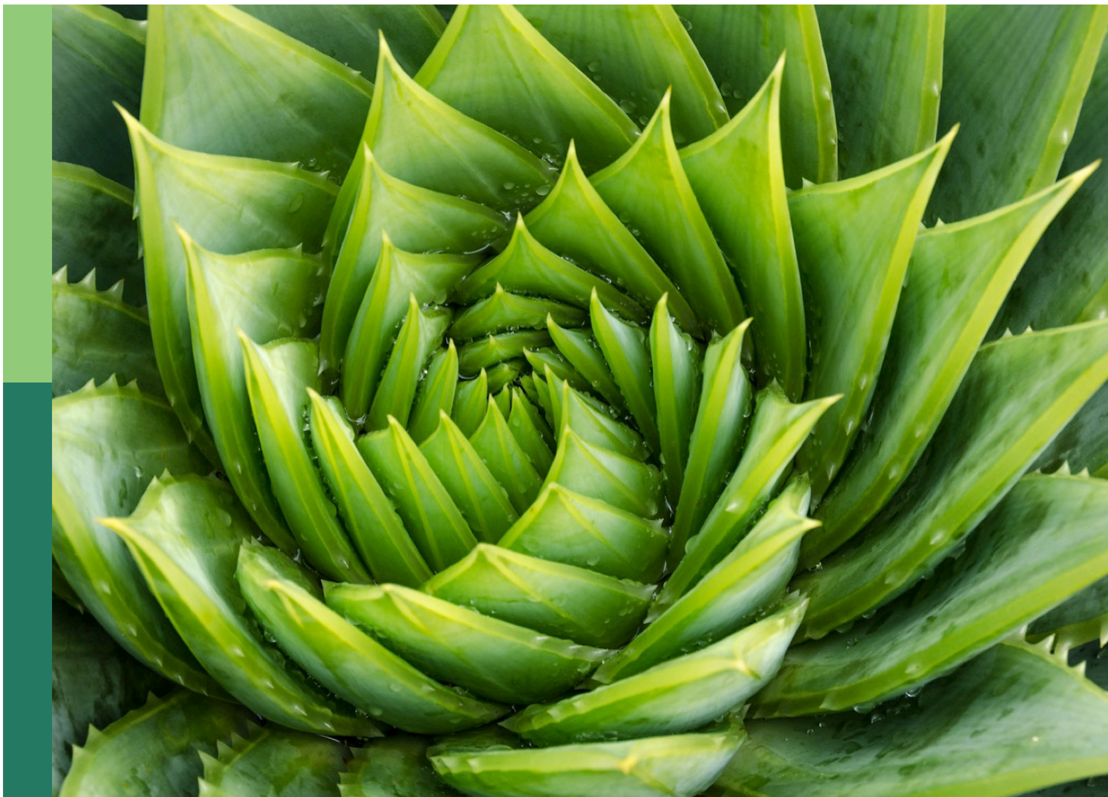
frontiers | Research Topics

Deep learning in crop diseases and insect pests

Edited by
Rujing Wang, Peng Chen and Po Yang

Published in
Frontiers in Plant Science



ISSN 1664-8714
ISBN 978-2-83251-774-1
DOI 10.3389/978-2-83251-774-1

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

Deep learning in crop diseases and insect pests

Topic editors

Rujing Wang — Hefei Institute of Technology Innovation, Hefei Institutes of Physical Science, Chinese Academy of Sciences (CAS), China
Peng Chen — Anhui University, China
Po Yang — The University of Sheffield, United Kingdom

Citation

Wang, R., Chen, P., Yang, P., eds. (2023). *Deep learning in crop diseases and insect pests*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-83251-774-1

Table of contents

frontiers | Frontiers in Plant Science

TYPE Editorial
PUBLISHED 13 February 2023
DOI 10.3389/fpls.2023.1145458

Editorial: Deep learning in crop diseases and insect pests

Peng Chen[1]*, Rujing Wang[2] and Po Yang[3]

[1]National Engineering Research Center for Agro-Ecological Big Data Analysis & Application, Information Materials and Intelligent Sensing Laboratory of Anhui Province, Institutes of Physical Science & Information Technology and School of Internet, Anhui University, Hefei, Anhui, China, [2]Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui, China, [3]Department of Computer Science, Sheffield University, Sheffield, United Kingdom

KEYWORDS

deep learning, BigData analysis, crop diseases and insect pests, feature representation, plant sensor

Check for updates

OPEN ACCESS

EDITED AND REVIEWED BY
Jucheng Yang,
Tianjin University of Science and Technology, China

*CORRESPONDENCE
Peng Chen
✉ pengchen@ustc.edu

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

RECEIVED 16 January 2023
ACCEPTED 27 January 2023
PUBLISHED 13 February 2023

CITATION
Chen P, Wang R and Yang P (2023) Editorial: Deep learning in crop diseases and insect pests. *Front. Plant Sci.* 14:1145458. doi: 10.3389/fpls.2023.1145458

Editorial on the Research Topic
Deep learning in crop diseases and insect pests

Many deep learning methods have been developed and successfully applied in the field of crop pests and diseases detection. In crop pest identification, deep learning methods can achieve good feature representation from large datasets based on various linear and nonlinear deep learning transformations and then discover the relationship in complex data based on specific supervised and unsupervised learning. However, with the in-depth study of plant diseases and pest infestations, deep learning technology also has limitations. The current agricultural infrastructure is the first limitation that is not yet sufficient to fully support the application of deep learning in the agricultural field. This requires a large number of computational resources and has a high time complexity caused by too many network parameters. The second reason is the lack of a large amount of labeled data and the subjectivity of manually labeled data in the agricultural domain. Moreover, it is difficult to obtain large-scale images of plant diseases and pests in real fields, and it is impossible to acquire images of multiple diseases and pests in one area.

At the same time, the detection of plant diseases and pests is limited by the complex background, illumination conditions, overlapping and occlusion of leaves, and similar color of foreground and background. In addition, there are other problems in the application of deep learning methods for plant pest detection, such as gradient disappearance and gradient explosion in the training process of the network, and overfitting of the network model. The most important problem is that most current deep learning networks are still considered as black-box models. Misidentification by a network for crop pest and disease detection can lead to disastrous results. For example, misidentification of the severity of crop damage can lead to the overuse of pesticides, which in turn can lead to soil contamination, environmental damage, and other vicious cycles.

In order to improve the identification and detection of crop pests and diseases, we propose this Research Topic "*Deep Learning in Crop Diseases and Insect Pests*" for the development of novel deep learning-based methods in crop pests and diseases detection. The Research Topic contains 16 original research articles based on detection of eight plant diseases, including those on grapes, strawberry, potato, pear, tomato etc., and detection of eight plant pests, focusing on tomato pest, wheat spike, etc. Eight papers developed different deep learning-based methods in this Research Topic for detection of crop diseases. Here five papers focused on specific crop, such as potato, grape, tomato, and strawberry. Yuan et al. presented an improved DeepLab v3+ deep learning network for the segmentation of grapevine leaf black rot spots to evaluate grape disease grade. The DeepLab v3+network uses ResNet101 as the backbone network, a channel attention module inserted into the residual module, and a feature fusion branch based on a feature pyramid network to fuse feature maps of different levels. Plant Village and from an orchard field test sets were used for testing the segmentation performance of the method. Li et al. proposed an integrated framework to realize the segmentation and detection of potato foliage diseases in complex backgrounds, combining instance segmentation model of Mask R-CNN to segment potato leaves in complex backgrounds, classification models of VGG16, ResNet50 and InceptionV3 to classify potato leaves, and semantic segmentation models of UNet, PSPNet, and DeepLabV3+ to divide potato leaves. It is important to detect the devastating diseases of potato early blight and late blight that affect potato planting and production. Albahli and Nawaz presentd a robust approach, namely the DenseNet-77-based CornerNet model, for the localization and classification of the tomato plant leaf abnormalities in the complex incidences of light variation, color, brightness changes, and the occurrence of blurring and noise on the 10 classes of tomato leaf images. You et al. proposed a strawberry disease detection scheme with unknown diseases, where the known strawberry diseases are detected with deep metric learning (DML)-based classifiers along with the unknown diseases that have certain symptoms. The DML-based post-filtering stage contains two different types of classifiers: softmax classifiers that are only for known diseases and the K-nearest neighbor (K-NN) classifier for both known and unknown diseases. The proposed scheme can be applied to identify disease-like symptoms caused by real known and unknown diseases or disorders for any kind of plant. Jiang et al. proposed two different but related deep learning techniques for the detection of unknown plant diseases; Open Set Recognition (OSR) and Out-of-Distribution (OoD) detection. OSR is premature to be applied in finegrained recognition tasks without outlier exposure that a certain part of OoD data (also called known unknowns) are prepared for training, where OoD detection requires intentionally prepared outlier data during training.

Moreover, two papers focused on disease detection for public datasets of crop diseases. Xia et al. devoted to plant disease identification and subtype discovery through a deep-embedding image-clustering strategy, Weighted Distance Metric, and the t-stochastic neighbor embedding algorithm (WDM-tSNE), which has been tested on public datasets of images, such as MNIST database (Modified National Institute of Standards and Technology database), PlantVillage, Aphanomyces Root Rot Image Dataset. Xu et al. proposed a transfer learning strategy with a vision transformer (ViT) model for versatile plant disease recognition, on multiple plant disease datasets. The method is first pre-trained in ImageNet with a selfsupervised loss function and with a supervised loss function in PlantCLEF2022, a large-scale dataset related to plants with 2,885,052 images and 80,000 classes. At last, one paper focused on appearance quality detection through detecting disease spots on pear fruits. Zhang et al. proposed an integrated framework combining instance segmentation, semantic segmentation and grading models, to assess the grading of the quality of the appearance of 'Huangguan' pear in a complex context. First, Mask R-CNN and Mask R-CNN with the introduction of the preprocessing module are used to segment 'Huangguan' pears from complex backgrounds; Second, DeepLabV3+, UNet and PSPNet are used to segment the 'Huangguan' pear spots to get the spots, and the ratio of the spot pixel area to the 'Huangguan' pear pixel area is calculated and classified into three grades; third, the grades of 'Huangguan' pear are obtained using ResNet50, VGG16 and MobileNetV3.

The other eight papers are dedicated to the study of insect pest detection and identification in this Research Topic. Most of papers focus on detecting multiple pests from complex background. To address the issues of pose-variant, serious overlap, dense distribution, and interclass similarity of agricultural pests, Jiao et al. proposed an end-to-end pest detection algorithm based on a deformable residual network to extract pest features and a global context aware module for obtaining region-of-interests of agricultural pests. Wang et al. addressed the issue of pest similarity in texture and scale, presented an ASP-Det to solve the texture-similarity problem and a Skip-Calibrated Convolution (SCC) module to balance the scale variation among the pest objects, and built a task-specific dataset named PestNet-AS that is collected and reannotated from PestNet dataset. Zhang et al. constructed a pest rotation detection (PRD21) using pest detection lamps in different natural environments, and performed a comparative study of image recognition through different target detection algorithms. The experimental results proved that rotation detection has a good effect on the detection and recognition rate of pests. Teng et al. proposed a robust pest detection network integrated with multiscale super-resolution (MSR) feature enhancement module to improve the detection performance of small-size, multi-scale, and high-similarity pests, and Soft-IoU (SI) mechanism to emphasize the position-based detection requirement by distinguishing the performance of different predictions with the same Intersection over Union (IoU). In addition, authors constructed a large-scale light-trap pest dataset (named LLPD-26), containing 26-class pests and 18,585 images with high-quality pest detection and classification annotations. Moreover, most methods required large-scale well-labeled pest datasets for their base-class training and novel-class fine-tuning, which hindered significantly the further promotion of deep convolutional neural network approaches in pest detection. Therefore, Wang et al. introducted a few-shot pest detection network to detect rarely collected pest species in natural scenarios. They presented a prior-knowledge-auxiliaried architecture for few-shot pest detection, built a hierarchical few-shot pest detection dataset in the wild in China over the past few years, and proposed a pest ontology relation module to combine insect taxonomy and inter-image similarity information.

Three papers focus on specific type of insect pests. Zhou et al. aimed at wheat spike detection and proposed a Transformer-based network named Multi-Window Swin Transformer (MWSwin Transformer) to use the ability of feature pyramid network to extract multi-scale features, integrated with self-attention mechanism by window strategy. They also proposed a Wheat Intersection over Union loss by incorporating the

Euclidean distance, area overlapping, and aspect ratio. Furthermore, they constructed a wheat spike detection data set (WSD-2022) to evaluate the performance of the proposed methods. Liu et al. aimed at tomato pest detection and proposed a tomato pest identification algorithm based on an improved YOLOv4 fusing triplet attention mechanism (YOLOv4-TAM) with a focal loss function to address the issue of imbalances in the number of positive and negative sample images. They also used the K-means++ clustering algorithm to obtain a set of anchor boxes that correspond to the pest data set. Kalfas et al. aimed to detect Drosophila suzukii infestation in fruit orchards. They trained convolutional neural network (CNN) classifiers with frequency (power spectral density) and time-frequency (spectrogram) representations to distinguish D. suzukii insects from one of their closest relatives, Drosophila Melanogaster, based on their wingbeat patterns recorded by the optical sensor.

This Research Topic demonstrates several deep learning-based methods to address the issues of crop pest and disease detection occurred in real and complex world, and demonstrates how the use of deep learning methods can improve the understanding and detection of crop pests and diseases. Some research can also be used to decrease the loss of crop yield loss and increase crop production. We welcome everyone to explore the 16 research papers and improve their works in the future.

Author contributions

PC drafted the manuscript. RW and PY checked the manuscript and suggested modifications. All authors contributed to the Editorial and approved the submitted version.

Funding

This study was supported by the National Natural Science Foundation of China (Nos. 62072002, 62273001), the Major Special Science and Technology Project of Anhui Province (202003A06020016), and the Special Fund for Anhui Agriculture Research System.

Acknowledgments

The authors are very grateful to the authors who contributed to this Research Topic, reviewers for evaluating the work, and the Frontiers editorial staff for their guidance.

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.

ORIGINAL RESEARCH
published: 03 January 2022
doi: 10.3389/fpls.2021.789630

A Novel Computational Framework for Precision Diagnosis and Subtype Discovery of Plant With Lesion

Fei Xia[1†], Xiaojun Xie[1,2†], Zongqin Wang[1], Shichao Jin[3,4], Ke Yan[5] and Zhiwei Ji[1,2]*

[1] College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China, [2] Center for Data Science and Intelligent Computing, Nanjing Agricultural University, Nanjing, China, [3] Plant Phenomics Research Centre, Academy for Advanced Interdisciplinary Studies, Regional Technique Innovation Center for Wheat Production, Key Laboratory of Crop Physiology and Ecology in Southern China, Ministry of Agriculture, Nanjing Agricultural University, Nanjing, China, [4] Collaborative Innovation Centre for Modern Crop Production co-sponsored by Province and Ministry, Jiangsu Key Laboratory for Information Agriculture, Nanjing Agricultural University, Nanjing, China, [5] Department of Building, School of Design and Environment, National University of Singapore, Singapore, Singapore

OPEN ACCESS

Edited by:
Peng Chen,
Anhui University, China

Reviewed by:
Guangming Zhang,
University of Texas Health Science Center at Houston, United States
Lisha Zhu,
University of Chicago, United States
Xin Gao,
Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences (CAS), China

***Correspondence:**
Zhiwei Ji
Zhiwei.Ji@njau.edu.cn

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

***Received:** 05 October 2021*
***Accepted:** 03 November 2021*
***Published:** 03 January 2022*

Citation:
Xia F, Xie X, Wang Z, Jin S, Yan K and Ji Z (2022) A Novel Computational Framework for Precision Diagnosis and Subtype Discovery of Plant With Lesion. Front. Plant Sci. 12:789630. doi: 10.3389/fpls.2021.789630

Plants are often attacked by various pathogens during their growth, which may cause environmental pollution, food shortages, or economic losses in a certain area. Integration of high throughput phenomics data and computer vision (CV) provides a great opportunity to realize plant disease diagnosis in the early stage and uncover the subtype or stage patterns in the disease progression. In this study, we proposed a novel computational framework for plant disease identification and subtype discovery through a deep-embedding image-clustering strategy, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm (WDM-tSNE). To verify the effectiveness, we applied our method on four public datasets of images. The results demonstrated that the newly developed tool is capable of identifying the plant disease and further uncover the underlying subtypes associated with pathogenic resistance. In summary, the current framework provides great clustering performance for the root or leave images of diseased plants with pronounced disease spots or symptoms.

Keywords: plant, disease diagnosis, subtype discovery, deep learning, t-SNE, image clustering

INTRODUCTION

Plants are often attacked by various pathogens (e.g., bacteria, viruses, fungi, etc.) during their growth and development (Suzuki et al., 2014), resulting in abnormal physiological and morphological changes in plants. In severe cases, it may disrupt its normal growth and development and even cause large-scale disasters, such as leaf spot disease (Ozguven and Adem, 2019), powdery mildew (Lin et al., 2019), brown spot and blast diseases (Phadikar and Goswami, 2016), and gray mold (Fahrentrapp et al., 2019). The prior symptoms of these diseases include leaf discoloration, tissue deformation or necrosis, and root atrophy, etc. Plant diseases, especially crop diseases, may cause social problems such as economic losses or food shortages in a certain area (Wilkinson et al., 2011). Therefore, early diagnosis of plant diseases, especially the precise prediction of plant disease severity and drug resistance (Bock et al., 2020), will help formulate effective control strategies, thereby effectively prevent the spread of diseases and reduce economic losses (Liang et al., 2019). To solve the above problems, many researchers made great efforts on the diagnosis of plant diseases by exploring the relationship between pathogen infection and plant disease symptoms (Bass et al., 2019; Vishnoi et al., 2021). However, these studies cannot provide real-time disease diagnosis and even evolution trajectory inference and will cause delays or misjudgments in decision-making. In recent years, plant phenomics (Tardieu et al., 2017; Pasala and Pandey, 2020) was generated, which can automatically and non-destructively obtain high-throughput plant phenotyping images (Lee et al., 2018; Li et al., 2020), which makes computer-aided rapid diagnosis and real-time monitoring of plant diseases possible.

Computationally, phenomics-based plant disease diagnosis can be grouped into two categories, one is *semantic* feature-based models, and the other is *non-sematic* feature-based models (e.g., deep learning [DL] models). The first category (conventional image processing) is characterized by the features of color (Gaikwad and Musande, 2017), texture (Hossain et al., 2019; Ismail et al., 2020), and shape (Chouhan et al., 2020) extracted from the lesion area of the phenotypic images to achieve disease diagnosis and prediction. For example, Zhang et al. (2017) segmented the lesions from the leaf images and extracted the shape and color features for disease recognition in cucumber. Moreover, some researchers realized the automatic diagnosis of plant diseases through a classifier built with texture features (Hossain et al., 2019; Ismail et al., 2020). In addition, computer vision (CV) and machine learning were applied to quantify root traits in real time for precision plant breeding (Rahaman et al., 2019; Falk et al., 2020). However, the variation of plant phenomics and the dependence of prior knowledge always limit the generalization of this type of method to different plant diseases. In recent years, DL has been widely used in image classification and clustering (Hu et al., 2020; Saleem et al., 2020). The representative characterizations of DL-based models include powerful capabilities for feature extraction, low dependence on domain knowledge, and high predictive accuracy (Too et al., 2019; Lee et al., 2020). In the past few years, DL was used to analyze the phenomics of plant disease. Various convolutional neural network (CNN) models were developed as the image multi-class classifiers to distinguish different plant leaf diseases from high-throughput phenomics (Brahimi et al., 2018; Zhang et al., 2019). Furthermore, DL is also very effective for grading the severity of plants with the same disease (Verma et al., 2020). Liang et al. (2019) combined ResNet50 (Wen et al., 2020) model and Shufflenet-V2 (Ghosh et al., 2020) to build a PD^{2}SE-Net network model, which realized the classification of plant diseases and the prediction of disease severity. Yu et al. (2006) applied VGG16 model on diseased leaf images for grading the severity of apple black rot (Wang et al., 2017). Although DL models are widely studied for plant disease diagnosis, they still face obvious challenges, such as poor generalization, unexplainable features, and high dependence on abundant training samples.

In this study, we proposed a novel image clustering method for both plant disease classification and subtype discovery. Firstly, all the original plant images were preprocessed to amplify the sample size. Secondly, we established a deep CNN to extract the features of phenotypic images. Finally, we designed a clustering strategy by integrating a Weighted Distance Metric (WDM) and the t-stochastic neighbor embedding algorithm, named "WDM-tSNE." To validate the effectiveness, we applied the proposed method on a batch of public plant image datasets, namely, Modified National Institute of Standards and Technology (MNIST) (Deng, 2012), Aphanomyces Root Rot (ARR) in lentil (Marzougui et al., 2019), cherry powdery mildew, strawberry leaf scorch disease, and three types of tomato disease from *PlantVillage* dataset (Mohanty et al., 2016). The experimental results show that our method obtained high performance on plant disease classification and subtype discovery. In particular, the WDM-tSNE strategy provides better clustering accuracy than the standard tSNE.

RELATED WORK

In this section, we briefly review the related work of plant disease diagnosis on semantic feature-based models, and non-sematic feature-based models.

Semantic Feature-Based Models

The general idea of this kind of method includes four steps: (1) image preprocessing; (2) lesion segmentation; (3) image features are defined and extracted for describing the pathology signatures of the lesion regions; and (4) the image samples are classified by using a machine-learning model (Vishnoi et al., 2021). Considering the fact that the accuracy of lesion segmentation directly affects the sample classification, many researchers used various image-segmentation strategies to achieve the extraction of the target regions, such as threshold-based segmentation methods (Tete and Kamlu, 2017), edge detection algorithms (Wang et al., 2018), and spatial clustering methods (Guan et al., 2017). After obtaining the lesion regions, researchers often define the color, texture, or shape features to characterize the disease state of each sample. Gaikwad and coworkers applied K-means to segment the lesion regions in the wheat leaf images and extracted the color features, such as color histogram (Stricker, 1994), color moments (Poonam and Jadhav, 2015), and the texture features [e.g., gray-Level co-occurrence matrix [GLCM] (Gadelmawla, 2004)] to construct a support-vector machine (SVM) model for the classification of wheat diseases (Gaikwad and Musande, 2017). Ali et al. (2017) applied Delta E (ΔE) segmentation to process the leave images of diseased potatoes and extract color and texture features based on red, green, and blue (RGB), hue, saturation, value (HSV), and local binary patterns (LBP) to implement the classification of early blight and late blight (Ismail et al., 2020). Ayyub and Manjramkar (2019) successfully classified the apple fruit diseases via a multi-class model by integrating improved sum and difference histogram (ISADH), completed local binary pattern (CLBP), and other color and texture features.

In general, this kind of method may obtain human-interpretable features and thus provide good performance on some plant diseases. However, three drawbacks exist. First, the calculation procedure of these methods is complicated. Second, these methods are highly dependent on expert knowledge. Third, they do not work well for real-time detection.

Non-sematic Feature-Based Models

In recent years, DL has promoted the development of CV, thereby providing new ideas for image analysis and automatic diagnosis of plant diseases. In particular, the CNN model has

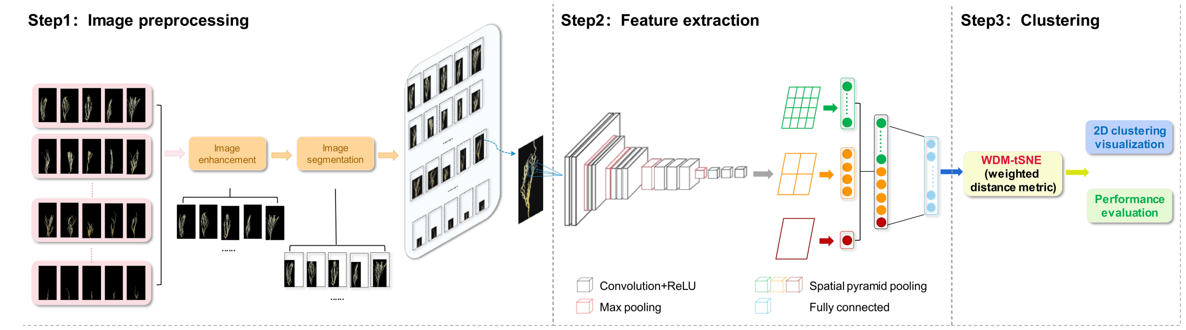


FIGURE 1 | The flowchart of the proposed framework. ReLU, Rectified Linear Unit.

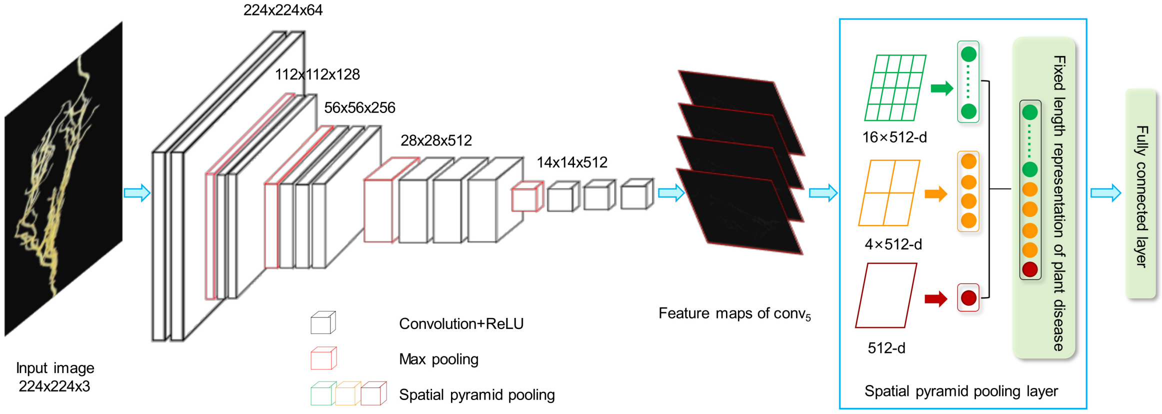


FIGURE 2 | CNN-based network for feature extraction. ReLU, Rectified Linear Unit; CNN, convolutional neural network.

been widely studied by researchers because of its powerful image processing and feature extraction capabilities and without the prior knowledge of domain experts (Syed-Ab-Rahman et al., 2021). At present, most of the existing works applied CNN, combined with transfer learning (Too et al., 2019) to implement plant disease diagnosis. Zhang et al. (2018) used two improved CNN models, GoogleNet and Cifar10, to classify nine types of corn diseases and obtain high accuracy. To reduce the number of parameters, Rahman et al. (2020) constructed a two-stage light CNN framework Simple-CNN to identify rice diseases with high accuracy. Moreover, other researchers made great efforts to develop novel computational models for predicting the severity of plant disease. For example, José et al. (2020) used five types of CNN models (AlexNet, GoogleNet, VGG16, ResNet50, and MobileNetV2) to estimate the severity of coffee leaf biotic stress. In addition, deep learning was also widely used to identify the diseases of fruit, root, and stem. Tan et al. (2016) presented a CNN model to recognize lesion images of diseased apples, such as scab skin, black rot, scar skin, and ring spot (Wenxue Tan, 2020). Nikhitha et al. (2019) used the Inception v3 model to detect the grades of infections in fruits (e.g., apple, banana, and cherry, *etc.*) based on color, size, and shape of the fruit (Nikhitha et al., 2019). Tusubira et al. (2020) achieved the automated scoring for root necrosis of diseased cassava by using deep CNN with semantic segmentation, which is done by classifying the necrotized and non-necrotized pixels of cassava root cross-sections without any additional feature engineering. Compared with the first category, DL models achieve higher recognition accuracy. However, we identify three limitations. First, they require large amounts of labeled data; second, they are overly sensitive to changes in the image; and third, the non-semantic features are hard to be explained.

To address the above limitations, we proposed an efficient pipeline for both disease diagnosis and severity estimation of plants with the lesion. A DL model combined with a novel clustering strategy contributes to higher prediction accuracy and lower computational cost.

MATERIALS AND METHODS

The proposed computational framework includes three steps (**Figure 1**) and will be explained in detail in the following subsections.

Image Preprocessing

Before extracting features, each image needs to be preprocessed, such as image enhancement and image segmentation. Image augmentation is to increase the diversity of samples (Halevy et al., 2009). we use horizontal flip (Connor Shorten,

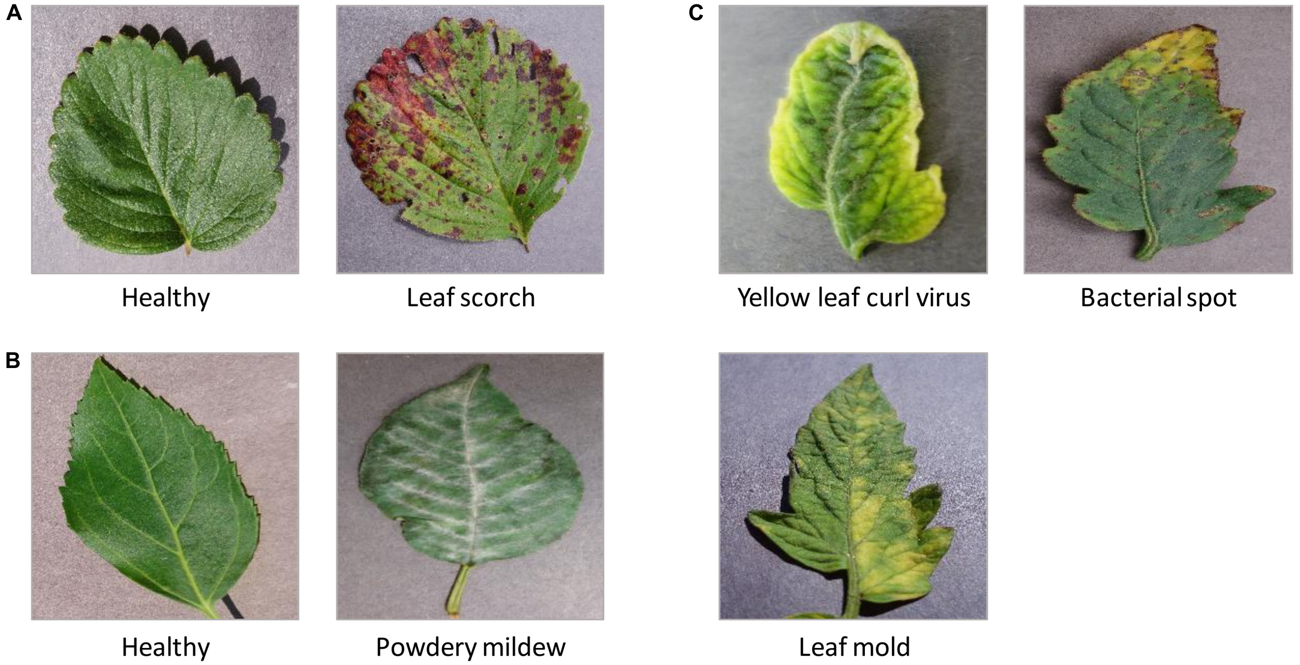


FIGURE 3 | The representative leaf images with diseases from PlantVillage. **(A)** Leaf scorch of strawberry; **(B)** cherry powdery mildew; **(C)** three types of leaf diseases on tomatoes: a bacterial spot of tomato, tomato leaf mold, and tomato yellow leaf curl virus (TYLCV).

2019) and affine transformation (Shen et al., 2019) on each image to enhance the size and quality of training datasets so that better DL models can be built. The purpose of image segmentation is to obtain areas related to plant tissues (root or leaf) from the original images. Therefore, the irrelevant region needs to be removed. In this study, we detected the relevant area by traversing all the pixels in each image and obtained the smallest circumscribed rectangle (Yu et al., 2006) of the outer contour of a plant tissue.

Feature Extraction

We developed a CNN model to extract the features from the plant images with the disease. The whole CNN model includes three layers: convolution layers, the spatial pyramid pooling (SPP) layer, and fully connected layer. The extracted high-dimensional features were further used to cluster the images with different severity levels. **Figure 2** shows the details of the feature extraction process using the lentil images as an example.

Creating the Feature Maps

As shown in **Figure 2**, the first step is to create the feature maps from each input image by using a series of convolutional, non-linear, and pooling. The convolutional layers can learn the low-level features, such as edges and curves, which provide the CNN with the important property of "translation invariance" (Kayhan and van Gemert, 2020). That makes it unnecessary to focus on the location of the disease on the plant roots or leaves and let alone to divide up the area of the spot. Convolution is done by applying filters to the input image data, which decreases its size (Yamashita et al., 2018). An additional operation called the Rectified Linear Unit (ReLU) (Atila and Sengür, 2021) was used after every convolution operation to generate a non-linear relationship between input and output. Finally, The pooling layer is used for secondary feature extraction, retaining the main features, reducing parameters, saving computing resources, preventing over-fitting, and improving model generalization (Suarez-Paniagua and Segura-Bedmar, 2018). Here, we define a spatial neighborhood with a 2 × 2 window and take the largest element from the rectified feature map within that window. Max pooling not only reduces the dimensionality of each feature map but also retains the most important information. Comparing with the typical VGG16 model (Qassim et al., 2018), the network structure of our model retains all the convolutional and pooling layers and the activation method, but removes three fully connected layers.

Let us say we have a plant image, and its size is 224 × 224. The representative array of this image will be 224 × 224 × 3 (3 refers to the channels of RGB). After the first operation of convolution, we obtained the feature maps as an array with 224 × 224 ×64. Passing this array through four convolutional layers, we finally obtained 512 feature maps with 14 × 14. The final output feature map (14 × 14 × 512) will be converted into one-dimensional vector.

Considering the fact that a CNN model may take time to train on large datasets, transfer learning (Pan and Yang, 2010) was considered in our study to re-use the model weights from pre-trained ImageNet (Krizhevsky et al., 2012) tasks. Here, we directly use the five convolutional layers from the entire architecture of the pre-trained the VGG16 model on ImageNet datasets.

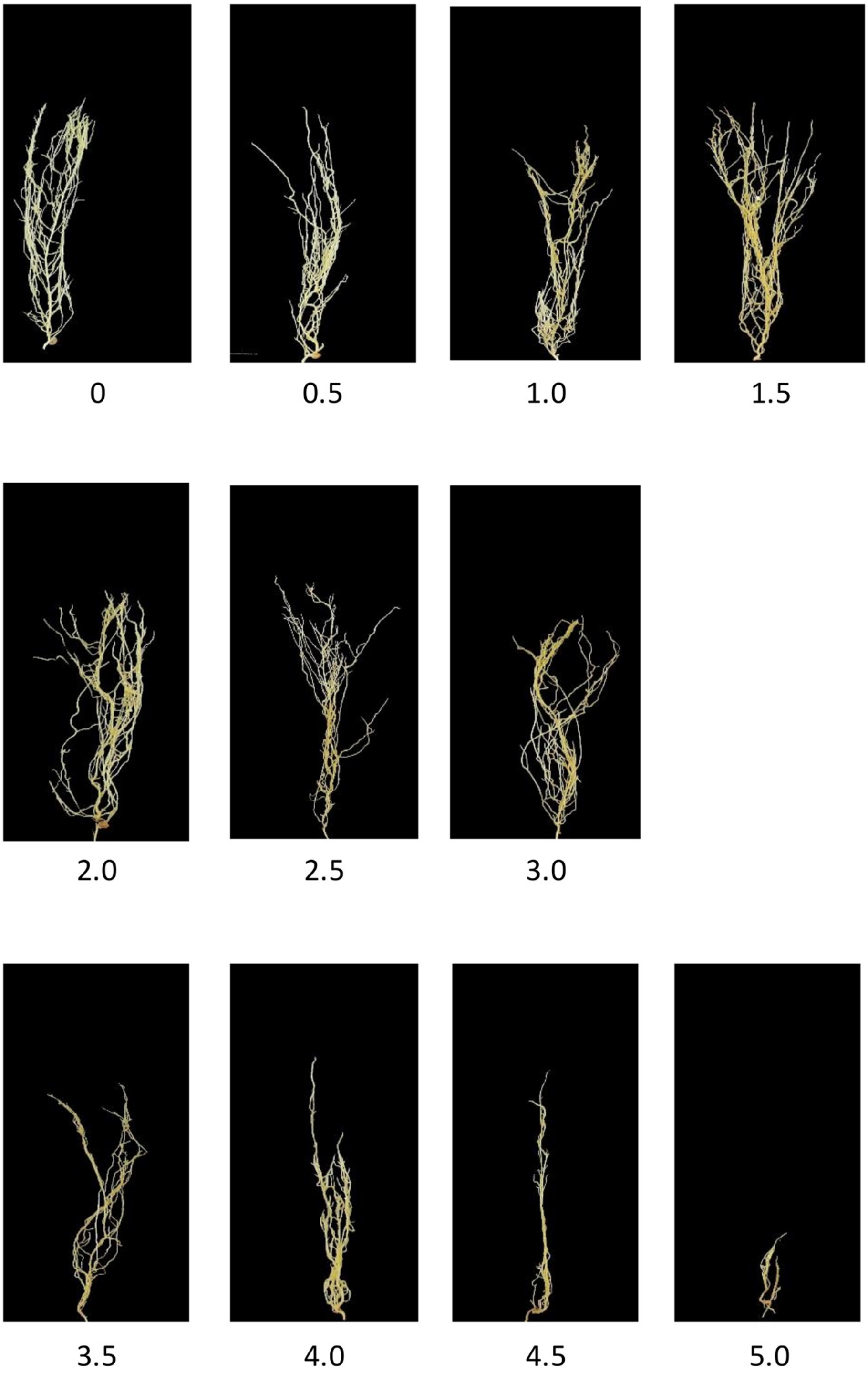


FIGURE 4 | Aphanomyces root rot disease severity scale.

Converting the Feature Maps to a Fixed Length Feature Vector

In this step, we convert all the two-dimensional feature maps to a single long continuous linear vector because the fully connected layer expects to receive one-dimensional inputs (Gu et al., 2018). Here, we introduce SPP (He et al., 2015) layer to remove the limitation of the fixed size of the images. The SPP layer was placed after the last convolutional layer and aggregated multi-scale

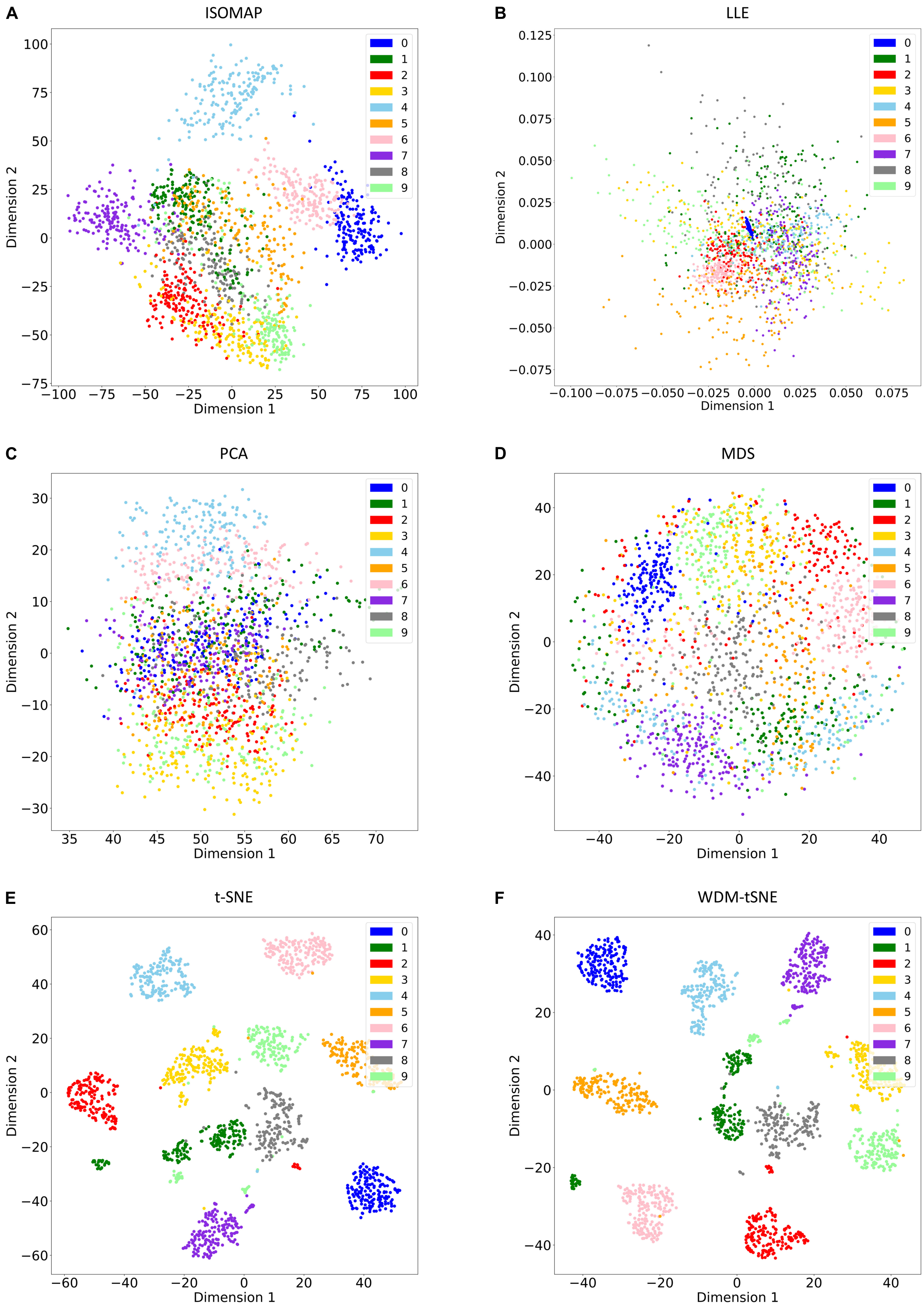


FIGURE 5 | The plots for the MNIST dataset based on six dimensionality reduction approaches, including **(A)** Isomap, **(B)** LLE, **(C)** PCA, **(D)** MDS, **(E)** t-SNE, and **(F)** WDM-tSNE. MNIST, Modified National Institute of Standards and Technology; ISOMAP, Isometric Mapping; PCA, Principal Component Analysis; LLE, Locally Linear Embedding; MDS, Multidimensional Scaling; t-SNE, t-Distributed Stochastic Neighbor Embedding; WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

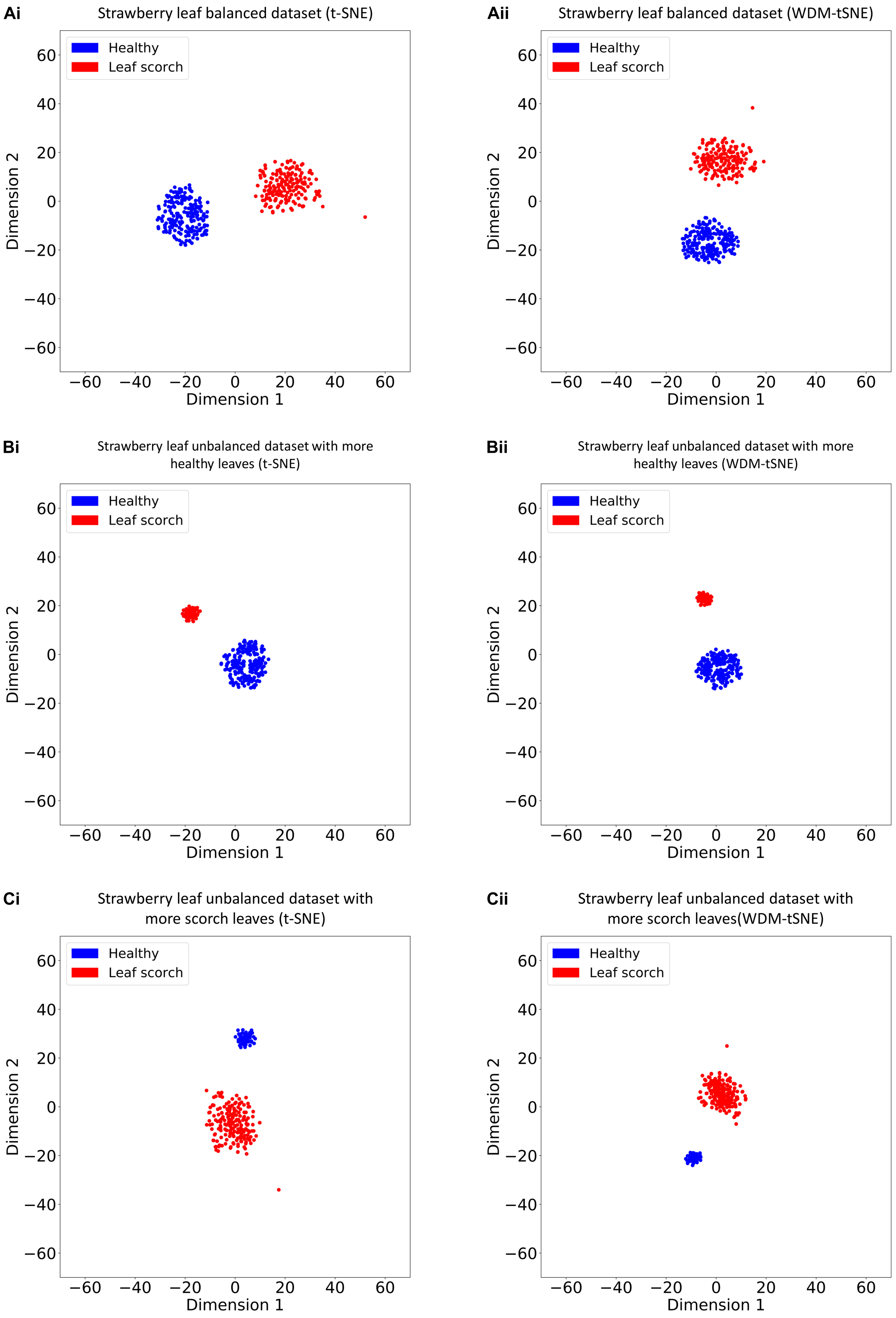


FIGURE 6 | The plots for the **(A)** balanced or **(B–C)** unbalanced datasets of strawberry leaf scorch based on (i) t-SNE and (ii) WDM-tSNE. WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

TABLE 1 | The performance of WDM-tSNE on the multiple datasets of strawberry.

	Balanced dataset		Unbalanced dataset with more healthy leaves		Unbalanced dataset with more scorch leaves	
	t-SNE	**WDM-tSNE**	**t-SNE**	**WDM-tSNE**	**t-SNE**	**WDM-tSNE**
Silhouette coefficient	0.723	0.729	0.755	0.788	0.725	0.799
Calinski-Harabasz	2014.769	2106.024	1026.573	1353.780	734.780	1391.950
Davies-Bouldin Index	0.4070	0.399	0.271	0.233	0.302	0.218

WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

features. As shown in **Figure 2**, each feature map (14 × 14) is divided into a lattice of n × n (n = 1,2,4) and each lattice is pooled, resulting in 21 features. This also means that the 512 feature maps of an original image are finally represented as a one-dimensional vector with a length of 10,752 (21 × 512). The output of the fully connected layer is 4,096, which means each image matrix will be converted to a feature vector with length 4,096 for clustering calculation.

Image Clustering

As mentioned above, each original image was finally represented as a 4,096 × 1 vector after the feature extraction process. The clustering of a group of original images is thus equivalent to a clustering task on a set of data points with a dimension of 4,096. Considering the fact that t-SNE is an efficient algorithm based on manifold learning for unsupervised clustering (Van der Maaten and Hinton, 2008), we designed an improved t-SNE algorithm for image clustering to classify plant diseases and graded the severity of a disease. The standard t-SNE algorithm assumes that the samples are distributed on a statistical manifold and converts the Euclidean distance between the samples into conditional probabilities to characterize the similarity between the samples (Talwalkar et al., 2008). However, the variables in the high-dimensional space often present complex non-linear relationships, and the Euclidean distance does not well reflect the real distribution of the samples, thus affecting its projection to the low-dimensional space. Within a manifold space, the Euclidean distance metrics can only represent the real distance between samples in a very small neighborhood subspace (Zhang et al., 2011).

Taken above together, we think that only the data points in the local neighborhood are applicable to the Euclidean distance, and they should be given greater weight in the conditional probability transformation. In this study, we adopted a WDM strategy to improve the t-SNE algorithm (WDM-tSNE) so that the similarity between samples can be better reflected after they are projected to a low-dimensional space. The details of WDM-tSNE are described as follows:

Firstly, we construct the distance matrix D of all the samples, where the element d_{ij} represents the distance between any two points X_i and X_j [Eq. (1)]:

$$d_{ij} = \sum\nolimits_{k=0}^{n} (X_{ik} - X_{jk})^2 \tag{1}$$

All the non-zero elements d_{ij} $(i \neq j)$ are sorted in ascending order, and the distance value that ranks approximately 10% is selected as the threshold of the neighborhood relationship, denoted asθ. If $d_{ij} \leq \theta$, X_i and X_j have a neighbor relationship and weighting their distance will make them closer in the low-dimensional space. Therefore, we define a WDM strategy to adjust the distance coefficient l between any pair of samples X_i and X_j:

$$l = \begin{cases} \dfrac{d_{ij} - d_{\min} + c}{d_{\max} - d_{\min}}, & if \;\; d_{ij} \leq \theta \\ 1, & otherwise \end{cases} \tag{2}$$

Under the Gaussian distribution centered on the point X_i, the conditional probability $P_{j|i}$ is used to measure the similarity between X_i and X_j. In other words, $P_{j|i}$means the probability that X_i chooses X_j as its neighbor. We thus construct conditional probability $P_{j|i}$ for X_i and X_j, and the probability distribution is defined as Eq. (3):

$$P_{j|i} = \frac{\exp(-l * ||X_i - X_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-l * ||X_i - X_k||^2/2\sigma_i^2)} \tag{3}$$

From Eq. (3), we have $P_{i|i} = 0$. Assuming that the points Y_i and Y_j in the low-dimensional space are projected from X_i andX_j, the similarity between the points Y_i and Y_j can be defined as:

$$Q_{j|i} = \frac{\exp(-||Y_i - Y_j||^2)}{\sum_{k \neq i} \exp(-||Y_i - Y_k||^2)} \tag{4}$$

According to the above description, we expect that if two points are similar in the high-dimensional space, they should be closer after being projected to the low-dimensional space. Here, we use Kullback-Leibler divergence (Van der Maaten and Hinton, 2008) to measure the difference between the above two conditional probability distributions and define the following objective function as Eq. (5):

$$C = \sum_i KL(P \;||\; Q_i) = \sum_i \sum_j P_{j|i} \log \frac{p_{j|i}}{Q_{j|i}} \tag{5}$$

However, the KL divergence (Kullback-Leibler divergence) is asymmetric $[KL(P||Q) \neq KL(Q||P)]$ (Afgani et al., 2008), which will cause the gradient calculation to be complicated. To optimize the KL divergence in SNE, t-SNE adopts symmetric SNE, that is, assuming $P_{j|i} = P_{i|j}$ and $Q_{j|i} = Q_{i|j}$. The conditional probability

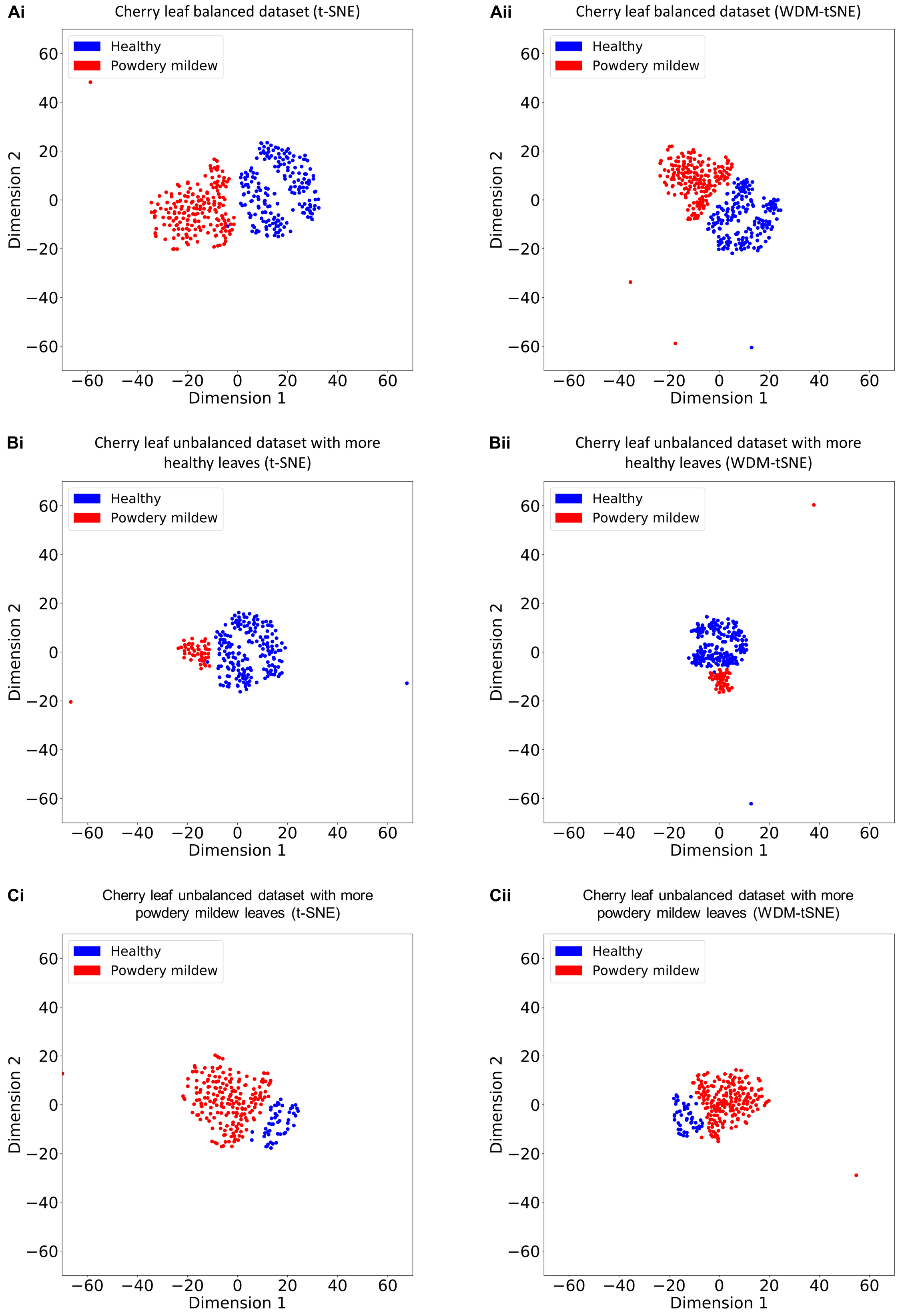


FIGURE 7 | The plots for the **(A)** balanced and **(B–C)** unbalanced datasets of the cherry leaf with powdery mildew based on (i) t-SNE and (ii) WDM-tSNE. WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

TABLE 2 | The performance of WDM-tSNE on the multiple datasets of cherry.

	Balanced dataset		Unbalanced dataset with more healthy leaves		Unbalanced dataset with more scorch leaves	
	t-SNE	**WDM-tSNE**	**t-SNE**	**WDM-tSNE**	**t-SNE**	**WDM-tSNE**
Silhouette coefficient	0.496	0.494	0.362	0.369	0.354	0.361
Calinski-Harabasz	511.877	540.454	81.172	103.808	119.424	131.842
Davies-Bouldin Index	0.773	0.764	0.969	0.841	0.836	0.829

WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

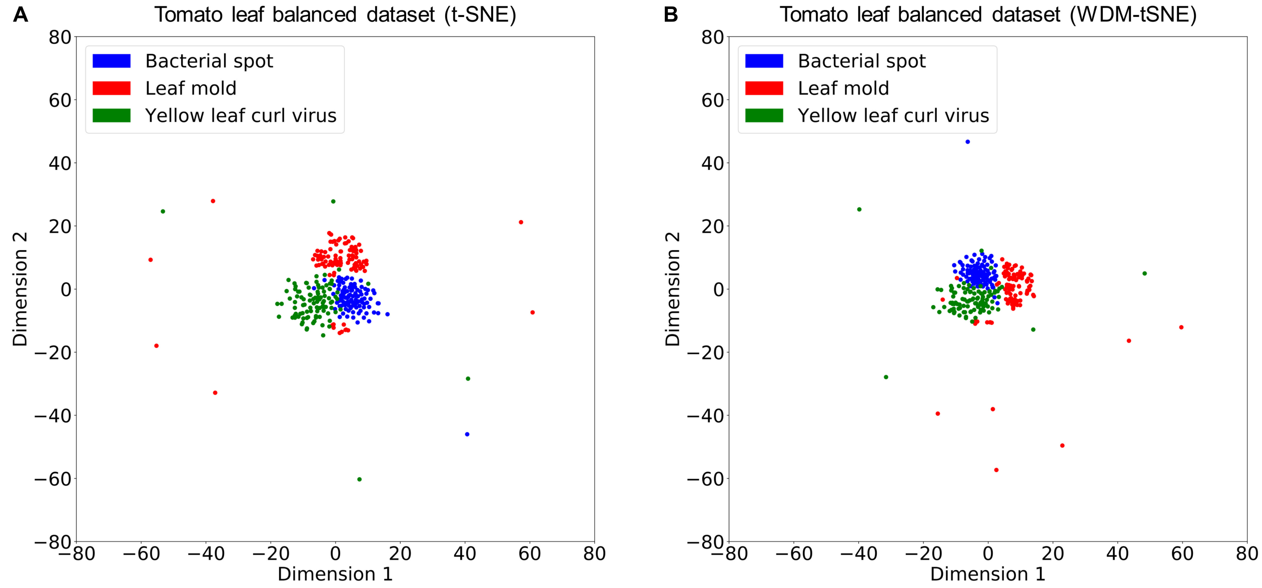


FIGURE 8 | The plots for the balanced datasets of three tomato leaf diseases based on **(A)** t-SNE and **(B)** WDM-tSNE. WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

$p_{j|i}$ can be replaced with the joint probability p_{ij}:

$$p_{ij} = \frac{\exp(-l * ||X_i - X_j||^2/2\sigma^2)}{\sum_{k\neq S}\exp(-l * ||X_k - X_S||^2/2\sigma^2)} \tag{6}$$

If X_i is an abnormal point, all the d_{ij} will be very large and may impact the calculation of P_{ij}. Therefore, we define the joint probability distribution P_{ij} as:

$$P_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{7}$$

To make the points in the same cluster in the low-dimensional space more closer and the points in different clusters are more distant (Van der Maaten and Hinton, 2008), the long-tailed t-distribution is used instead of the Gaussian distribution. The joint probability of two points in the low-dimensional space can be defined as:

$$Q_{ij} = \frac{(1 + ||y_i - y_j||^{-1})}{\sum_{k\neq S}(1 + ||y_k - y_S||^2)^{-2}} \tag{8}$$

TABLE 3 | The performance of WDM-tSNE on the dataset of tomato disease.

	Balanced dataset	
	t-SNE	**WDM-tSNE**
Silhouette coefficient	0.263	0.273
Calinski-Harabasz	25.538	40.427
Davies-Bouldin Index	1.615	1.279

WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

Therefore, Eq. (5) can be written as Eq. (9):

$$C = KL(P||Q) = \sum_i \sum_j P_{ij} \log_{Q_{ij}}^{p_{ij}} \tag{9}$$

The formula (9) can be optimized by using the gradient descent strategy shown in formula (10):

$$\frac{\delta c}{\delta y_i} = 4\sum_j (P_{ij} - Q_{ij})\,(Y_i - Y_j)\,\left(1 + ||Y_i - Y_j||^2\right)^{-1} \tag{10}$$

Finally, all the point pairs of X_i and X_j in the high-dimensional space are projected to the two-dimensional space as Y_i and Y_j.

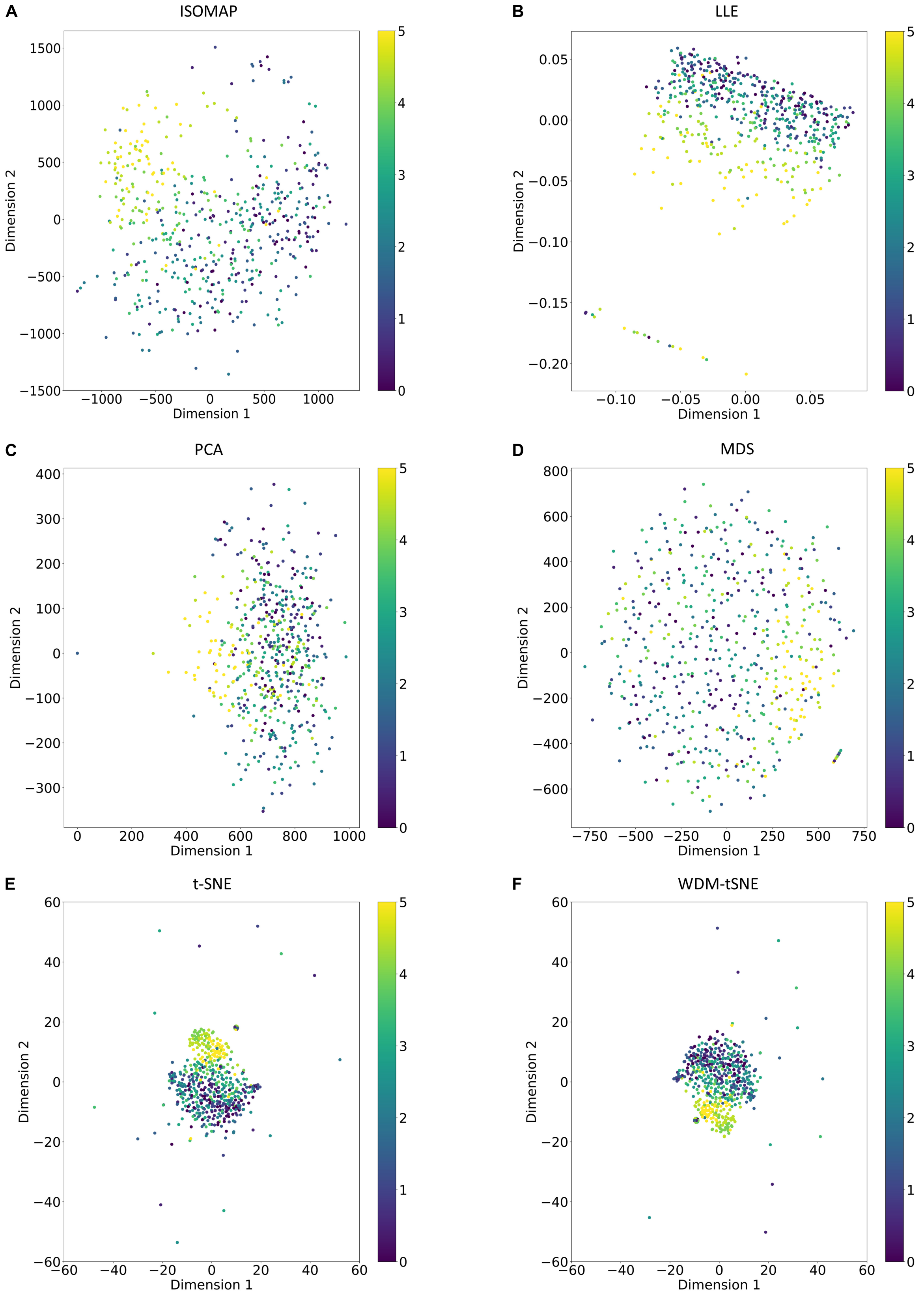


FIGURE 9 | The plots for the balanced dataset of ARR based on six dimensionality reduction approaches, including **(A)** Isomap, **(B)** LLE, **(C)** PCA, **(D)** MDS, **(E)** t-SNE, and **(F)** WDM-tSNE. The samples with 11 rates were plotted. ARR, Aphanomyces Root Rot.

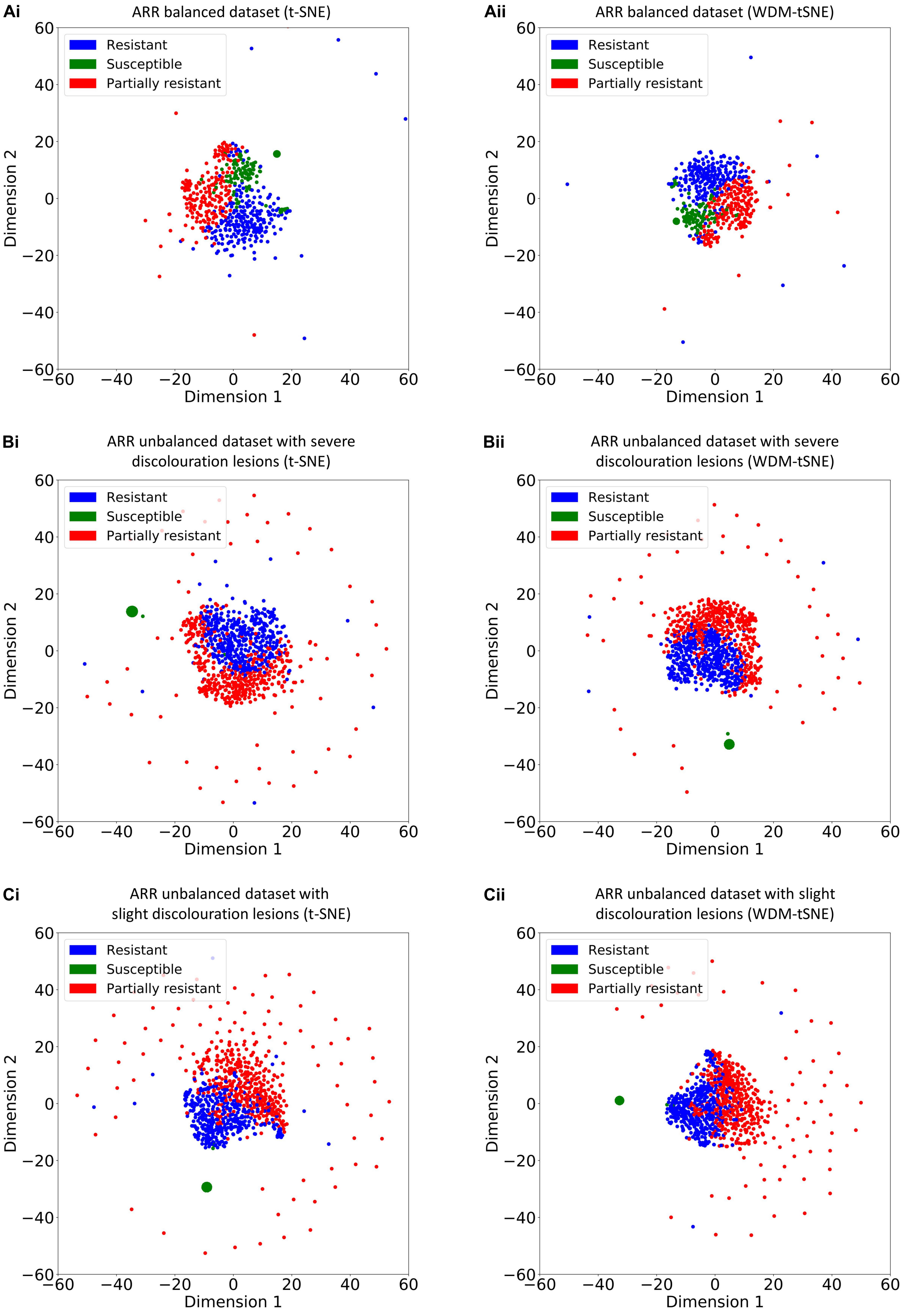


FIGURE 10 | The plots for the **(A)** balanced and **(B–C)** unbalanced datasets of ARR are based on (i) t-SNE and (ii) WDM-tSNE. WDM-tSNE, Aphanomyces Root Rot; Weighted Distance Metric and the t-stochastic neighbor embedding algorithm; ARR, Aphanomyces Root Rot.

The visualization of all the points Y can show the clustering effect of image samples.

Experimental Protocol

In this section, we introduced the experimental protocol designed for the validation of the proposed approach, such as data collection, simulation design, evaluation metric, and parameter optimization.

Data Collection

The MNIST database (Modified National Institute of Standards and Technology database) (Baldominos et al., 2019), a large database of handwritten digits, was used for data collection, which is not only used for training various image processing systems but also for testing machine-learning algorithms (Pastor-López et al., 2021). Currently, the MNIST database contains 60,000 training images and 10,000 testing images. In this study, we selected a data-subset Scikit-learn containing 1,797 8 × 8 digital images to test our proposed approach for image clustering.

PlantVillage (Barbedo, 2019) is a large, open-access image database. Currently, it stores 54,306 leaf images, associate with 26 plant diseases of 14 species (Albert et al., 2017; Brahimi et al., 2017; Ferentinos, 2018). This dataset is widely employed to test the performance of machine-learning models (Wang et al., 2017). In this study, we mainly focused on the following image sets from PlantVillage: (1) three types of leaf diseases on tomatoes (**Figure 3C**), such as bacterial spot of tomato (Adhikari et al., 2020), tomato leaf mold (Rivas and Thomas, 2005), and tomato yellow leaf curl virus (TYLCV) (Prasad et al., 2020); (2) cherry powdery mildew (Gupta et al., 2017; **Figure 3B**); (3) leaf scorch of strawberry (Dhanvantari, 1967; **Figure 3A**).

Aphanomyces Root Rot Image Dataset (Marzougui et al., 2019) contains up to 6,460 lentil images with root rot. ARR is a soil-borne disease that severely reduces lentil production. Based on the percentage of the brown discoloration area of the root and the softness of the hypocotyl (McGee et al., 2012), Marzougui et al. (2019) labeled the relative severity of all the root images using 0–5 disease scoring scale (McGee et al., 2012). For example, A score of 0 means that there are no obvious symptoms and good resistance to root rot; 1.5 means that the root has 15–25% of partial discoloration lesions; 3.5 means that the entire root has completely turned brown, and the hypocotyl has some symptoms. Eleven representative images with scores from 0 to 5 are shown in **Figure 4**. Furthermore, Marzougui et al. (2020) proposed three subtypes of ARR based on the visual score to evaluate the Rot severity: (1) resistant subtype with score 0–1.5; (2) partially resistant with score 2–3; (3) susceptible subtype with score 3.5–5. In this study, we selected 950 representative images of ARR for experimental simulation.

Simulation Design

Firstly, 1,797 digital images from MNIST were used to test the proposed method. Furthermore, we also compared the WDM-tSNE with the other five clustering strategies on MNIST. Secondly, a binary clustering test was further implemented on 400 strawberry and 400 cherry images to identify the diseased samples from the control. Thirdly, 300 tomato images were selected to test the clustering performance of our approach on three different diseases. Finally, we selected 950 ARR images to explore potential subtypes for the lentil invaded by Aphanomyces. We manually constructed balanced datasets and unbalanced datasets to evaluate if our approach is steady. The sample size for each dataset is presented in **Supplementary File 1**.

Clustering Performance Evaluation

In this study, we defined three types of metrics to assess the clustering performance. (1) Silhouette Coefficient (SC) (Dinh et al., 2019); (2) Calinski-Harabasz Index (CHI) (Łukasik et al., 2016); (3) Davies-Bouldin Index (DBI) (Vergani and Binaghi, 2018).

Silhouette Coefficient was firstly proposed by Rousseeuw (1987), which considered both the degree of cohesion and separation to measure the clustering performance. The *SC* value of sample *j* can be calculated by Eq. (11):

$$SC_j = \frac{C_j - S_j}{\max\{C_j, S_j\}} \tag{11}$$

where C_j and S_j represent the degree of cohesion and separation, respectively. We can clearly see that good clustering means smaller C_j and larger S_j .

Calinski-Harabasz Index is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion. A larger CHI means that the clusters themselves are tighter and the cluster-clusters are more dispersed [Eq. (12)]:

$$CH = \left[\frac{\sum_{k-1}^{K} n_k ||c_k - c||^2}{K-1}\right] / \left[\frac{\sum_{k-1}^{K}\sum_{i-1}^{n_k} ||d_i - c_k||^2}{N-K}\right] \tag{12}$$

In Eq. (12), N and K are the number of samples and clusters, respectively. The variables n_k and c_k are the no. of points and centroid of the *h*-th cluster respectively, *c* is the global centroid.

Davies-Bouldin Index measures the average similarity between clusters [Eq. (13)].

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{13}$$

In Eq. (13), R_{ij} denotes the similarity between each cluster C_i and its most similar one C_j:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{14}$$

s_i denotes the average distance between each point of cluster *i*. d_{ij} denotes the distance between cluster centroids *i* and *j*.

Parameter Optimization

All the simulations were performed using Python with TensorFlow on Ubuntu 14.04 platform. The hardware setups are 2.30?GHz CPU and 4.00 GB RAM. CNN model is composed of 13 convolutional layers, and each layer uses a stacked 3 × 3 small convolution kernel to replace the large-size convolution kernel. After each convolutional layer, a 2 × 2 max pooling is used. In

TABLE 4 | The performance of WDM-tSNE on the multiple datasets of lentil.

	Balanced dataset		Unbalanced dataset with severe discoloration lesions		Unbalanced dataset with slight discoloration lesions	
	t-SNE	WDM-tSNE	t-SNE	WDM-tSNE	t-SNE	WDM-tSNE
Silhouette coefficient	0.214	0.232	0.192	0.207	0.189	0.225
Calinski-Harabasz	130.182	165.652	163.077	182.195	161.421	237.701
Davies-Bouldin Index	1.279	1.147	1.799	1.667	1.402	1.235

WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

the WDM-tSNE model, the gradient descent strategy is used to optimize the cost function C [Formula (9)], and the momentum term $\alpha^{(t)}$ is introduced to reduce the number of iterations (T). When the value of the cost function reaches 95% of the previous time, it indicates that the best result has been obtained, and the iteration is stopped. If $T < 250$, we set $\alpha^{(t)} = 0.5$; otherwise, $\alpha^{(t)} = 0.8$. The initial learning rate is set to 100, which is updated by the adaptive learning algorithm after each iteration.

RESULTS

Validation on Modified National Institute of Standards and Technology Dataset

As a golden-standard image dataset, MNIST was firstly tested by our method. A total 1,797 digital images were imported to the CNN module and converted to a 1,797 × 64 matrix. Moreover, all the 1,797 samples in a 64-D space were then projected to 2D space by six dimensionality reduction approaches, namely, ISOMAP (Isometric Mapping), PCA (Principal Component Analysis), LLE (Locally Linear Embedding), MDS (Multidimensional Scaling), t-SNE (t-Distributed Stochastic Neighbor Embedding), and the proposed WDM-tSNE (**Figure 5**). From **Figure 5**, we found that LLE and PCA obtained the worst performance of dimensionality reduction as the 10 types of digital images in 2D space cannot be separated at all. ISOMAP and MDS work better rather than the first two, but the boundaries of inter-clusters are still blurred. In contrast, t-SNE and WDM-tSNE are significantly better than the previous four methods. Particularly, multiple evaluation metrics indicates that the WDM-tSNE strategy obtained higher clustering accuracy on MNIST superior to the standard t-SNE (**Supplementary Table 1**). For the geometric distribution of the samples in 2D space, WDM-tSNE can obtain better partitions of clusters (**Supplementary Table 1**).

The Proposed Model Works Well for Disease Diagnosis

We then applied our method on 400 strawberry images with leaf scorch. **Figure 6** shows that the scorched leaf images can be easily identified from the healthy samples. Both balanced and unbalanced datasets revealed that the clustering performance is steady. **Table 1** indicates that WDM-tSNE provides better clustering performance rather than t-SNE. Similarly, we also tested our approach on 400 cherry leaf images with powdery mildew. WCD-tSNE not only makes the samples in the same cluster more concentrated, but also guarantees the distance between different clusters is as far away as possible (**Figure 7**). Compared with t-SNE, WDM-tSNE has a better clustering effect (**Table 2**). In addition to the binary-clustering, we also tested the multi-clustering situation on the leaf images of diseased tomato. **Figure 8** reveals that three distinct leaf diseases on tomatoes can be clearly identified (**Table 3**). Taken above together, we suggest that the proposed framework is an effective tool for identifying plant disease with high accuracy.

The Proposed Model Works Well for Subtype Discovery

Different from the experiments shown above, we further applied our model on 950 lentil root images infected by Aphanomyces euteiches to identify the underlying subtypes associate with Aphanomyces resistance. Firstly, 550 representative images (balanced dataset) of ARR with 11 rates of severity were projected to 2D space through six machine-learning approaches (**Figure 9**). **Figures 9E,F** shows that both t-SNE and WDM-tSNE can uncover the disease trajectory of all the samples from mild to severe. Secondly, we selected 550 images (50 samples for each rate) to test if WDM-tSNE has the ability to reveal the underlying subtypes of the plant samples with the same disease. **Figure 10** shows that three clusters are obviously detected from balanced and unbalanced datasets. The clustering performance of WDM-tSNE is superior to t-SNE (**Table 4**). In the balanced dataset with 550 samples, 231 were predicted as a mild subtype with an average score of 1.93, 205 were predicted as a partially moderate subtype (average score: 2.45), and 114 were marked as a severe subtype (average score: 3.74) (**Figure 11**). **Figure 11** also suggests that the samples with serious symptoms can be easily detected (cluster 3). However, the visual score based on the percentage of discolored lesions on the entire root system defined by Marzougui et al. may cause bias when dividing mild and moderate samples. Therefore, the data annotations based on expert knowledge are also one of the factors that affect the accuracy of the algorithm.

DISCUSSION

Plant diseases are not only a threat to food security on a global scale, but also cause disastrous consequences for smallholder farmers whose livelihoods depend on healthy crops (Mohanty et al., 2016). Identifying a disease correctly when it first appears is a crucial step for efficient disease management. Various efforts have been developed to prevent the loss of the plant due to

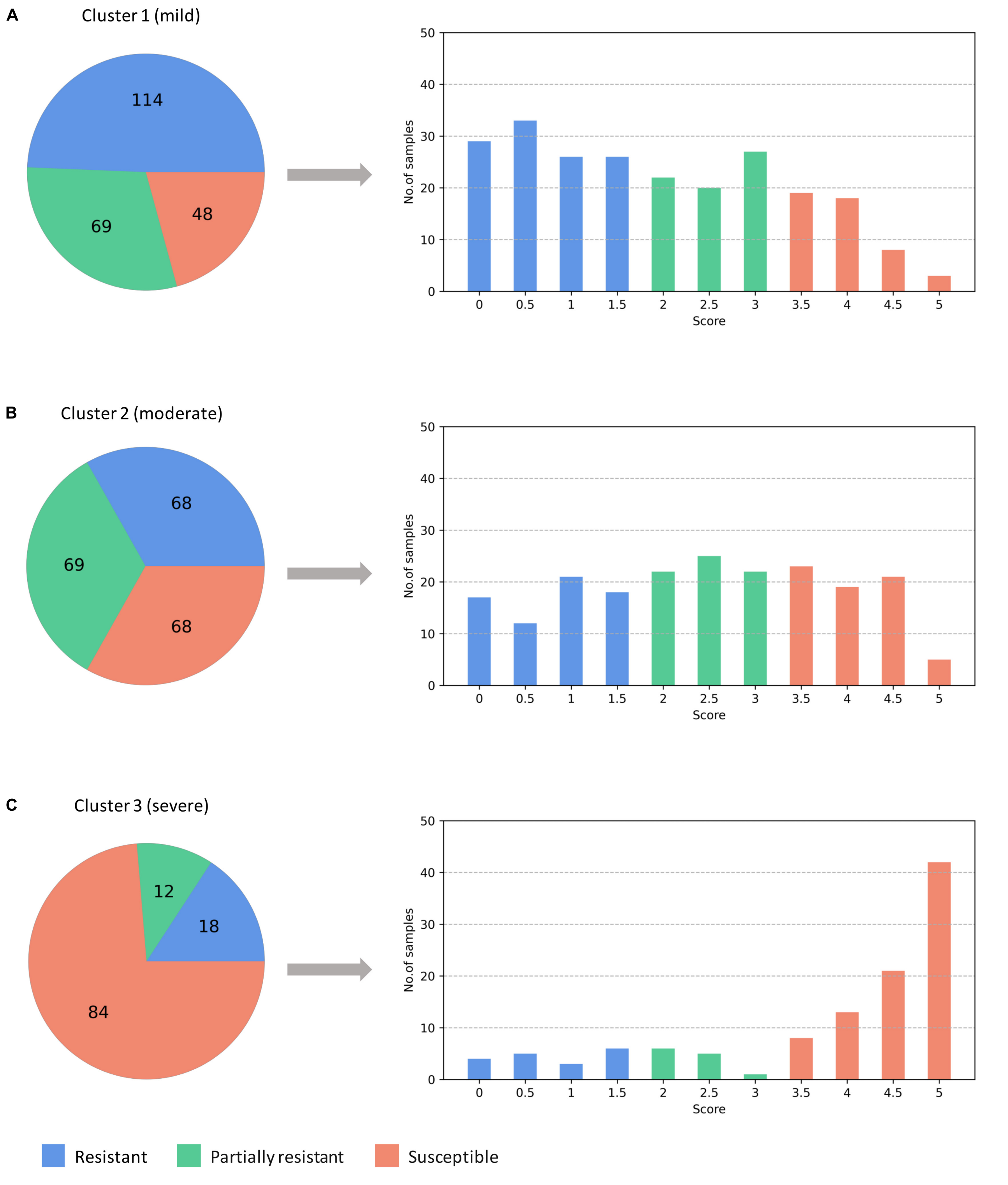


FIGURE 11 | The predicted three subtypes of ARR: **(A)** mild; **(B)** moderate; **(C)** severe. The numbers denote how many samples are assigned to one of the subtypes. ARR, Aphanomyces Root Rot.

diseases. For computer-vision-based plant diseases detection, conventional image processing or manual design of features plus classifiers are often used (Tsaftaris et al., 2016). This kind of method usually makes use of the different properties of plant disease to design the imaging scheme and chooses appropriate light sources and shooting angles, which is helpful to obtain images with uniform illumination. In the real complex natural environment, plant diseases detection is faced with many challenges, such as the small differences between the lesion area and the background, low contrast, large variations in the scale of the lesion area and various types, and a lot of noise in the lesion image (Liu and Wang, 2021). In addition, over-depend on expert knowledge to manually design the features of diseased plant often limits the generalization. In recent years, DL methods are widely used in various CV tasks for plant disease diagnosis. The most challenges of DL-based strategies include small sample size problem, fine-grained identification of small-size lesions in the early stage, and the performance under the influence of illumination and occlusion (Liu and Wang, 2021).

In this study, we proposed a computational framework for both plant disease identification and severity estimation (**Figure 1**). Firstly, we designed a CNN network structure as a feature extractor to obtain the image features of lesion regions of a diseased plant. The input original images are not required with a fixed size, which avoid the impacts of image distortion or geometric distortion on feature extraction. Secondly, a dimension reduction strategy, WDM-tSNE, was developed for the imaging clustering tasks by improving the t-SNE with WDM. WDM-tSNE successfully realized the efficient clustering of high-dimensional samples in low-dimensional space.

To validate the effectiveness, we applied the proposed model on a bunch of plant image datasets. The experimental results revealed that our method not only identifies multiple distinct diseases of the same plant but also estimates the severity of the same disease. **Figures 5**, **6** indicate that our model is able to distinguish multiple diseases in a low-dimensional space. **Figures 7**, **8** show that the diseased samples can be easily identified from the health samples. From **Figure 9**, we concluded that the proposed method can be used for subtype discovery or severity estimation from the same disease (ARR). The 10-fold cross-validation on the ARR dataset revealed that our model is robust (**Supplementary Table 2**). Furthermore, we applied our model on three small-scale datasets for cherry, strawberry, and tomato. The sample size of each class is only 50. Our analyses show that our model works well on small-scale image datasets (**Supplementary Figure 1** and **Supplementary Table 3**).

Considering the fact that the class imbalance may impact the clustering performance, we constructed multiple balanced and unbalanced datasets for ARR (lentil), cherry, and strawberry (**Supplementary File 1**). Regardless of binary-class or multi-class, WDM-tSNE shows better clustering performance than t-SNE (**Tables 1–4**). It indicates that the sample variation does not affect the performance of our method.

The proposed WDM-tSNE outperformed other approaches. After extracting the features from images through the CNN module, we compared the clustering performance of WDM-tSNE with the other five dimension-reduction algorithms. **Figures 5**, **9** proved that WDM-tSNE is not only significantly better than ISOMAP, LLE, PCA, and MDS, but also prior to tSNE.

Recent advances in genomics technologies have greatly accelerated the progress in plant science (Varshney et al., 2021). There are some studies to link phenotypic data to genomic data for discovering the responsible genes or mutations that contributed to plant disease progression (Bolger et al., 2019). Particularly, the systems biology approaches developed by integrating multi-omics data will allow us to identify potential targets and predict new therapeutic strategies (Di Silvestre et al., 2018).

There are several limitations of our current method. Firstly, the features extracted from the plant images by the CNN module are non-semantic, thus, it is hard to interpretable for disease diagnosis and management. Secondly, the current approach only focused on a single disease for each cluster of the image but did not pay attention to the images of plants suffering from multiple diseases. Thirdly, we have not applied the current model on the high-throughput phenotypic images obtained from real natural environments. Finally, we cannot guarantee the clustering performance on the image samples of diseased plants whose severity is manually labeled by different experts.

CONCLUSION

This paper proposes a novel computational framework for plant disease identification and subtype discovery from phenomics data. Our proposed method has achieved high accuracy and good generalization ability in all four public datasets than other deep embedding clustering of images, e.g., t-SNE, ISOMAP, etc.

Specifically, our method does not depend on prior knowledge. Moreover, the size of input images is also unlimited. As a novel embedding strategy, WDM-tSNE provides the perfect clustering performance rather than other methods. The samples in 2D space present great distributions after space embedding, which is significant to reveal the underlying patterns and trajectory of plant disease.

In the future, we will further explore the association between the environmental parameters (climate, hydrology, and soil, etc.) and plant disease evolution.

DATA AVAILABILITY STATEMENT

All the raw images involved in this study can be accessed through the links: https://xf-data-bucket.oss-cn-hangzhou.aliyuncs.com/data.rar. Source code is available at GitHub: https://github.com/JakeJiUThealth/WDM1.0. 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

ZJ, FX, and XX designed the project. FX and XX analyzed the data. ZJ performed the mathematical modeling and optimization. ZJ, FX, ZW, SJ, and KY discussed the results. ZJ and FX wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Science Foundation of Jiangsu Province (no. BK20211210). This work was also supported by the startup award of a new professor at Nanjing Agricultural University (no. 106/804001), and the National Science Foundation of Zhejiang Province (no. LY20F020003).

ACKNOWLEDGMENTS

The authors thank Prof. Haiyan Jiang at the College of Artificial Intelligence at Nanjing Agricultural University to provide valuable discussions.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.789630/full#supplementary-material

Supplementary Figure 1 | The plots for the small-scale balanced datasets are based on t-SNE and WDM-tSNE. WDM-tSNE, Weighted Distance Metric and the t-stochastic neighbor embedding algorithm.

Supplementary Table 1 | Clustering performance on MNIST dataset. MNIST, Modified National Institute of Standards and Technology.

Supplementary Table 2 | A 10-fold cross-validation on the balanced dataset of ARR samples. ARR, Aphanomyces Root Rot.

Supplementary Table 3 | Clustering performance on three small-scale datasets. The sample size of each class is 50.

Supplementary File 1 | The sample assignment for each dataset involved in this study.

REFERENCES

Adhikari, P., Adhikari, T. B., Louws, F. J., and Panthee, D. R. (2020). Advances and challenges in bacterial spot resistance breeding in tomato (Solanum lycopersicum L.). *Int. J. Mol. Sci.* 21:1734. doi: 10.3390/ijms21051734

Afgani, M., Sinanovic, S., and Haas, H. (2008). "Anomaly detection using the Kullback-Leibler divergence metric," in *Proceeding of the First International Symposium on Applied Sciences on Biomedical and Communication Technologies* (IEEE), 1–5.

Albert, C., Cruz, A. L., De Bellis, L., and Ampatzidis, Y. (2017). X-FIDO: an effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Front. Plant Sci.* 8:1741. doi: 10.3389/fpls.2017.01741

Ali, H., Lali, M. I, Nawaz, M. Z., Sharif, M., and Saleem, B. A. (2017). Symptom based automated detection of citrus diseases using color histogram and textural descriptors. *Comput. Electron. Agric.* 138, 92–104. doi: 10.1016/j.compag.2017.04.008

Atila, O., and Sengür, A. (2021). Attention guided 3D CNN-LSTM model for accurate speech based emotion recognition. *Appl. Acoustics* 182:108260. doi: 10.1016/j.apacoust.2021.108260

Ayyub, S. R. N. M., and Manjramkar, A. (2019). "Fruit disease classification and identification using image processing," in *Proceeding of the 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (IEEE), 1–5.

Baldominos, A., Seaz, Y., and Isasi, P. (2019). A survey of handwritten character recognition with MNIST and EMNIST. *Appl. Sci.* 9:3169. doi: 10.3390/app9153169

Barbedo, J. G. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* 180, 96–107. doi: 10.1016/j.biosystemseng.2019.02.002

Bass, D., Stentiford, G. D., Wang, H. C., Koskella, B., and Tyler, C. R. (2019). The pathobiome in animal and plant diseases. *Trends Ecol. Evol.* 34, 996–1008. doi: 10.1016/j.tree.2019.07.012

Bolger, A. M., Poorter, H., Dumschott, K., Bolger, M. E., Arend, D., Osorio, S., et al. (2019). Computational aspects underlying genome to phenome analysis in plants. *Plant J.* 97, 182–198. doi: 10.1111/tpj.14179

Brahimi, M., Arsonovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., and Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. *Hum. Mach. Learn.* 93–117.

Brahimi, M., Boukhalfa, K., and Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Appl. Artificial Intell.* 31, 299–315. doi: 10.1080/08839514.2017.1315516

Chouhan, S. S., Singh, U. P., and Jain, S. (2020). Applications of computer vision in plant pathology: a survey. *Arch. Comput. Methods Eng.* 27, 611–632. doi: 10.1007/s11831-019-09324-0

Bock, C. H., Barbedo, J. G. A., Del Ponte, E. M., Bohnenkamp, D., and Mahlein, A.-K. (2020). From visual estimates to fully automated sensor-based measurements of plant disease severity: status and challenges for improving accuracy. *Phytopathology Res.* 2, 1–30. doi: 10.1186/s42483-020-00049-8

Connor Shorten, T. M. K. (2019). A survey on image data augmentation for deep learning. *J. Big Data* 6, 1–48.

Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Process. Magazine* 29, 141–142. doi: 10.1109/msp.2012.2211477

Dhanvantari, B. N. (1967). The leaf scorch disease of strawberry (Diplocarpon earliana) and the nature of resistance to it. *Can. J. Bot.* 45, 1525–1543. doi: 10.1139/b67-157

Di Silvestre, D., Bergamaschi, A., Bellini, E., and Mauri, P. (2018). Large scale proteomic data and network-based systems biology approaches to explore the plant world. *Proteomes* 6:27. doi: 10.3390/proteomes6020027

Dinh, D. T., Fujinami, T., and Huynh, V.-N. (2019). "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *International Symposium on Knowledge and Systems*, eds J. Chen, V. Huynh, G. N. Nguyen, and X. Tang (Singapore: Springer), 1–17. doi: 10.1007/978-981-15-1209-4_1

Fahrentrapp, J., Ria, F., Geilhausen, M., and Panassiti, B. (2019). Detection of gray mold leaf infections prior to visual symptom appearance using a five-band multispectral sensor. *Front. Plant Sci.* 10:628. doi: 10.3389/fpls.2019.00628

Falk, K., Jubery, Z., Mirnezami, S. V., Parmley, K. A., Sarkar, S., Singh, A., et al. (2020). Computer vision and machine learning enabled soybean root phenotyping pipeline. *Plant Methods* 16, 1–19. doi: 10.1186/s13007-019-0550-5

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

Gadelmawla, E. S. (2004). A vision system for surface roughness characterization using the gray level co-occurrence matrix. *NDT E Int.* 37, 577–588. doi: 10.1016/j.ndteint.2004.03.004

Gaikwad, V. P., and Musande, V. (2017). "Wheat disease detection using image processing," in *Proceeding of the 1st International Conference on Intelligent Systems and Information Management (ICISIM)* (IEEE), 1–3.

Ghosh, S., Mondal, M. J., Sen, S., Chatterjee, S., Roy, N., and Patnaik, S. (2020). "A novel approach to detect and classify fruits using ShuffleNet V2," in *Proceeding of the IEEE Applied Signal Processing Conference (ASPCON)* (IEEE), 1–5.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recogn.* 77, 354–377.

Guan, C., Yuen, K. K. F., and Chen, Q. (2017). "Towards a hybrid approach of k-means and density-based spatial clustering of applications with noise for image segmentation," in *Proceeding of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (IEEE), 396–399.

Gupta, V., Sengar, N., Dutta, M. K., Travieso, C. M., and Alonso, J. B. (2017). "Automated segmentation of powdery mildew disease from cherry leaves using image processing," in *International Conference and Workshop on Bioinspired Intelligence (IWOBI)* (IEEE), 1–4. doi: 10.34133/2020/5839856

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intell. Syst.* 24, 8–12.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1904–1916. doi: 10.1109/tpami.2015.2389824

Hossain, E., Hossain, F., and Rahaman, M. A. (2019). "A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier," in *International Conference on Electrical, Computer and Communication Engineering (ECCE)* (IEEE), 1–6.

Hu, H., Guan, Q., Chen, S., Ji, Z., and Lin, Y. (2020). Detection and recognition for life state of cell cancer using two-stage cascade CNNs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 887–898. doi: 10.1109/TCBB.2017.2780842

Ismail, W., Attique Khan, M., Shah, S. A., and Younus, M. (2020). "An adaptive image processing model of plant disease diagnosis and quantification based on color and texture histogram," in *Proceeding of the 2nd International Conference on Computer and Information Sciences (ICCIS)* (IEEE), 1–6.

José, G. M., Esgario, R. A. K., and Ventura, J. A. (2020). Deep learning for classification and severity estimation of coffee leaf biotic stress. *Comput. Electron. Agric.* 169:105162. doi: 10.1016/j.compag.2019.105162

Kayhan, O. S., and van Gemert, J. C. (2020). "On translation invariance in cnns: convolutional layers can exploit absolute spatial location," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 14274–14285.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks. *NIPS'12*," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Vol. 1, (IEEE), 1097–1105.

Lee, S. H., Goeau, H., Bonnet, P., and Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.* 170:105220. doi: 10.1016/j.compag.2020.105220

Lee, U., Chang, S., Putra, G. A., Kim, H., and Kim, D. H. (2018). An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *PLoS One* 13:e0196615. doi: 10.1371/journal.pone.0196615

Li, D., Quan, C., Song, Z., Li, X., Yu, G., Li, C., et al. (2020). High-throughput plant phenotyping platform (HT3P) as a novel tool for estimating agronomic traits from the lab to the field. *Front. Bioeng. Biotechnol.* 8:623705. doi: 10.3389/fbioe.2020.623705

Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., Sun, W. J. C., et al. (2019). PD2SE-Net: computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron. Agric.* 157, 518–529.

Lin, K., Gong, L., Huang, Y., Liu, C., and Pan, J. (2019). Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Front. Plant Sci.* 10:155. doi: 10.3389/fpls.2019.00155

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

Łukasik, S., Kowalski, P. A., Charytanowicz, M., and Kulczycki, P. (2016). "Clustering using flower pollination algorithm and Calinski-Harabasz index," in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC)* (IEEE), 1–5.

Marzougui, A., Ma, Y., McGee, R. J., Khot, L. R., and Sankaran, S. (2020). Generalized linear model with elastic net regularization and convolutional neural network for evaluating aphanomyces root rot severity in lentil. *Plant Phenomics* 2020:2393062. doi: 10.34133/2020/2393062

Marzougui, A., Ma, Y., Zhang, C., McGee, R. J., Coyne, C. J., Main, D., et al. (2019). Advanced imaging for quantitative evaluation of aphanomyces root rot resistance in lentil. *Front. Plant Sci.* 10:383. doi: 10.3389/fpls.2019.00383

McGee, R. J., Coyne, J. C., Pilet-Nayel, M. L., Moussart, A., Tivoli, B., Baranger, A., et al. (2012). Registration of pea germplasm lines partially resistant to aphanomyces root rot for breeding fresh or freezer pea and dry pea types. *J. Plant Registrations* 6, 203–207. doi: 10.3198/jpr2011.03.0139crg

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

Nikhitha, M., Roopa Sri, S., and Uma Maheswari, B. (2019). "Fruit recognition and grade of disease detection using inception V3 model," in *Proceeding of the 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (IEEE), 1040–1043.

Ozguven, M. M., and Adem, K. (2019). Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Phys. A: Stat. Mechan. Appl.* 535:122537. doi: 10.1016/j.physa.2019.122537

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowledge Data Eng.* 22, 1345–1359.

Pasala, R., and Pandey, B. B. (2020). Plant phenomics: high-throughput technology for accelerating genomics. *J. Biosci.* 45:111.

Pastor-López, I., Sanz, B., Tellaeche, A., Psaila, G., Gaviriade la Puerta, J., and Bringas, P. G. (2021). Quality assessment methodology based on machine learning with small datasets: industrial castings defects. *Neurocomputing* 456, 622–628. doi: 10.1016/j.neucom.2020.08.094

Phadikar, S., and Goswami, J. (2016). "Vegetation indices based segmentation for automatic classification of brown spot and blast diseases of rice," in *Proceeding of the 3rd International Conference on Recent Advances in Information Technology (RAIT)* (IEEE).

Poonam, S. J., and Jadhav, D. S. (2015). Video summarization using higher order color moments (VSUHCM). *Proc. Comput. Sci.* 45, 275–281. doi: 10.1016/j.procs.2015.03.140

Prasad, A., Sharma, N., Hari-Gowthem, G., Muthamilarasan, M., and Prasad, M. (2020). Tomato yellow leaf curl virus: impact, challenges, and management. *Trends Plant Sci.* 25, 897–911. doi: 10.1016/j.tplants.2020.03.015

Qassim, H., Verma, A., and Feinzimer, D. (2018). "Compressed residual-VGG16 CNN model for big data places image recognition," in *Proceeding of the IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE), 1–7.

Rahaman, M. M., Ahsan, M. A., and Chen, M. (2019). Data-mining techniques for image-based plant phenotypic traits identification and classification. *Sci. Rep.* 9:19526. doi: 10.1038/s41598-019-55609-6

Rahman, C. R., Arko, S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., et al. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* 194, 112–120.

Rivas, S., and Thomas, C. M. (2005). Molecular interactions between tomato and the leaf mold pathogen Cladosporium fulvum. *Annu. Rev. Phytopathol.* 43, 395–436. doi: 10.1146/annurev.phyto.43.040204.140224

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65.

Saleem, M. H., Potgieter, J., and Arif, K. M. (2020). Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers. *Plants (Basel)* 9:1319. doi: 10.3390/plants9101319

Shen, N., Li, X., Zheng, S., Zhang, L., Fu, Y., Liu, X., et al. (2019). Automated and accurate quantification of subcutaneous and visceral adipose tissue from magnetic resonance imaging based on machine learning. *Magn. Reson. Imaging* 64, 28–36. doi: 10.1016/j.mri.2019.04.007

Stricker, S. (1994). "The capacity of color histogram indexing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 1–5.

Suarez-Paniagua, V., and Segura-Bedmar, I. (2018). Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinform.* 19(Suppl. 8):209. doi: 10.1186/s12859-018-2195-1

Suzuki, N., Rivero, R. M., Shulaev, V., Blumwald, E., and Mittler, R. (2014). Abiotic and biotic stress combinations. *New Phytol.* 203, 32–43. doi: 10.1111/nph.12797

Syed-Ab-Rahman, S. F., Hesamian, M. H., and Prasad, M. (2021). Citrus disease detection and classification using end-to-end anchor-based deep learning model. *Appl. Intell.* 1–12.

Talwalkar, A., Kumar, S., and Rowley, H. (2008). "Large-scale manifold learning," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 1–8.

Tan, W., Zhao, C., and Wu, H. (2016). Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimedia Tools Appl.* 75, 16741–16761.

Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., and Bennett, M. (2017). Plant phenomics, from sensors to knowledge. *Curr .Biol.* 27, R770–R783. doi: 10.1016/j.cub.2017.05.055

Tete, T. N., and Kamlu, S. (2017). "Plant disease detection using different algorithms," in *Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering* (*RICE*), 103–106.

Too, E., Li, Y., 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.

Tsaftaris, S. A., Minervini, M., and Scharr, H. (2016). Machine learning for plant phenotyping needs image processing. *Trends Plant Sci.* 21, 989–991. doi: 10.1016/j.tplants.2016.10.002

Tusubira, J. F., Akera, B., Nsumba, S., Nakatumba-Nabende, J., and Mwebaze, E. (2020). Scoring root necrosis in cassava using semantic segmentation. *Comput. Vision Pattern Recogn.* 1–10.

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.

Varshney, R. K., Bohra, A., Yu, J., Graner, A., Zhang, Q., and Sorrells, M. E. (2021). Designing future crops: genomics-assisted breeding comes of age. *Trends Plant Sci* 26, 631–649. doi: 10.1016/j.tplants.2021.03.010

Vergani, A. A., and Binaghi, E. (2018). "A soft davies-bouldin separation measure," in *Proceeding of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (IEEE), 1–8.

Verma, S., Chug, A., and Singh, P. (2020). "Impact of hyperparameter tuning on deep learning based estimation of disease severity in grape plan," in *Recent Advances on Soft Computing and Data Mining* (Cham: Springer), 161–171. doi: 10.1007/978-3-030-36056-6_16

Vishnoi, V. K., Kumar, K., and Kumar, B. (2021). Plant disease detection using computational intelligence and image processing. *J. Plant Dis. Protect.* 128, 19–53. doi: 10.1007/s41348-020-00368-0

Wang, G., Sun, Y., and Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Comput. Intell. Neurosci.* 2017:2917536. doi: 10.1155/2017/2917536

Wang, Z., Wang, K., Yang, F., Pan, S., and Han, Y. (2018). Image segmentation of overlapping leaves based on Chan–Vese model and Sobel operator. *Inform. Processing Agric.* 5, 1–10. doi: 10.1016/j.inpa.2017.09.005

Wen, L., Li, X., and Gao, L. (2020). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural. Comput. Appl.* 32, 6111–6124. doi: 10.3934/mbe.2019165

Wenxue Tan, C. Z. H. W. (2020). Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimed. Tools. Appl.* 75, 16741–16761. doi: 10.1007/s11042-015-2940-7

Wilkinson, K., Grant, W. P., Green, L. E., Hunter, S., Jeger, M. J., Lowe, P., et al. (2011). Infectious diseases of animals and plants: an interdisciplinary approach. *Philos. Trans. R Soc. Lond. B Biol. Sci.* 366, 1933–1942. doi: 10.1098/rstb.2010.0415

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629. doi: 10.1007/s13244-018-0639-9

Yu, F., Chou, A., and Ko, K.-I. (2006). On the complexity of finding circumscribed rectangles and squares for a two-dimensional domain. *J. Complexity* 22, 803–817.

Zhang, K., Zhang, L., and Wu, Q. (2019). Identification of cherry leaf disease infected by Podosphaera pannosa via convolutional neural network. *Int. J. Agric. Environ. Inform. Syst.* 10, 98–110. doi: 10.4018/ijaeis.2019040105

Zhang, S., Wu, X., You, Z., and Zhang, L. (2017). Leaf image based cucumber disease recognition using sparse representation classification. *Comput. Electron. Agric.* 134, 135–141. doi: 10.1016/j.compag.2017.01.014

Zhang, X., Qiao, Y., Meng, F., Fan, C., and Zhang, M. J. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access.* 6, 30370–30377. doi: 10.1109/access.2018.2844405

Zhang, Z., Wang, J., and Zha, H. (2011). Adaptive manifold learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 253–265.

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.

ORIGINAL RESEARCH
published: 15 February 2022
doi: 10.3389/fpls.2022.795410

An Improved DeepLab v3+ Deep Learning Network Applied to the Segmentation of Grape Leaf Black Rot Spots

Hongbo Yuan, Jiajun Zhu, Qifan Wang, Man Cheng and Zhenjiang Cai*

College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding, China

The common method for evaluating the extent of grape disease is to classify the disease spots according to the area. The prerequisite for this operation is to accurately segment the disease spots. This paper presents an improved DeepLab v3+ deep learning network for the segmentation of grapevine leaf black rot spots. The ResNet101 network is used as the backbone network of DeepLab v3+, and a channel attention module is inserted into the residual module. Moreover, a feature fusion branch based on a feature pyramid network is added to the DeepLab v3+ encoder, which fuses feature maps of different levels. Test set TS1 from Plant Village and test set TS2 from an orchard field were used for testing to verify the segmentation performance of the method. In the test set TS1, the improved DeepLab v3+ had 0.848, 0.881, and 0.918 on the mean intersection over union (mIOU), recall, and F1-score evaluation indicators, respectively, which was 3.0, 2.3, and 1.7% greater than the original DeepLab v3+. In the test set TS2, the improved DeepLab v3+ improved the evaluation indicators mIOU, recall, and F1-score by 3.3, 2.5, and 1.9%, respectively. The test results show that the improved DeepLab v3+ has better segmentation performance. It is more suitable for the segmentation of grape leaf black rot spots and can be used as an effective tool for grape disease grade assessment.

Keywords: grape black rot, semantic segmentation, DeepLab V3+, channel attention, feature pyramid network

Edited by:
Rujing Wang,
Hefei Institute of Technology Innovation, Hefei Institutes of Physical Science, Chinese Academy of Sciences (CAS), China

Reviewed by:
Andrea Tangherloni,
University of Bergamo, Italy
Abbas Atefi,
California Polytechnic State University, United States

****Correspondence:***
Man Cheng
Chengman1982@163.com

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *15 October 2021*
Accepted: *24 January 2022*
Published: *15 February 2022*

Citation:
Yuan H, Zhu J, Wang Q, Cheng M and Cai Z (2022) An Improved DeepLab v3+ Deep Learning Network Applied to the Segmentation of Grape Leaf Black Rot Spots.
Front. Plant Sci. 13:795410.
doi: 10.3389/fpls.2022.795410

INTRODUCTION

Grapes are one of the most grown economic fruits in the world. Grapes are often used in the production of wine, fermented beverages, and raisins (Kole et al., 2014). In the cultivation of grapes, the larger the area planted, the larger the scale of damage when a disease occurs as well as the greater the economic losses caused. Black rot, which is a fungal disease, is one of the most important grape diseases in the world (Molitor and Berkelmann-Loehnertz, 2011). Black rot spots are black in color and have a small spot area compared to grape leaves. Generally, the assessment of black rot damage on grapes is done by judging the size of the spot on the leaves. This operation is currently performed mainly by hand. However, the manual assessment of spot size and leaf damage area is highly subjective, difficult to quantify, and inefficient. The use of computers and image processing techniques for the identification and segmentation of black rot spots on grapevine leaves can facilitate rapid and accurate assessment of damage for targeted treatment, which is important for ensuring grapevine yield and growers' economic incomes.

The methods of image segmentation have experienced three basic stages from classic segmentation methods, machine learning method, and deep learning method with the development of image processing and computer technology. These methods have been applied in agricultural disease detection. The classical image segmentation, such as threshold segmentation (Mehl et al., 2002; Kim et al., 2005), usually uses color and texture features (Samajpati and Degadwala, 2016) to separate the disease spots from the background. Chaudhary et al. (2012) transformed the RGB image into CIELAB, HIS, and YCbCr color space according to the different color features between the disease spots and leaf, respectively. Then the disease spots were segmented with threshold calculated by the OTSU method based on color features. Ma et al. (2017) achieved segmentation of disease spots from the background by fusion features of the super red index, the H-component of HSV, and the b-component of color space for the greenhouse vegetable images with 97% accuracy. Jothiaruna et al. (2019) proposed a method that integrated color features and region growing for the segmentation of leaves disease spots with an average segmentation accuracy of 87%. Sinha and Shekhawat (2020) segmented peacock disease spots on olive leaves according to the different textures of the leaves and spots, and the purpose of disease detection was realized. The classical image segmentation methods require high image quality, and the recognition result will be poor or even invalid if the environmental conditions changed when the image acquiring. Therefore, the generality and robustness of those methods are unsatisfactory, and the accuracy in practical application is not guaranteed.

With the development of machine learning, many researchers began to try to apply it to disease spots segmentation to improve the accuracy and robustness of segmentation. Zhou et al. (2014) inputted the color histogram of the image into the support vector machine (SVM) model to segment the Cercospora disease spots for sugar beet, and the average accuracy, recall, and *F* value were more than 0.87. Bai et al. (2017) used a fuzzy C-means algorithm for segmentation of cucumber leaves spots disease in complex backgrounds, and the experimental results showed that the average error did not exceed 0.12%. Pan et al. (2019) segmented pear blackspot disease in hyperspectral images using SVM with an overall accuracy of 97.5%. Singh (2019) applied a particle swarm optimization algorithm for the segmentation of downy mildew spots in sunflower leaves with an average accuracy of 98%. Appeltans et al. (2021) removed soil pixels from hyperspectral images by linear discriminant analysis classification and used a logistic regression supervised machine learning classifier for pixel classification of leek leaves to segment the spots of leek white tip disease with an accuracy of 96.74%. Machine learning methods can achieve satisfactory segmentation results using small sample size, but these methods require multiple steps of image preprocessing and are relatively complex to execute. In addition, the machine learning-based segmentation methods are relatively weakly adapted to unstructured environments and need researchers to manually design feature extraction and classifiers, which makes the work more difficult.

With the improvement of computer hardware performance, deep learning has been developed rapidly (Lecun et al., 2015). Common deep learning algorithms are full convolutional neural network algorithm (FCN; Long et al., 2015), DeepLab (Chen et al., 2017), U-Net (Ronneberger et al., 2015), V-Net (Milletari et al., 2016), USE-Net (Rundo et al., 2019), SegNet (Badrinarayanan et al., 2017), etc. Lin et al. (2019) designed a semantic segmentation model based on convolutional neural network (CNN) for pixel-level segmentation of cucumber leaves powdery mildew disease spots, which provided a valuable tool for cucumber breeders to assess the severity of powdery mildew. Jiang et al. (2020) combined deep learning and SVM to segment the leaves disease images of four rice species with an accuracy of 96.8%. Wang et al. (2021) used DeepLab v3+ and U-Net methods to segment disease spots from cucumber leaves, and calculate their damage levels with an average accuracy of 92.85%. Lin et al. (2019) constructed a U-Net-based semantic segmentation model for cucumber powdery mildew spots segmentation with an average accuracy of 96.08%. Wspanialy and Moussa (2020) used U-Net neural network for segmentation of tomato leaves and spots in leaves with an average accuracy of 98% and then assessed the disease hazard level. Hu et al. (2021) segmented tea leaves and disease spots using a CNN and assessed the damage level. Liang et al. (2019) used PD^{2}SE-Net neural network to segment plant disease spots areas and assessed their damage levels with an overall accuracy of more than 91%. The deep learning approach has all the work done by the CNN, which does not require too much pre-processing process or artificial selection of potential features compared to classical image processing methods and machine learning methods. The deep learning approach not only reduces the difficulty of plant leaves spots segmentation but also has higher accuracy and robustness.

Our group has developed a method to improve the recognition accuracy for grape leaf black rot by combine image enhancement technology and a deep learning network (Zhu et al., 2021). It can recognize the disease spots and calculate the number, but cannot segment the disease spots from the background. To realize the spot segmentation of grape leaf black rot, this paper designs a CNN based on an improved DeepLab v3+.

MATERIALS AND METHODS

Dataset and Test Environment Setup

The open dataset Plant Village (Hughes and Salathe, 2016) was used to perform experiments in this work, which provides symptoms of 26 common diseases on leaves of 14 plant species with a total of 54,309 RGB images. We selected 1,180 images of grape leaves infected with black rot as test subjects, and all these images were confirmed by researchers studying grape diseases. The selected images were taken in an indoor environment with a uniform gray background, and each image included only one frontal view of a grape leaf with 256 × 256 pixels. The areas of disease spots were manually labeled by

LabelMe (Russell et al., 2008) software. The average number of diseases present in an image was around 15, with more than 17,000 segmentation targets present in total. Before the experimental training, 1,180 data images were divided into training and test sets, and 1,072 images were selected for training the network and 108 images were selected as the test set for evaluating the network, which was named TS1. Furthermore, to increase the credibility of the model, a large number of images of grape leaves with disease spots from orchard sites were collected *via* the Internet. A total of 108 images of grape leaves with black rot spots in natural environments were selected by researchers studying grape diseases for an extra test set, which was named TS2. During the process of network training, the training set was divided into two parts in the form of training and validation data. The division ratio of training and validation data was 9:1. The training data were used for model fitting, and the validation data were used to adjust the super parameters of the model and to preliminarily evaluate the ability of the model. The test set was used to evaluate the generalization ability of the final model. In this study, the number of epochs was 120, the input batch was four, the learning rate was 0.001, and the size of the input image was 512 × 512. The VOC 2007 dataset format was used for the dataset. The experiments were conducted on Windows 10 with the Pytorch deep learning framework. The test computer contained an 8 GB GPU GeForce GTX 1070Ti and an AMD Ryzen 51600X Six-Core processor. Python language was used for programming.

Segmentation Method of Grape Leaf Black Rot Spots

To improve the segmentation performance of grapevine leaf black rot spots, a deep learning network based on the DeepLab v3+ was constructed. It is the third version of DeepLab, with high segmentation effectiveness and speed. In the improved DeepLab v3+ network constructed in this paper, the residual part in the backbone network ResNet101 incorporates a plug-and-play attention mechanism module. This improves the performance of various CNNs without increasing the complexity of the model. Moreover, a feature fusion branch based on a feature pyramid network (FPN) was added to the DeepLab v3+ encoder, which performs feature fusion on high-resolution and low-resolution feature maps. Finally, in the improved DeepLab v3+, one 4-fold up-sampling is replaced with two 2-fold up-sampling. Furthermore, the continuity of pixels in the obtained images is stronger and the network segmentation effect is improved.

Channel Attention Module

The efficient channel attention (ECA; Wang et al., 2020) module is a local cross-channel interaction strategy without dimensionality reduction, which can be efficiently implemented *via* one-dimensional (1D) convolution. The ECA module is obtained by improving on Squeeze-and-Excitation (SE; Hu et al., 2020), which is an effective channel attention learning method. It predicts a weight to be weighted for each output channel. The SE method first uses global average pooling (GAP) for each feature channel individually to reduce the two-dimensional feature channel to a real number. Then, two fully-connected layers capture the non-linear cross-channel interaction. Finally, a Sigmoid function generates the channel weights with a value between 0 and 1. This weight is added to the feature channel as a weight to generate the next level of input data. The characteristic of SE is to use the correlation between channels instead of the correlation in the spatial distribution. By controlling the magnitude of the weight, the important features are enhanced and the unimportant features are weakened so that the extracted features are more directional. Compared with SE, the improvement of ECA is that the GAP operation of feature channels does not reduce the dimensionality. Instead, it captures local cross-channel interaction information by considering each channel and its K nearest neighbors. The ECA module can be used as a very lightweight plug-and-play module to improve the performance of various CNNs (Gao et al., 2020; Wang et al., 2020). Its implementation process is shown in **Figure 1**. The blue part uses GAP to aggregate convolutional features without performing dimensionality reduction operations. The ECA module can be efficiently implemented *via* a 1D convolution of size *k*, where the size of the convolution kernel *k* represents the coverage of local cross-channel interaction, that is, how many neighbors near the channel participate in the attention prediction of this channel. Wang et al. (2020) studied the *k* value of the CNN network with ResNet-101 as the backbone, and the *k* of the ECA module was set to 3, 5, 7, and 9 for training. The accuracy value was used to evaluate the effect of *k*. The experimental results showed that the accuracy was 78.47%, 78.58%, 78.0%, and 78.57% corresponding to the *k* value of 3, 5, 7, and 9, respectively. Therefore, *k* was set to 5 in this paper. The yellow part is the result of implementation *via* 1D convolution, and then the Sigmoid function can be used to generate the channel weights to obtain the normalized weights between 0 and 1. Finally, the original feature image X, whose matrix size is H × W × C, is multiplied by the weight generated by the Sigmoid function to obtain a new feature image X′, and the matrix size is H × W × C.

In this method, the backbone network of DeepLab v3+ is constructed using ResNet101, and an ECA module is inserted into the residual (Bottleneck; He et al., 2016) module of ResNet101. This method can realize the adaptive adjustment of the convolution kernel size in the channel of each residual block. The purpose is to improve the segmentation effect of the model. **Figure 2** shows a schematic diagram of the insertion of ECA in the residual module of ResNet101.

Feature Fusion Branching Based on a FPN

In the process of learning image features by CNNs, the resolution of the image is gradually reduced due to the deep convolution operation, resulting in low-resolution deep features at the output. In this way, there will be recognition errors for objects with a relatively small proportion of pixels in the image. The accuracy of multi-scale detection can be improved if the features at different levels of the network training process can be combined. An FPN (Lin et al., 2017) is a method that can fuse the feature maps of different layers. Feature maps that can reflect

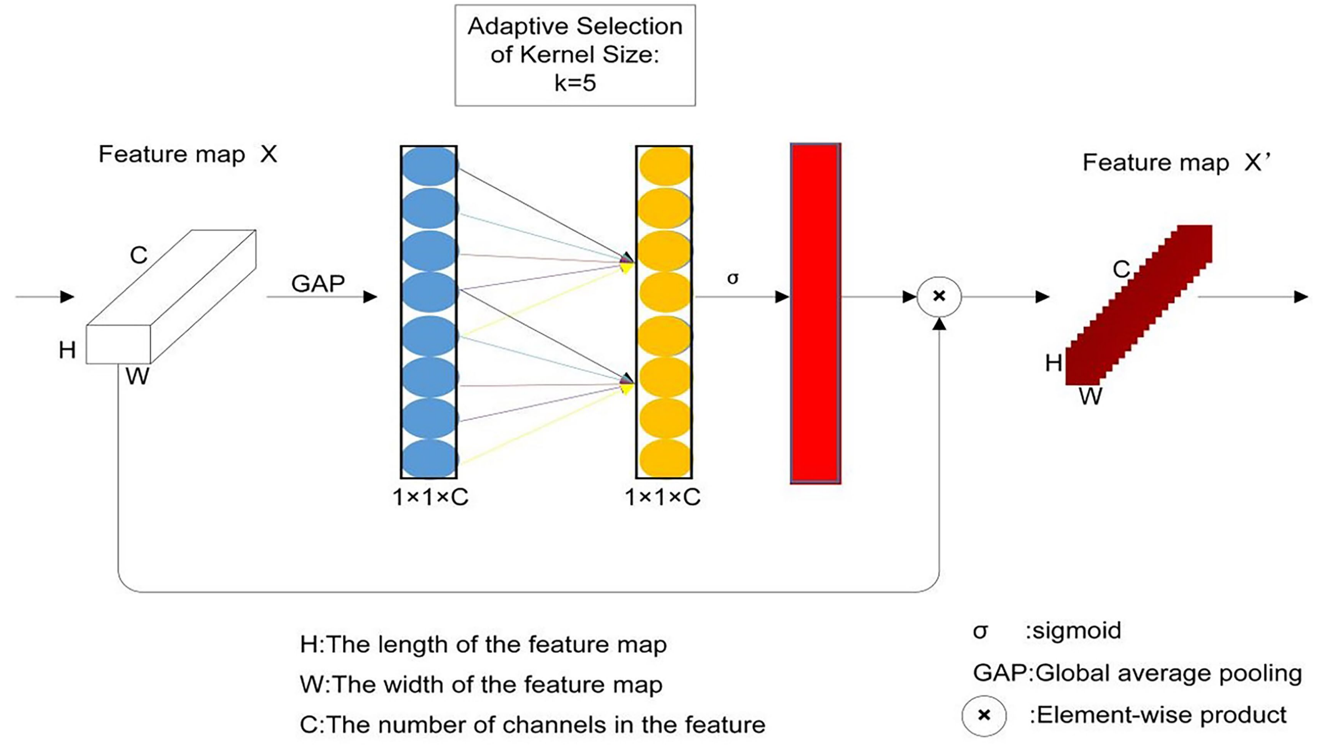

Adaptive Selection of Kernel Size: k=5
Feature map X
C
H
W
GAP
1×1×C
1×1×C
σ
×
Feature map X'
C
H
W
H:The length of the feature map
W:The width of the feature map
C:The number of channels in the feature
σ :sigmoid
GAP:Global average pooling
× :Element-wise product

FIGURE 1 | Efficient channel attention module.

semantic information at different scales can be obtained through the fusion of FPNs. The feature fusion process of feature pyramids is shown in **Figure 3**. As shown, the left side is the feature maps of three different layers, whose resolutions become smaller from the bottom to the top. The middle part is the FPN, which can up-sample the deep-level features to convert them to the size of the shallow-level feature map and then fuses them with the shallow-level features. The right side is the feature map obtained after the FPN, which contains not only the deep level features but also the features of different levels. Here, the feature maps generated by Block3 and Block2 in the backbone network ResNet101 of DeepLab v3+ were fused. The feature map sizes of Block3 and Block2 were 1/16 and 1/8, and the number of channels was 1,024 and 512. In the FPN, the feature maps in Block3 and Block2 were subjected to 1×1 convolutional dimension reduction. The number of feature map channels in Block3 was changed from 1,024 to 256, and the number of feature map channels in Block2 was changed from 512 to 256. Then, the feature map of Block3 was up-sampled by a factor of 2 to change the size of the feature map from 1/16 to 1/8. Finally, the feature maps of Block3 and Block2 were combined to obtain the fused feature maps. The fused feature map has richer semantic and spatial information because it contains features from both levels, which can improve the segmentation effect of DeepLab v3+ network.

Improved DeepLab v3+ Network Structure

The improved DeepLab v3+ network consists of two parts, an encoder and decoder (Chen et al., 2018), which shows in **Figure 4**. The encoder part trains the network, progressively obtains the feature maps, and captures higher-level semantic information. The decoder part semantically projects the features learned by the encoder part into the pixel space to achieve pixel segmentation. In the encoder, the backbone network is constructed using ResNet101 and the ECA module is inserted in its residual module. Moreover, to enhance the semantic information of the feature map, the feature maps of Block2 and Block3 of the ResNet101 network are fused. Atrous Spatial Pyramid Pooling (ASPP; Chen et al., 2018) is connected behind the ResNet101 backbone network. Dilated convolution with different sampling rates can be sampled in parallel by ASPP, which is equivalent to capturing the context of images at multiple scales. Dilated convolution (Yu et al., 2017) adds atrous to the convolution map during the convolution operation to expand the reception field so that each convolution output can contain a larger range of information. In addition to the convolution kernel, the dilated convolution also has a hyper-parameter dilation rate. It refers to the number of intervals between the convolution kernel during convolution mapping, that is, the number of atrous inserted. **Figure 5** shows the execution process of convolution. Here, **Figure 5A** is the standard convolution process and **Figure 5B** is the process of dilated convolution.

The encoder module has three outputs. The first is the low-level feature (LF) output by Block1 in the backbone network. The second is the fusion feature (FF) of Block2 and Block3 output by the FPN. The last one is the high-level

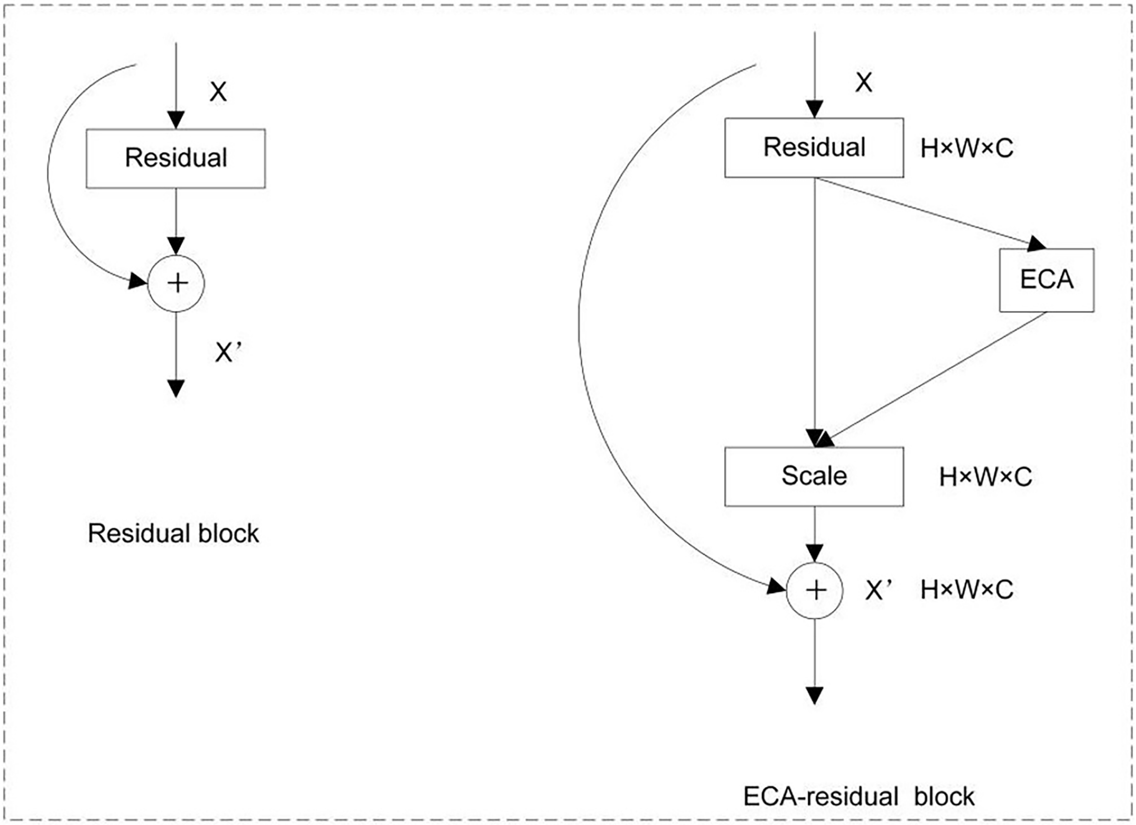


FIGURE 2 | Application of the ECA module in residuals.

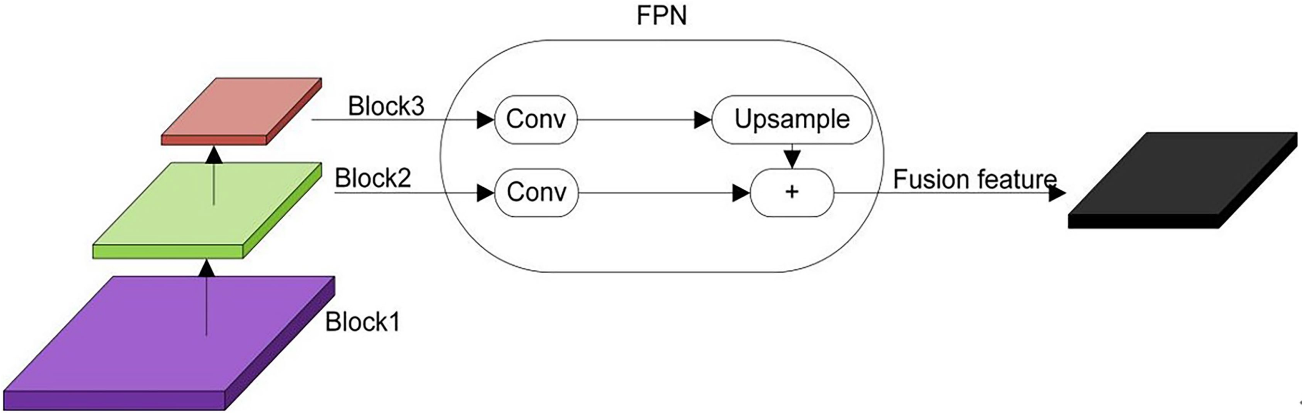


FIGURE 3 | Feature pyramid execution process.

feature (HF) output by the ASPP module after 1 × 1 convolution. High-level feature output concatenates to FF after it has undergone 2-fold up-sampling, and then the second 2-fold up-sampling is performed. The result of this operation is concatenated to the LF, which has been convoluted by 1 × 1 convolution. A 3 × 3 convolution is performed after the above operation, and then a single four-fold up-sampling is performed. Then, the dense classification of pixels is obtained, which is image segmentation.

Parameters Setting of Improved DeepLab v3+ Network

The stochastic gradient descent method was applied to the end-to-end training of the deep learning network, and the

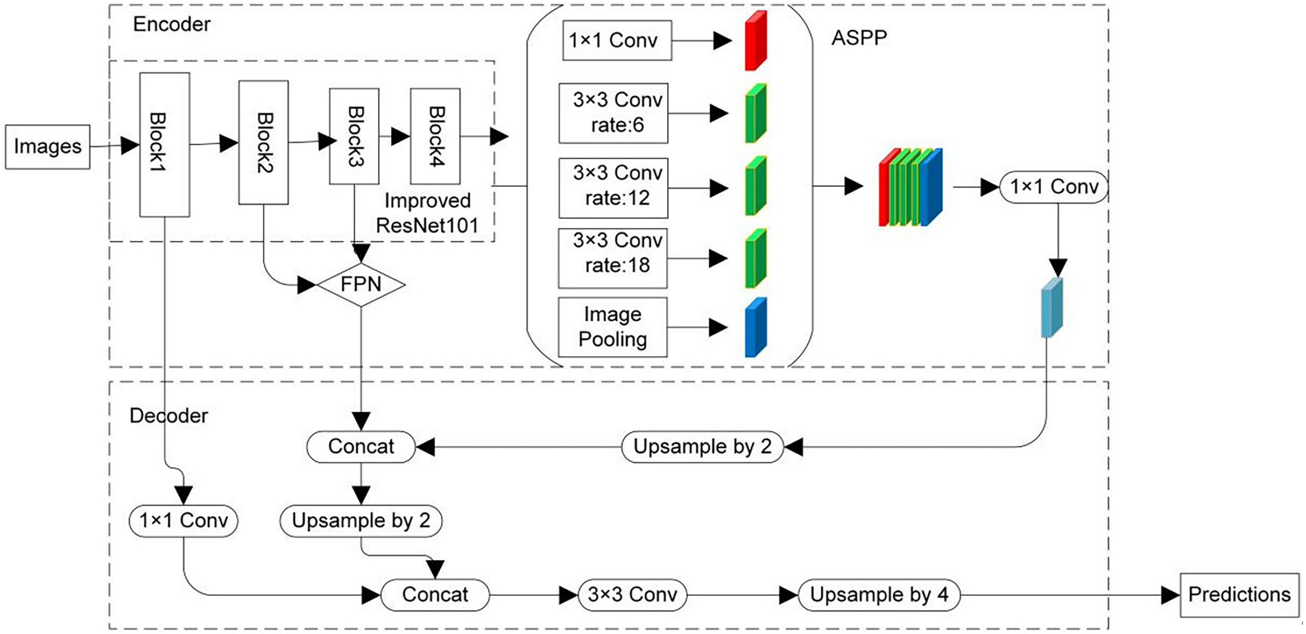


FIGURE 4 | Improved DeepLab v3+ network structure.

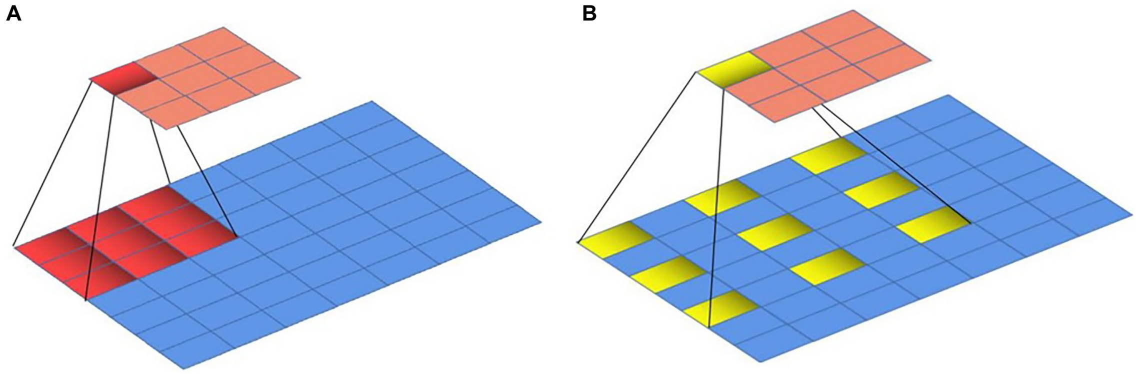


FIGURE 5 | Convolution execution process. **(A)** Standard convolution work process, **(B)** The dilated convolution work process.

loss function was set to Dice_Loss as shown in Equation (1). The weight decay rate was set to 0.001, and the kinetic energy factor was set to 0.8. The initial learning rate was set to 0.001, the learning rate decay mode was exponential decay, and the Batch_size was set to 4. The maximum iteration period (Epochs) was set to 120, and the network input size was set to 512×512. The data set was stored in the format of the VOC 2007 data set, and pre-trained model weights were loaded in the experiment to speed up the convergence of the model.

$$Dice\_Loss = \frac{FP + FN}{FP + 2TP + FN} \tag{1}$$

where TP represents the true positives, indicating that the black rot area of grape leaves automatically segmented by the model overlaps with the real disease area; FP represents the false positives, indicating that the model misidentified the background area as a black rot spot area and segmented it; TN represents the true negatives, indicating that the model identified the real background area as the background area; and FN represents the false negatives, indicating that the model misidentified the real black rot area as the background area.

Evaluation Indicators

In this study, to evaluate the performance of the improved DeepLab v3+ network segmentation, the mean intersection over union (mIOU), the dice coefficient (Dice), the pixel accuracy (ACC), precision (*P*), recall (*R*), and *F1*-score were selected as evaluation metrics.

The mIOU is a common evaluation metric in semantic segmentation methods. In semantic segmentation, the predicted and true regions are obtained by pixel operation, and Equation (2) is as follows:

$$\mathrm{mIOU}=\frac{1}{2}\sum_{i=0}^{1}\frac{\mathrm{p_{ii}}}{\sum_{j=0}^{1}\mathrm{p}_{ij}+\sum_{j=0}^{1}\mathrm{p}_{ji}-\mathrm{p}_{ii}} \tag{2}$$

where p_{ij} denotes the number of pixels that originally belonged to class *i* but are predicted to be class *j*, p_{ii} denotes the number of pixels whose true label is class *i* predicted to be class *i*, and p_{ji} denotes the number of pixels that originally belonged to class *j* but are predicted to be class *i*. In this study, the pixels in each image were classified into two classes: black rot spots and background.

The Dice value is usually used to calculate the similarity of two samples, and the value range is (0,1). A Dice value close to 1 indicates a high set similarity, that is, the target is better segmented from the background; while a Dice value close to 0 indicates that the target cannot be effectively segmented from the background. The dice value equation is as follows:

$$Dice=\frac{2TP}{FP+2TP+FN} \tag{3}$$

The ACC is the ratio of the number of correctly predicted pixels to the total number of pixels in the category, and its equation is as follows:

$$ACC=\frac{TP+TN}{TP+FN+FP+TN} \tag{4}$$

The *P*, *R*, and *F1*-score were calculated by the following equation:

$$\begin{cases} P=\dfrac{TP}{TP+FP} \\ R=\dfrac{TP}{TP+FN} \\ F1-score=2\times\dfrac{P\cdot R}{P+R} \end{cases} \tag{5}$$

Comparison of the Effects of Different Improvements of DeepLab v3+

To verify the effectiveness of the neural network constructed in this paper for grape leaf spot segmentation, eight sets of comparison experiments with different improvements were designed. These eight different improvements were named from Imp1 to Imp8, as shown in **Table 1**. In Imp1, the three dilated convolutions of the ASPP model of the original DeepLab v3+ network were modified to four dilated convolutions, and their dilated rate sizes were 4, 8, 12, and 16, respectively. Theoretically, the increase of dilated convolutions and the change of dilated rate sizes will improve the fusion effect of semantic features. In Imp2, the ResNet 101, backbone of the DeepLab v3+, was replaced with Wide ResNet (Zagoruyko and Komodakis, 2016), which can improve the network segmentation performance by changing the width of the network without changing the network depth. The residual module of the backbone ResNet101 was inserted into the ECA module in Imp3, and the ECA model can adaptively adjust the convolutional kernel size in each channel of the residual block, which can improve the segmentation effect of the network. The coding side of the DeepLab v3+ network was added with a feature fusion branch based on the FPN in Imp4. The FPN can fuse different levels of feature maps and can obtain feature maps that can reflect semantic information at different scales. In imp5, the ASPP part of DeepLab v3+ was combined with DenseNet (Yang et al., 2018) to form DenseASPP, and the new module had a larger receiver field and more densely sampled points. Imp1, Imp3, and Imp4 were combined as Imp6. Imp3 and Imp5 were combined as Imp7. Imp3 and Imp4 were combined as Imp8, which is the improvement method used in this paper.

RESULTS

The Segmentation Results of Improved DeepLab v3+ for Grape Leaves Black Rot

The training dataset with annotation information was fed into the improved DeepLab v3+ network for training. The network was trained for 120 epochs, which required around 8.3 h. During the training process, the training model was saved once every 1 epoch, and a total of 120 completed models were saved. The convergence of the model can be reflected by the loss values generated during the training process. **Figure 6** shows the changes in the loss values of the training data and validation data in the training set during the training process. The training loss and validation loss gradually converged to stability during the training process, and the final training loss and validation loss values stabilized at 0.132.

TABLE 1 | Different DeepLab v3+ improvement methods.

Improvement methods	Improvement content
Imp1	Modify the three dilated convolutions of ASPP in the original network to four dilated convolutions with a dilated rate size of 4, 8, 12, and 16, respectively
Imp2	Replace the ResNet backbone in the original network with wider ResNet
Imp3	Insert the ECA module in the residual module of the backbone ResNet101
Imp4	A feature fusion branch based on an FPN is added to the coding side of the original network
Imp5	Combine the ASPP part of the original network with DenseNet to form DenseASPP
Imp6	Imp1 + Imp3 + Imp4
Imp7	Imp3 + Imp5
Imp8	Imp3 + Imp4

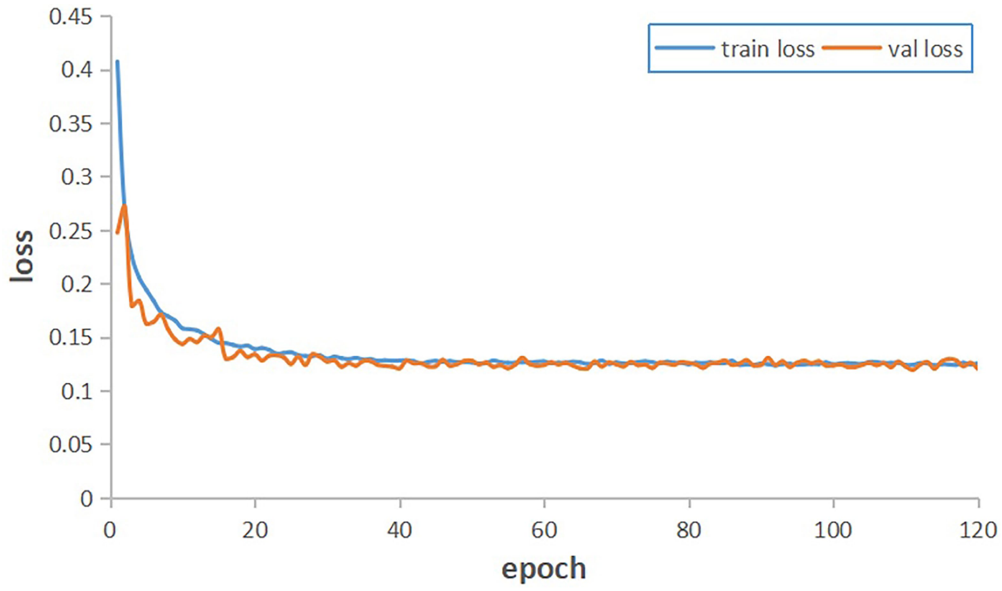


FIGURE 6 | Improved DeepLab v3+ training results.

TABLE 2 | Statistics of the segmentation results of the test set TS1 by the before and after improved DeepLab v3+.

Algorithm	Evaluation indicators					
	mIOU	ACC	Dice	*P*	*R*	*F1*-score
DeepLab v3+	0.823	0.984	0.903	0.949	0.861	0.903
DeepLab v3+ (improved)	0.848	0.987	0.918	0.957	0.881	0.918

To verify the performance of the model, the optimal model at the end of training was selected to be used for segmentation trials on test set TS1. The statistical results of the experiment before and after improved DeepLab v3+ are shown in **Table 2**. As can be seen from **Table 2**, the improved DeepLab v3+ outperforms the pre-improvement DeepLab v3+ in all evaluation metrics. In particular, it improved 3.0, 2.3, and 1.7% in mIOU, *R*, and *F1*-score, respectively. The effects of the segmentation are shown in **Figure 7**.

Figure 8 shows the segmentation results of DeepLab v3+ before and after improvement applied to black rot spots of grape leaves in test set TS1. **Figure 8A** shows the original image, **Figure 8B** shows the manually labeled and segmented image, **Figure 8C** shows the segmentation results of DeepLab v3+ before improvement, and **Figure 8D** shows the segmentation results of DeepLab v3+ after improvement. The blue markers in **Figure 8** indicate the small spots targeted in the original image that were not identified and segmented by the original network model but were correctly segmented by the improved network model. The yellow markers indicate that the semantic segmentation network correctly identified and segmented some small spots in the original image even though they were not manually labeled and segmented due to human oversight. This also demonstrates that the use of deep learning methods can reduce subjective errors caused by manual segmentation. The red markers indicate that the leaf edges were misidentified as spots and segmented by the network model due to shadows. This indicates that there is a requirement for background conditions for disease spot recognition using deep learning. Furthermore, **Figure 8** shows that although the improved network model could segment the spots at the same location, the improved network model was more accurate and the segmented spots overlapped more with the actual spots.

Experiments with the Plant Village dataset demonstrated that the improved DeepLab v3+, which incorporates an attention mechanisms and feature pyramids, could improve the segmentation of black rot spots on grape leaves. An additional dataset, TS2, with 108 images from photos taken in different orchard fields was used for testing to verify the effectiveness of the method in an orchard field setting. The TS2 dataset was tested experimentally using the DeepLab v3+ network before and after the improvement. **Figure 9** shows the experimental results of the DeepLab v3+ algorithm before and after the improvement on TS2. **Figure 9A** is the original image, **Figure 9B** is the unimproved DeepLab

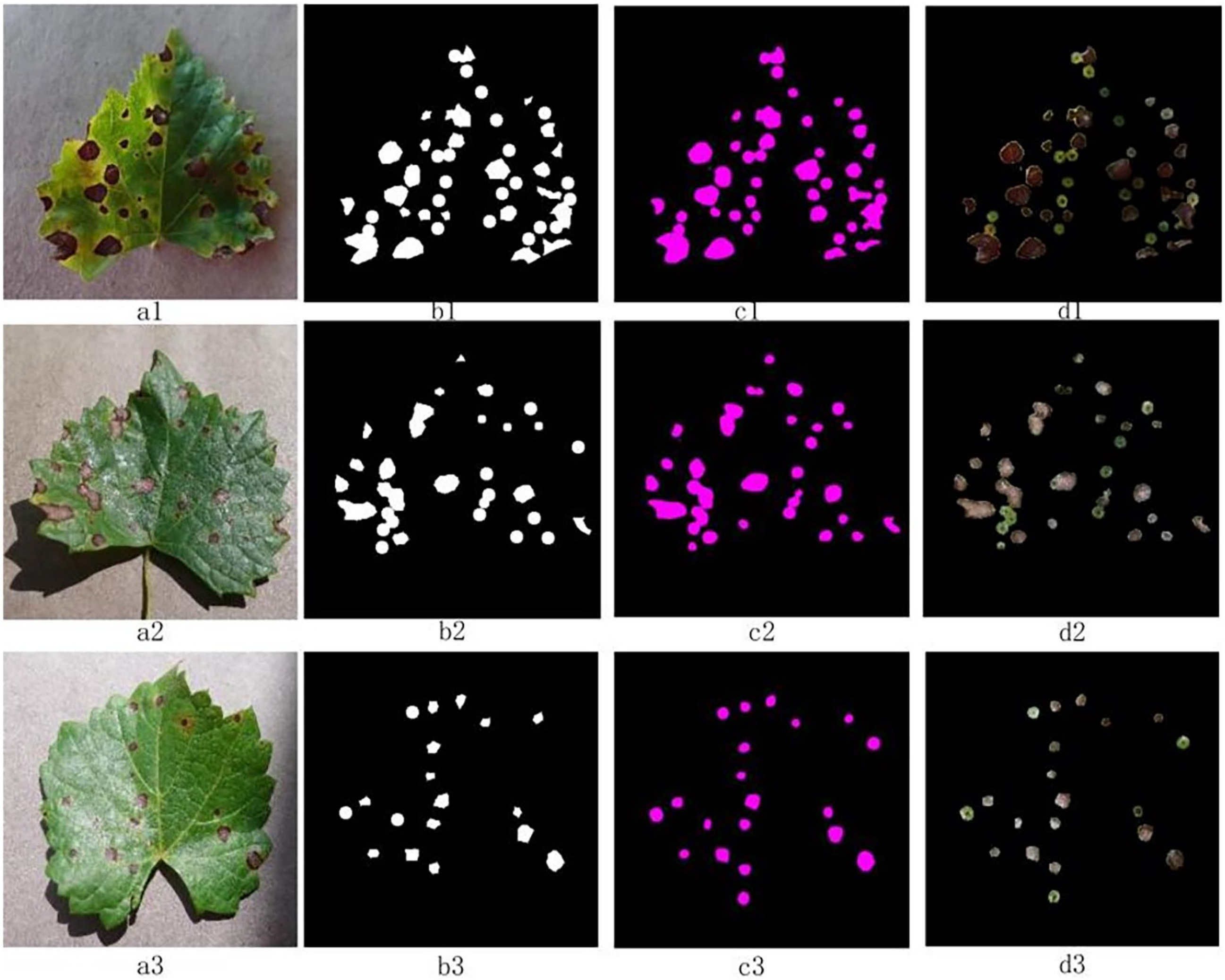


FIGURE 7 | Segmentation effects of the improved DeepLab v3+ on the test set TS1 image. The "a" column is the original image, the "b" column is the labeled mask, the "c" column is the segmentation result of the model, and the "d" column is the disease spot extraction result.

v3+ segmentation result, and **Figure 9C** is the improved DeepLab v3+ segmentation result. To show the network segmentation effect before and after the improvement, different colors are marked in **Figures 9B,C**. The yellow markers show that the improved network was more comprehensive in terms of the segmentation effect. The red markers show that the improved network was more accurate in segmentation. The blue markers show that the improved network was less affected by the background under the interference of complex background. The experimental results show that the improved DeepLab v3+ network performed better than the unimproved DeepLab v3+ network. Moreover, comparing the experimental segmentation effects shows that the improved DeepLab v3+ network can be applied to an actual orchard situation.

The statistical results of DeepLab v3+ before and after the improvement are shown in **Table 3** for test set TS2. **Table 3** shows that the improved DeepLab v3+ did not segment as well as TS1 for grape leaf black rot spots in a natural environment. This is because the images in TS1 were indoor environments, and the grape leaves were tiled with a single and simple background. In contrast, there were negative effects, such as overlapping leaves, gaps formed by shading, and lighting in the orchard field environment, which caused interference for accurate segmentation. Moreover, for large and dense spot areas, the network model would segment the dense spot areas as a whole; thus incorrectly classifying some backgrounds as spot areas. However, segmentation using the improved DeepLab v3+ still outperformed the one before the improvement, especially reaching scores of 0.756, 0.734, and 0.805 in mIOU, *R*, and *F1*-score, respectively, which were 3.3, 2.5, and 1.9% higher than those before improvement. This indicates that the proposed method improves the segmentation performance

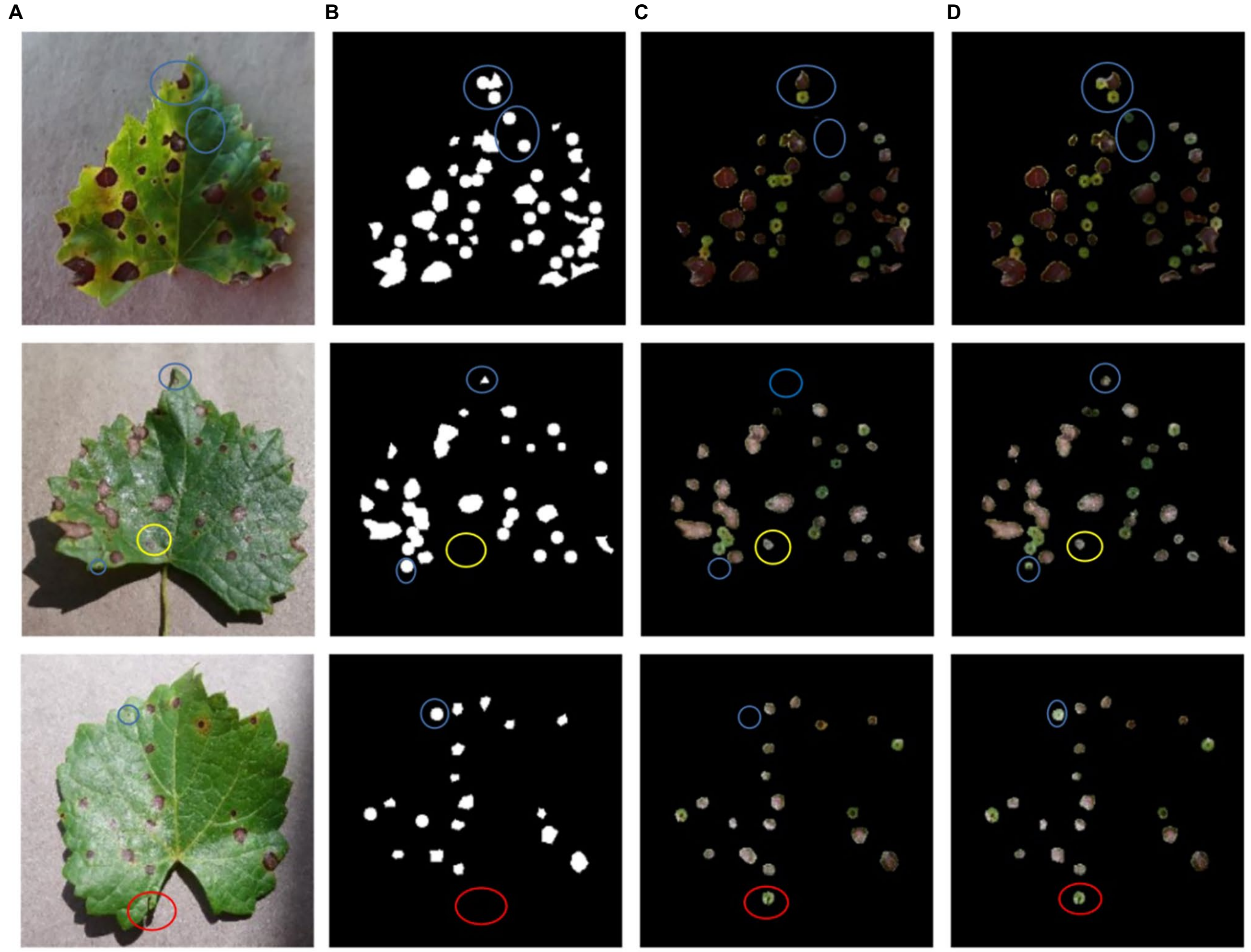


FIGURE 8 | A comparison of network training results before and after DeepLab v3+ improvement. **(A)** The original image, **(B)** the manually labeled and segmented image, **(C)** the DeepLab v3+ segmentation results, **(D)** the improved DeepLab v3+ segmentation results.

of DeepLab v3+, and its ubiquity and adaptability for application in a real environment are better compared with the unimproved network model.

Comparison of the Effects of Different Improvements of DeepLab v3+

For the above eight DeepLab v3+ improvement methods, the same training set was used for training, and the performances were tested with the test set TS1. To compare the results of different improvement methods, the parameters of the network, such as the learning rate, epoch, and batch size, were kept consistent during the experiments. The test results are shown in **Table 4**, where the four parameters mIOU, ACC, Dice, *P*, *R*, *F1*-score, and Pt are used for comparison. The Pt is the storage space occupied by the weight file generated after network training. **Table 4** shows that the performance indicators of the unimproved DeepLab v3+ on the test set TS1 were 0.823, 0.984, and 0.811 for mIOU, ACC, and Dice, respectively. **Table 5** shows that, compared with the DeepLab v3+ network before improvement, the scores of mIOU, ACC, and Dice were higher for the other six of the eight improved methods, except for Imp1 and Imp2. Compared with the DeepLab v3+ before improvement, Imp3 and Imp4 were 1.6% and 1.3% higher in mIOU and 0.5% and 1.3% higher in Dice, respectively. This indicates that fusing ECA or adding FPN in DeepLab v3+ network could improve the segmentation performance of the model. Although the improved method of Imp5 had improved mIOU and Dice by 1.4% and 1%, respectively. The Pt generated by this method required more memory space than that of Imp3 and Imp4. Moreover, Imp6 is a fusion of Imp1, Imp2, and Imp3, but its mIOU and Dice were lower than Imp3 and Imp4. This shows that the additional change of the dilated rate of the dilated convolution did not improve the performance of the network, which was consistent with the test results of Imp1. Besides, Imp7 is a fusion of Imp3 and Imp5, because fusing ECA in Imp3 alone or modifying ASPP to DenseASPP in Imp5 alone could improve network

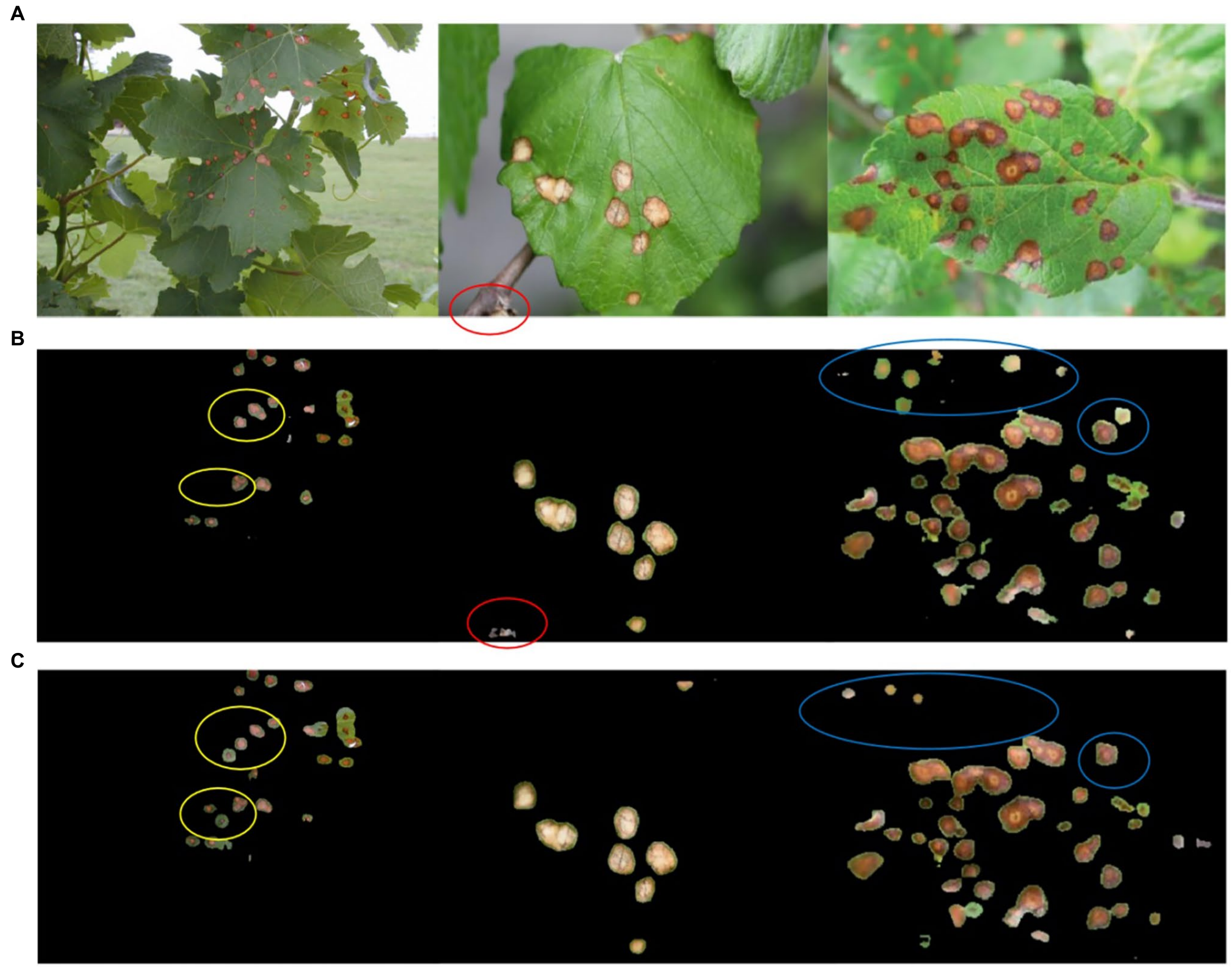


FIGURE 9 | A comparison of segmentation results of test set TS2 images before and after improvement of DeepLab v3+. **(A)** The original figure, **(B)** the segmentation results of DeepLab v3+ without improvement, **(C)** the segmentation results of the improved DeepLab v3+.

TABLE 3 | Statistics of the segmentation results of test set TS2 images before and after DeepLab v3+ improvement.

Algorithm	**Evaluation indicators**					
	mIOU	**ACC**	**Dice**	***P***	***R***	***F1*-score**
DeepLab v3+	0.732	0.874	0.845	0.916	0.785	0.845
DeepLab v3+ (improved)	0.756	0.889	0.861	0.925	0.805	0.861

performance. Thus, Imp7 scored higher in mIOU than Imp3 and Imp5, and the Dice value was in line with Imp5 and higher than Imp3. However, the introduction of DenseASPP led to a larger computation within the network and its obtained weight file was relatively large, which was consistent with the performance of Imp5. The final improved method adopted in this paper was Imp8, which fuses Imp3 and Imp4 and adds both ECA and FPN in the DeepLab v3+ network. Here, Imp8 scored 0.848, 0.987, 0.918, 0.957, 0.881, and 0.918 for mIOU, ACC, Dice, P, *R*, and *F1*-score, respectively, after the same test set test, and it received the highest scores among all eight methods. Moreover, its weight file occupied 241,553 kb of space, which was in the middle level among the eight improved methods. This indicates that the Imp8 method used in this paper has a better overall performance compared to other improvement methods.

A comparison of the training performance of the unimproved DeepLab v3+ and the improved network using

TABLE 4 | Comparison of the test results of different improvement methods of DeepLab v3+.

Type	Evaluation indicators						
	mIOU	**ACC**	**Dice**	***P***	***R***	***F1*-score**	**Pt (kb)**
Imp1	0.812	0.982	0.896	0.945	0.852	0.896	572,794
Imp2	0.818	0.982	0.900	0.947	0.857	0.900	554,101
Imp3	0.839	0.985	0.912	0.954	0.874	0.912	232,841
Imp4	0.836	0.987	0.911	0.953	0.872	0.911	241,541
Imp5	0.837	0.986	0.911	0.954	0.873	0.911	310,161
Imp6	0.833	0.986	0.909	0.952	0.869	0.909	241,533
Imp7	0.841	0.986	0.914	0.955	0.876	0.914	310,173
Imp8	0.848	0.987	0.918	0.957	0.881	0.918	241,553

TABLE 5 | Detection statistics results of the two methods for the grape leaves in **Figure 11**.

Leaf	Number of real disease spots			Pixels of real disease spots		
	Actual number	**Detected by the detection method**	**Detected by the segmentation method**	**Actual pixels**	**Segmented by the detection method**	**Segmented by the segmentation method**
Left	18	16	18	2,301	/	2,237
Middle	17	10	16	2,328	/	2,228
Right	14	12	12	2,132	/	2,066

The disease spots were manually segmented using LabelMe and then the number of pixels was counted by a self-developed python program. All these operations were carried out under the guidance and supervision of the grape disease specialist.

the Imp8 method is shown in **Figure 10**. The training set loss curves are shown in **Figure 10A**, where the red curve is before improvement and the blue curve is after improvement. When training until the model converged, the value of the red curve was about 0.17 and the value of the blue curve is about 0.132, which indicates that the improved model fit better on the training set than before improvement. **Figure 10B** shows the validation set loss curves, where the red curve is before improvement and the blue curve is after improvement. When training until the model converged, the value of the red curve was about 0.16, while the value of the blue curve was about 0.13, which indicates that the generalization ability of the model after the improvement was better than that before the improvement. Therefore, the improved DeepLab v3+ always converged faster and had better model fitting ability than the pre-improvement one whether on the training set or the validation set.

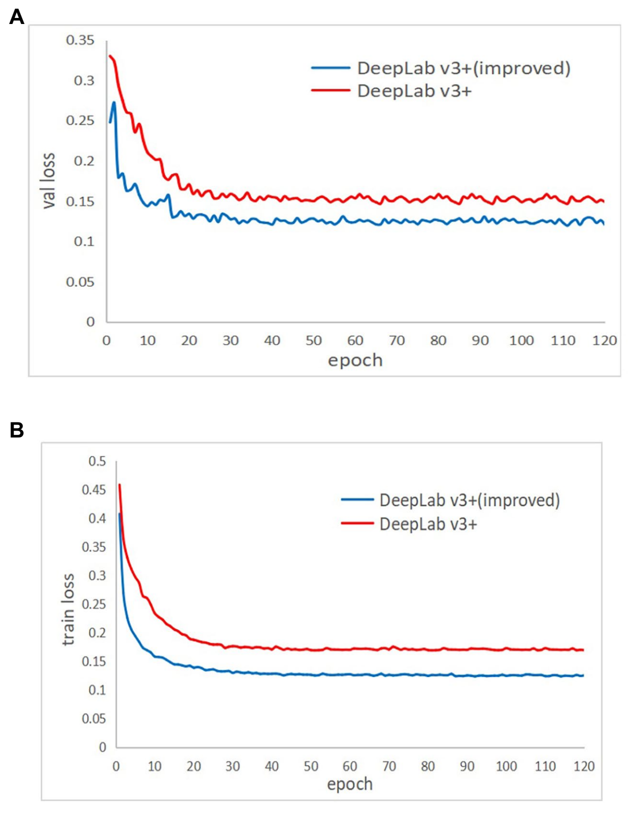


FIGURE 10 | Comparison of the training results of the network before and after the improvement of DeepLab v3+. **(A)** The training set, **(B)** the validation set.

DISCUSSION

Effect Comparison Between Detection and Segmentation for Disease Spots

The grape leaf black rot disease spots can be recognized in the previous research of our group, and the spots were accurately segmented from the background in this paper. The effect of disease spots detection and segmentation for test set TS1 is compared in **Figure 11**. **Figure 11A** shows

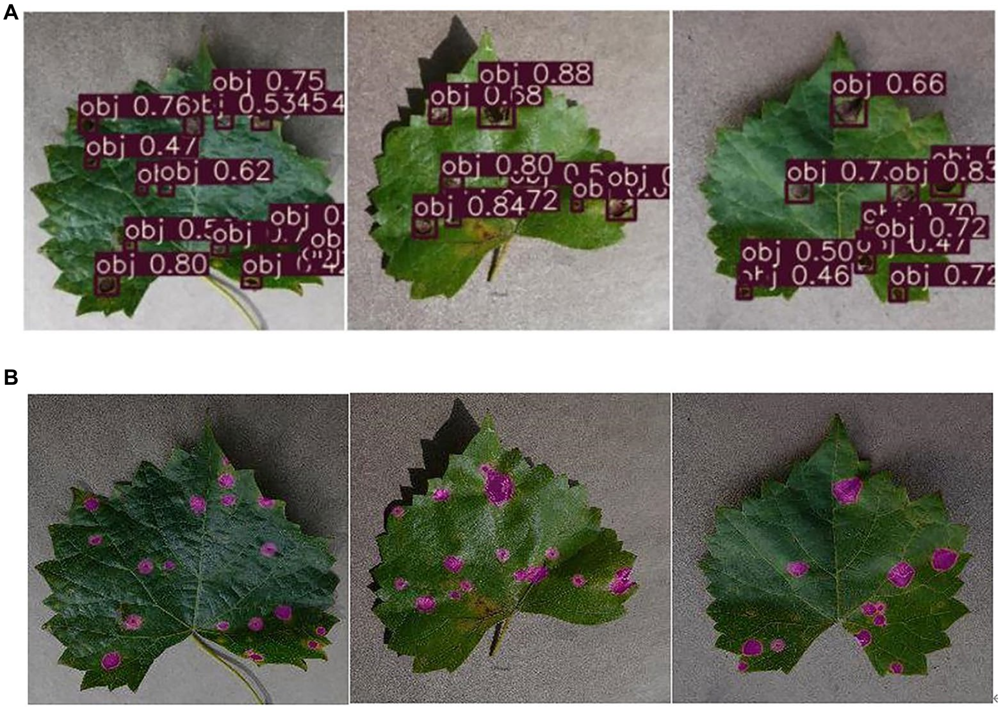


FIGURE 11 | The effect comparison between detection and segmentation on diseased spot. **(A)** The results of disease spots detection, **(B)** the results of disease spots segmentation.

the result of detection using the previous recognition method (Zhu et al., 2021), the number and location of the disease spots can be recognized, but cannot be segmented from the background. **Figure 11B** shows the result of segmentation using the method in this paper. The disease spots are not only recognized but also segment from the background according to their contour shape. **Table 5** shows the detection statistics results of the two methods for the grape leaves in **Figure 11**. As shown in **Table 5**, the segmentation method not only recognizes the number of disease spots but also obtains the number of pixels of spots. In addition, the segmentation method also detects and segments some tiny spots, which shows that this method is also better than the previous methods in recognition performance.

Comparison of Different Segmentation Algorithms

In this paper, DeepLab v3+ was chosen as the base algorithm to be improved for the segmentation of grape leaf black rot spots. This choice was based on the comparison of three common current mainstream deep learning segmentation algorithms. Pyramid Scene Parsing Network (PSP Net; Zhao et al., 2017) and U-Net are the other two common deep learning segmentation methods besides DeepLab v3+. PSPNet consists of a ResNet backbone that imposes a dilated convolution and a pyramid pooling module, which can mine global contextual information for fast network training. U-Net is an FCN with a simple structure, which can obtain very accurate segmentation results using few training images and is widely used in medical image analysis.

In this study, these three semantic segmentation networks were trained using the same dataset, and segmentation experiments of black rot spots were conducted on the test set TS1. **Figure 12** shows the segmentation results of three different networks. As shown, PSPNet could segment the black rot spots, but the network performed poorly for the segmentation of connected spots, and it mistakenly segmented the leaf part between two spots. The segmentation effect of U-net was better than PSPNet, which could separate the lesion area independently, but the segmentation was not fine enough. Improved DeepLab v3+ was better than the other two methods.

Table 6 shows the experimental statistical results of the different segmentation methods. In terms of ACC, there was no significant difference between the three methods, but in the mIOU metric, improved DeepLab v3+ was 10.6 and 4.4% higher than PSPNet and U-net, respectively. In terms of the *R* value, improved DeepLab v3+ was 8.2 and 3.4% higher than PSPNet and U-net, respectively. The experimental results showed that the improved DeepLab v3+ had better segmentation performance compared with PSPNet and U-net, and the improved DeepLab v3+ could further improve the segmentation performance of black rot spots on grape leaves.

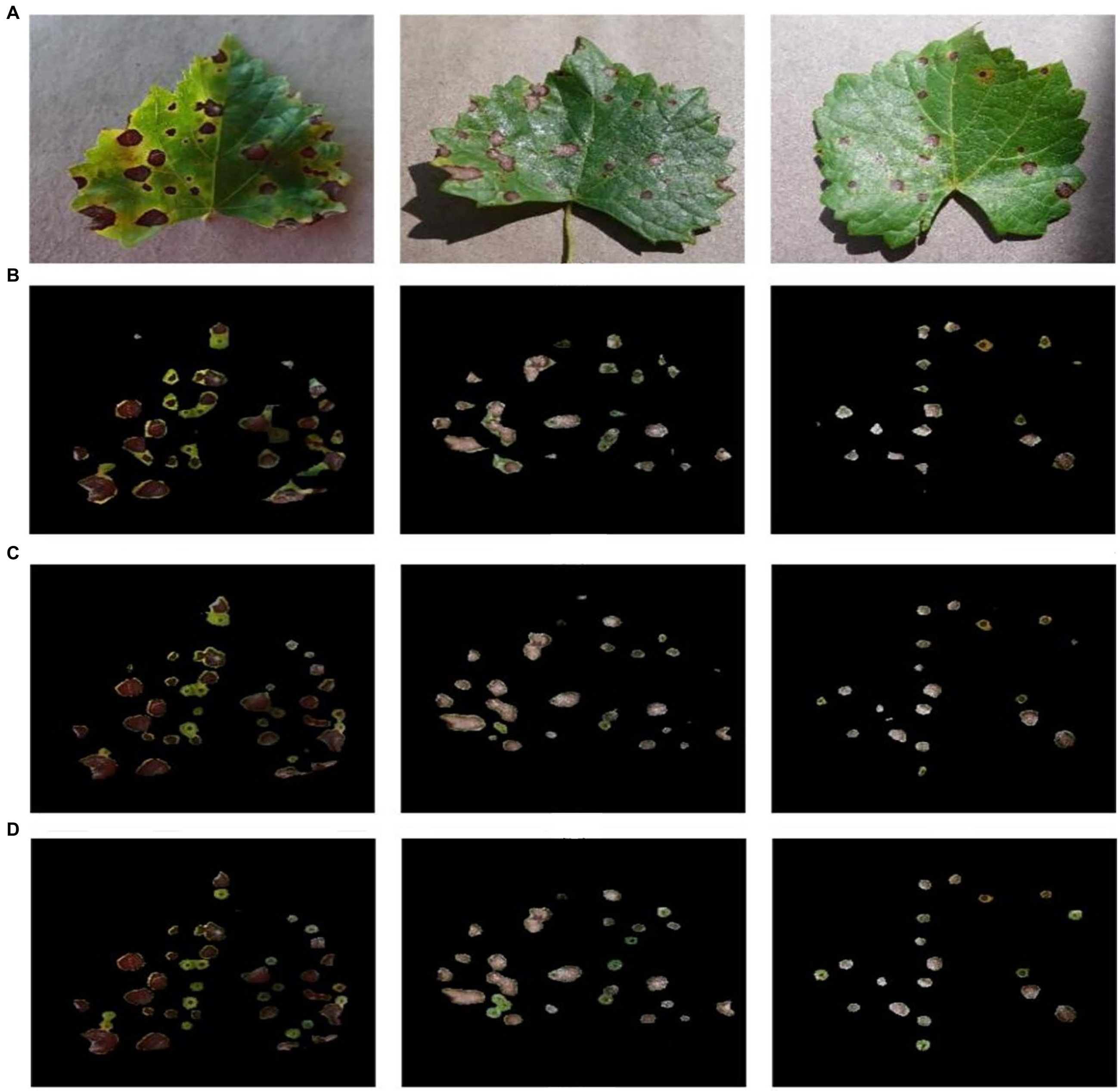


FIGURE 12 | Comparison of the segmentation results of different segmentation algorithms on the test set TS1 images. **(A)** The original image, **(B)** the PSP Net segmentation and extraction results, **(C)** the U-Net segmentation and extraction results, **(D)** improved DeepLab v3+ segmentation and extraction results.

TABLE 6 | Statistical segmentation results of different segmentation algorithms on the test set TS1 images.

Algorithm	**Evaluation indicators**					
	mIOU	**ACC**	**Dice**	***P***	***R***	***F1*-score**
PSP Net	0.767	0.972	0.868	0.929	0.814	0.868
U-Net	0.812	0.98	0.896	0.945	0.852	0.896
DeepLab v3+ (improved)	0.848	0.987	0.918	0.957	0.881	0.918

CONCLUSION

This paper proposes an improved DeepLab v3+ network model for the segmentation of black rot spots on grape leaves. This method inserts the ECA module into the residual module of the original DeepLab v3+ backbone network. Moreover, a feature fusion branch based on a FPN is added at the encoder end. One 4-fold up-sampling to two 2-fold up-sampling are modified in the original network. To verify the performance of the improved network model, two test sets based on Plant Village and an orchard field environment were constructed for experiments. The experimental results showed that the improved DeepLab v3+ network model exhibited better performance on both test sets than before improvement, and the improved model could be applied to the segmentation of black rot spots on grapes in real production environments. This approach can not only provide an effective tool for classifying grape disease extent classes but also be applied to the evaluation of other plant leaf and fruit diseases. In future work, we will attempt to combine super-resolution image enhancement with this approach to further improve the effect of small target recognition and segmentation.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

HY, JZ, and MC conceived the idea and proposed the method. JZ and QW contributed to the preparation of equipment and acquisition of data. JZ wrote the code and tested the method. JZ, QW, and MC validated results. HY and JZ wrote the paper. HY, JZ, and ZC revised the paper. All authors have read and approved the final manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (No. 32001412), the Key Research and Development Program of Hebei Province (19227206D).

REFERENCES

Appeltans, S., Pieters, J. G., and Mouazen, A. M. (2021). Detection of leek white tip disease under field conditions using hyperspectral proximal sensing and supervised machine learning. *Comput. Electron. Agric.* 190:106453. doi: 10.1016/j.compag.2021.106453

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, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Bai, X., Li, X., Fu, Z., Lv, X., and Zhang, L. (2017). A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images. *Comput. Electron. Agric.* 136, 157–165. doi: 10.1016/j.compag.2017.03.004

Chaudhary, P., Chaudhari, A. K., Cheeran, A. N., and Godara, S. (2012). Color transform based approach for disease spot detection on plant leaf. *Int. J. Comp. Sci. Telecom.* 3, 4–9.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. (2017). DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Match. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Pertanika J. Trop. Agric. Sci.* 11211, 137–143. doi: 10.1007/978-3-030-01234-2_49

Gao, C., Cai, Q., and Ming, S. (2020). YOLOv4 object detection algorithm with efficient Channel attention mechanism. in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, December 25, 2020, 1764–1770.

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*. December 2016, 770–778.

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

Hu, G., Wei, K., Zhang, Y., Bao, W., and Liang, D. (2021). Estimation of tea leaf blight severity in natural scene images. *Precis. Agric.* 22, 1239–1262. doi: 10.1007/s11119-020-09782-8

Hughes, D. P., and Salathe, M. (2016). An open access repository of images on plant heath to enable the development of mobile disease diagnostics. Available at: http://arxiv.org/abs/1511.08060v2

Jiang, F., Lu, Y., Chen, Y., Cai, D., and Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.* 179:105824. doi: 10.1016/j.compag.2020.105824

Jothiaruna, N., Sundar, K. J. A., and Karthikeyan, B. (2019). A segmentation method for disease spot images incorporating chrominance in comprehensive color feature and region growing. *Comput. Electron. Agric.* 165:104934. doi: 10.1016/j.compag.2019.104934

Kim, M. S., Lefcourt, A. M., Chen, Y. R., and Tao, Y. (2005). Automated detection of fecal contamination of apples based on multispectral fluorescence image fusion. *J. Food Eng.* 71, 85–91. doi: 10.1016/j.jfoodeng.2004.10.022

Kole, D. K., Ghosh, A., and Mitra, S. (2014). "Detection of downy mildew disease present in the grape leaves based on fuzzy set theory," in *Advanced Computing, Networking and Informatics. Vol. 1*. eds. M. K. Kundu, D. P. Mohapatra, A. Konar and A. Chakraborty (Cham: Springer International Publishing), 377–384.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., and Sun, W. (2019). PD 2 SE-net: computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron. Agric.* 157, 518–529. doi: 10.1016/j.compag.2019.01.034

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 30th IEEE Conference on Computer Vision and Pattern recognition*, *CVPR*, January 2017, 936–944.

Lin, K., Gong, L., Huang, Y., Liu, C., and Pan, J. (2019). Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Front. Plant Sci.* 10:155. doi: 10.3389/fpls.2019.00155

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, 3431–3440.

Ma, J., Du, K., Zhang, L., Zheng, F., Chu, J., and Sun, Z. (2017). A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. *Comput. Electron. Agric.* 142, 110–117. doi: 10.1016/j.compag.2017.08.023

Mehl, P. M., Chao, K., Kim, M., and Chen, Y. R. (2002). Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. *J. Agric. Saf. Health* 18, 219–226. doi: 10.13031/2013.7790

Milletari, F., Navab, N., and Ahmadi, S. A. (2016). V-net: fully convolutional neural networks for volumetric medical image segmentation. in *2016 4th International Conference on 3D Vision (3DV)*, IEEE, 2016, October 25, 2016, 565–571.

Molitor, D., and Berkelmann-Loehnertz, B. (2011). Simulating the susceptibility of clusters to grape black rot infections depending on their phenological development. *Crop Prot.* 30, 1649–1654. doi: 10.1016/j.cropro.2011.07.020

Pan, T. T., Chyngyz, E., Sun, D. W., Paliwal, J., and Pu, H. (2019). Pathogenetic process monitoring and early detection of pear black spot disease caused by *Alternaria alternata* using hyperspectral imaging. *Postharvest Biol. Technol.* 154, 96–104. doi: 10.1016/j.postharvbio.2019.04.005

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, October 5, 2015, 12–20.

Rundo, L., Han, C., Nagano, Y., Zhang, J., Hataya, R., Militello, C., et al. (2019). USE-net: incorporating squeeze-and-excitation blocks into U-net for prostate zonal segmentation of multi-institutional MRI datasets. *Neurocomputing* 365, 31–43. doi: 10.1016/j.neucom.2019.07.006

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* 77, 157–173. doi: 10.1007/s11263-007-0090-8

Samajpati, B. J., and Degadwala, S. D. (2016). Hybrid approach for apple fruit diseases detection and classification using random forest classifier. in *2016 International Conference on Communication and Signal Processing (ICCSP)*, April 6, 2016, 1015–1019.

Singh, V. (2019). Sunflower leaf diseases detection using image segmentation based on particle swarm optimization. *Artif. Intell. Agric.* 3, 62–68. doi: 10.1016/j.aiia.2019.09.002

Sinha, A., and Shekhawat, R. S. (2020). Olive spot disease detection and classification using analysis of leaf image textures. *Procedia Comput. Sci.* 167, 2328–2336. doi: 10.1016/j.procs.2020.03.285

Wang, