.3	0.8795	66.8
9		8, 64	0.8892	53.4	0.8836	58.2	0.8854	58.0
10		16, 32	0.9439	49.5	0.8566	79.1	0.8625	71.3
11		16, 64	0.9377	40.9	0.9299	41.5	0.8947	56.8
12		32, 64	0.9248	57.3	0.8945	62.5	0.8891	69.5
13	Sigmoid	8, 16	0.9391	51.5	0.9229	53.4	0.9014	65.6
14		8, 32	0.9806	29.1	0.9749	33.1	0.9718	32.3
15		8, 64	0.9791	30.2	0.9644	39.4	0.9512	42.5
16		16, 32	0.9388	51.7	0.9214	53.9	0.8984	66.6
17		16, 64	0.9676	37.6	0.9630	40.2	0.9575	39.6
18		32, 64	0.9858	24.9	0.9714	35.3	0.9688	34.0

Figure 8C shows that the sample points of the CNN model were centrally distributed near the regression line, and the model had the best fit. In contrast, the PLSR and SVR model sample points were more dispersed, not exactly distributed near the regression line, and shifted relative to the regression line. The RMSE of the test set for the two models were 47.9 and 39.8, respectively, which cannot meet the accuracy requirements of moisture content online detection.

3.4 Validation test results

The moisture content detection based on the weighing method defaulted to the same initial moisture content of the same batch of material. A deviation occurred between the moisture content of the material used for the oven test and the drying experiment, which will inevitably affect the accuracy of moisture content detection. The validation test results are shown in Table 3. The RMSE of the five validation experiments was 1.47, which indicated that the error caused by defaulting to the same initial moisture content of the same batch of materials was within the acceptable range. The results of the five validation experiments showed that the fit of the moisture content detection model based on the CNN established in this study was acceptable, and the moisture content online detection system can accurately detect the moisture content of materials in the drying process.

4 Conclusion

In this study, a multi-sensor data acquisition platform was set up, and a CNN prediction model was established with raw

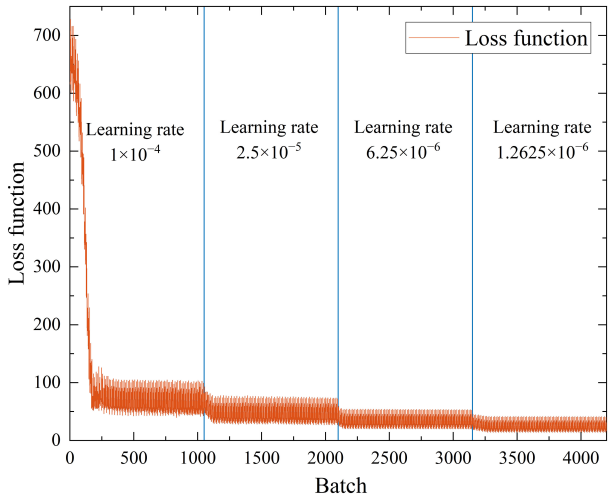


FIGURE 6
Loss function curve with variable learning rate optimization.

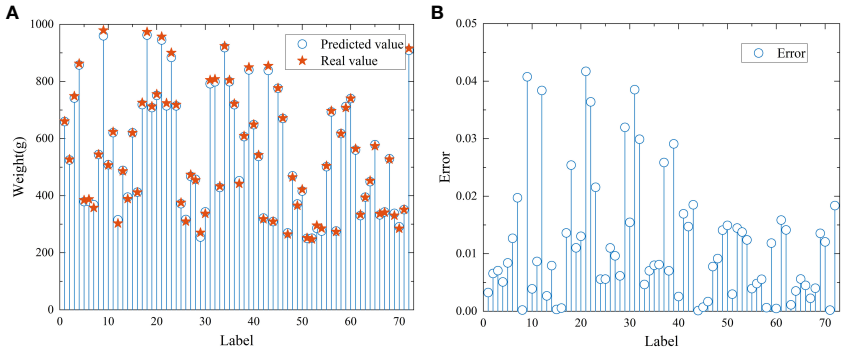


FIGURE 7
The prediction results of the CNN model test set. **(A)** and **(B)** are plots of fitting effects and prediction errors, in that order.

monitoring data from the load sensor, air velocity sensor, temperature sensor and the tray position as inputs and the real weight of materials as outputs. The optimal activation function and number of convolutional kernels for the prediction model were selected. The optimal activation function was ReLU, and the optimal number of convolutional kernels was (16, 32). The training process of the CNN was optimized with a variable learning rate to optimize the model performance further. The final performance of the CNN prediction model was satisfactory (with R^2 and RMSE of 0.9989 and 6.9, respectively) and was significantly better than that of the traditional linear PLSA model (with R^2 and RMSE of 0.9489 and 47.9, respectively) and the nonlinear SVR model (with R^2 and RMSE of 0.9648 and 39.8, respectively). A moisture content online detection system was constructed based on the CNN prediction model. Validation experiments were carried out, and the R^2 and RMSE of the validation experiments were 0.9901 and 1.47, respectively. The validation experiments showed that the CNN prediction model was fully applicable to moisture content online detection, and the detection system based on this model fully met the accuracy requirements of moisture content online detection.

In the moisture content online detection system proposed in this study, the detection of initial moisture content still has errors. Future research can use more advanced and convenient technology to detect

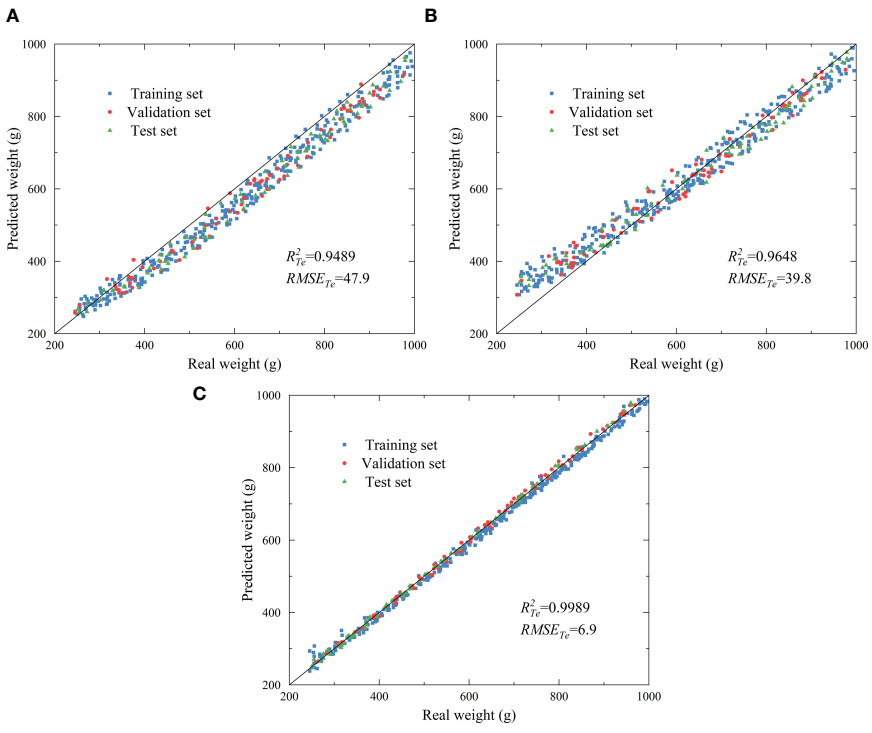


FIGURE 8
Scatterplot of PLSR **(A)**, SVM **(B)**, and CNN **(C)** models.

TABLE 3 The validation test results.

	Group					R^2	RMSE
	1	2	3	4	5		
Real moisture content	86.77%	81.38%	75.90%	64.46%	45.10%	0.9901	1.47
Predicted moisture content	86.58%	80.97%	74.99%	65.86%	47.90%		

the initial moisture content of materials quickly and effectively. In addition, this system was built with computer as the host computer. Scholars can compile the detection model into the microcontroller in future research, which is conducive to the application and promotion of the detection system in actual production.

Overall, this study established a moisture content online detection system based on multi-sensor fusion and CNN prediction model, realizing real-time moisture content detection during agricultural products' drying process. This study will provide technical support for further optimization of the drying process and will also promote the intelligent development of agricultural product drying equipment.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

TY: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. XZ: Conceptualization, Formal analysis, Investigation, Project administration, Supervision, Validation, Writing – review & editing. HX: Validation, Writing – review & editing. CS: Validation, Writing – review & editing. JZ: Validation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This study was supported by the Shihezi University Achievement Transformation and Technology Promotion Project (No. CGZH201808).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Ari, D., and Alagoz, B. B. (2022). An effective integrated genetic programming and neural network model for electronic nose calibration of air pollution monitoring application. *Neural. Comput. Appl.* 34, 12633–12652. doi: 10.1007/s00521-022-07129-0

Arvidsson, S., Gullstrand, M., Sirmacek, B., and Riveiro, M. (2021). Sensor fusion and convolutional neural networks for indoor occupancy prediction using multiple low-cost low-resolution heat sensor data. *Sensors* 21, 1036. doi: 10.3390/s21041036

Burnos, P., and Rys, D. (2017). The effect of flexible pavement mechanics on the accuracy of axle load sensors in vehicle weigh-in-motion systems. *Sensors* 17, 2053. doi: 10.3390/s17092053

Celik, E., Parlak, N., and Cay, Y. (2022). Development of an integrated corn dryer with an indirect moisture measuring system. *Sadhana-Academy Proc. Eng. Sci.* 47, 1. doi: 10.1007/s12046-021-01775-1

Chang, A., Zheng, X., Xiao, H., Yao, X., Liu, D., Li, X., et al. (2022). Short- and medium-wave infrared drying of cantaloupe (Cucumis melon L.) slices: drying kinetics and process parameter optimization. *Processes* 10, 114. doi: 10.3390/pr10010114

Cho, J.-S., Choi, J.-Y., and Moon, K.-D. (2020). Hyperspectral imaging technology for monitoring of moisture contents of dried persimmons during drying process. *Food Sci. Biotechnol.* 29, 1407–1412. doi: 10.1007/s10068-020-00791-x

Dalvi-Isfahan, M. (2020). A comparative study on the efficiency of two modeling approaches for predicting moisture content of apple slice during drying. *J. Food Process Eng.* 43, e13527. doi: 10.1111/jfpe.13527

Gao, H., Li, Y., Zhao, Y., and Song, Y. (2023). Dual channel feature attention-based approach for RUL prediction considering the spatiotemporal difference of multisensor data. *IEEE Sensors J.* 23, 8514–8525. doi: 10.1109/jsen.2023.3246595

Guan, Y., Meng, Z., Sun, D., Liu, J., and Fan, F. (2022). Rolling bearing fault diagnosis based on information fusion and parallel lightweight convolutional network. *J. Manufact. Syst.* 65, 811–821. doi: 10.1016/j.jmsy.2022.11.012

Hu, Y., Ma, B., Wang, H., Zhang, Y., Li, Y., and Yu, G. (2023). Detecting different pesticide residues on Hami melon surface using hyperspectral imaging combined with 1D-CNN and information fusion. *Front. Plant Sci.* 14, 1105601. doi: 10.3389/fpls.2023.1105601

Ju, H.-Y., Vidyarthi, S. K., Karim, M. A., Yu, X.-L., Zhang, W.-P., and Xiao, H.-W. (2023). Drying quality and energy consumption efficient improvements in hot air drying of papaya slices by step-down relative humidity based on heat and mass transfer characteristics and 3D simulation. *Dry. Technol.* 41, 460–476. doi: 10.1080/07373937.2022.2099416

Kirsanov, D., Mukherjee, S., Pal, S., Ghosh, K., Bhattacharyya, N., Bandyopadhyay, R., et al. (2021). A pencil-drawn electronic tongue for environmental applications. *Sensors* 21, 4471. doi: 10.3390/s21134471

Li, J., Zhou, Q., Cao, L., Wang, Y., and Hu, J. (2022a). A convolutional neural network-based multi-sensor fusion approach for *in-situ* quality monitoring of selective laser melting. *J. Manufact. Syst.* 64, 429–442. doi: 10.1016/j.jmsy.2022.07.007

Li, L., Xie, S., Ning, J., Chen, Q., and Zhang, Z. (2019). Evaluating green tea quality based on multisensor data fusion combining hyperspectral imaging and olfactory visualization systems. *J. Sci. Food Agric.* 99, 1787–1794. doi: 10.1002/jsfa.9371

Li, X., Jiang, H., Liu, Y., Wang, T., and Li, Z. (2022b). An integrated deep multiscale feature fusion network for aeroengine remaining useful life prediction with multisensor data. *Knowledge-Based Syst.* 235, 107652. doi: 10.1016/j.knosys.2021.107652

Li, Y., Ma, B., Li, C., and Yu, G. (2022c). Accurate prediction of soluble solid content in dried Hami jujube using SWIR hyperspectral imaging with comparative analysis of models. *Comput. Electron. Agric.* 193, 106655. doi: 10.1016/j.compag.2021.106655

Liu, D., Zheng, X., Xiao, H., Yao, X., Shan, C., Chang, A., et al. (2021). Optimization of sequential freeze-infrared drying process. *Trans. Chin. Soc. Agric. Eng.* 37, 293–302. doi: 10.11975/j.issn.1002-6819.2021.17.034

Ma, T., Wang, A., Dalca, A., and Sabuncu, M. (2023). Hyper-convolutions *via* implicit kernels for medical image analysis. *Med. Image Anal.* 86, 102796. doi: 10.1016/j.media.2023.102796

Meng, X., Zhang, J., Xiao, G., Chen, Z., Yi, M., and Xu, C. (2021). Tool wear prediction in milling based on a GSA-BP model with a multisensor fusion method. *Int. J. Adv. Manufact. Technol.* 114, 3793–3802. doi: 10.1007/s00170-021-07152-w

Noppitak, S., and Surinta, O. (2022). dropCyclic: snapshot ensemble convolutional neural network based on a new learning rate schedule for land use classification. *IEEE Access* 10, 60725–60737. doi: 10.1109/access.2022.3180844

Pongsuttiyakorn, T., Sooraksa, P., and Pornchalermpong, P. (2019). Simple effective and robust weight sensor for measuring moisture content in food drying process. *Sensors Mater.* 31, 2393–2404. doi: 10.18494/sam.2019.2347

Reyer, S., Awiszus, S., and Muller, J. (2022). High-precision laboratory dryer for characterization of the drying behavior of agricultural and food products. *Machines* 10, 372. doi: 10.3390/machines10050372

Samaras, S., Diamantidou, E., Ataloglou, D., Sakellariou, N., Vafeiadis, A., Magoulianitis, V., et al. (2019). Deep learning on multi sensor data for counter UAV applications-A systematic review. *Sensors* 19, 4837. doi: 10.3390/s19224837

Tan, C., Li, F., Lv, S., Yang, Y., and Dong, F. (2021). Gas-liquid two-phase stratified flow interface reconstruction with sparse batch normalization convolutional neural network. *IEEE Sensors J.* 21, 17076–17084. doi: 10.1109/jsen.2021.3081432

Tong, J., Liu, C., Zheng, J., and Pan, H. (2023). Multi-sensor information fusion and coordinate attention-based fault diagnosis method and its interpretability research. *Eng. Appl. Artif. Intell.* 124, 106614. doi: 10.1016/j.engappai.2023.106614

Wan, S., Li, X., Zhang, Y., Liu, S., Hong, J., and Wang, D. (2022). Bearing remaining useful life prediction with convolutional long short-term memory fusion networks. *Reliabil. Eng. Sys. Saf.* 224, 108528. doi: 10.1016/j.ress.2022.108528

Wang, B., Lei, Y., Li, N., and Wang, W. (2021). Multiscale convolutional attention network for predicting remaining useful life of machinery. *IEEE Trans. Ind. Electron.* 68, 7496–7504. doi: 10.1109/tie.2020.3003649

Wang, D., Li, Y., Song, Y., Jia, L., and Wen, T. (2022). Bearing fault diagnosis method based on complementary feature extraction and fusion of multisensor data. *IEEE Trans. Instrument. Measure.* 71, 3527610. doi: 10.1109/tim.2022.3212542

Wang, D., Lin, H., Xiao, H., Liu, Y., Ju, H., Dai, J., et al. (2014). Design of online monitoring system for material moisture content in air-impingement drying process. *Trans. Chin. Soc. Agric. Eng.* 30, 316–324. doi: 10.3969/j.issn.1002-6819.2014.19.038

Wang, D., Zhao, F., Wang, R., Guo, J., Zhang, C., Liu, H., et al. (2023). A Lightweight convolutional neural network for nicotine prediction in tobacco by near-infrared spectroscopy. *Front. Plant Sci.* 14, 1138693. doi: 10.3389/fpls.2023.1138693

Xie, T., Huang, X., and Choi, S.-K. (2022). Intelligent mechanical fault diagnosis using multisensor fusion and convolution neural network. *IEEE Trans. Ind. Inf.* 18, 3213–3223. doi: 10.1109/tii.2021.3102017

Xu, X., Tao, Z., Ming, W., An, Q., and Chen, M. (2020). Intelligent monitoring and diagnostics using a novel integrated model based on deep learning and multi-sensor feature fusion. *Measurement* 165, 108086. doi: 10.1016/j.measurement.2020.108086

Yang, M., Liu, N., Wu, Y., Yang, S., Yang, L., Pu, Y., et al. (2023a). Development and experiments of an online moisture content measuring device in thin layer hot-air drying process. *Trans. Chin. Soc. Agric. Eng.* 38, 47–56. doi: 10.11975/j.issn.1002-6819.202211176

Yang, T., Zheng, X., Vidyarthi, S. K. K., Xiao, H., Yao, X., Li, Y., et al. (2023b). Artificial neural network modeling and genetic algorithm multiobjective optimization of process of drying-Assisted walnut breaking. *Foods* 12, 1897. doi: 10.3390/foods12091897

Yang, T., Zheng, X., Xiao, H., Shan, C., Ya1o, X., Li, Y., et al. (2023c). Drying temperature precision control system based on improved neural network PID controller and variable-temperature drying experiment of cantaloupe slices. *Plants-Basel* 12, 2257. doi: 10.3390/plants12122257

Yu, H., Hu, Y., Qi, L., Zhang, K., Jiang, J., Li, H., et al. (2023). Hyperspectral detection of moisture content in rice straw nutrient bowl trays based on PSO-SVR. *Sustainability* 15, 8703. doi: 10.3390/su15118703

Zeng, Y., Liu, R., and Liu, X. (2021). A novel approach to tool condition monitoring based on multi-sensor data fusion imaging and an attention mechanism. *Measure. Sci. Technol.* 32, 055601. doi: 10.1088/1361-6501/abea3f

Zhao, J., Ye, X., Yue, T., and Li, Y. (2023). CLDM: convolutional layer dropout module. *Mach. Vision And Appl.* 34, 63. doi: 10.1007/s00138-023-01411-4

Zheng, Z., Wang, S., Zhang, C., Wu, M., Cui, D., Fu, X., et al. (2023). Hot air impingement drying enhanced drying characteristics and quality attributes of ophiopogonis radix. *Foods* 12, 1441. doi: 10.3390/foods12071441

Zhong, Y., Teng, Z., and Tong, M. (2023). LightMixer: A novel lightweight convolutional neural network for tomato disease detection. *Front. Plant Sci.* 14, 1166296. doi: 10.3389/fpls.2023.1166296

Zhu, J., Tang, Y., Shao, X., and Xie, Y. (2021). Multisensor fusion using fuzzy inference system for a visual-IMU-wheel odometry. *IEEE Trans. Instrument. Measure.* 70, 2505216. doi: 10.1109/tim.2021.3051999

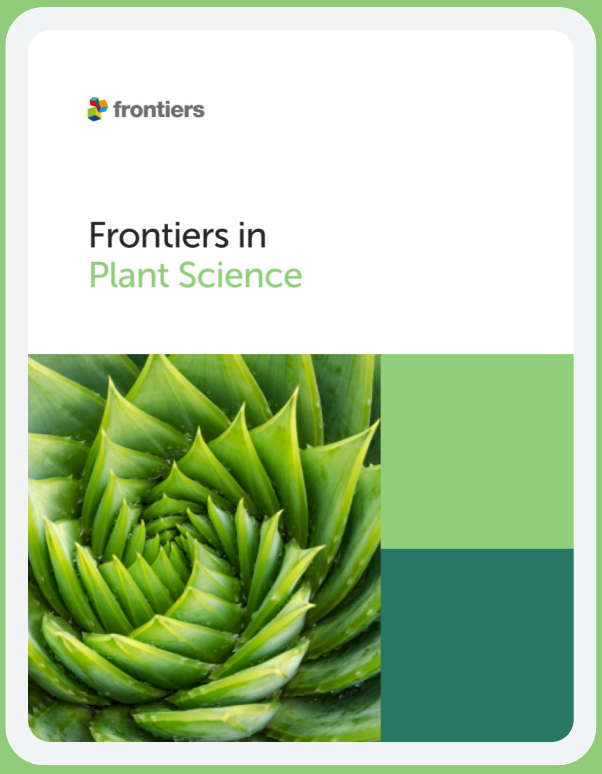
Frontiers in
Plant Science

Cultivates the science of plant biology and its applications

The most cited plant science journal, which advances our understanding of plant biology for sustainable food security, functional ecosystems and human health.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact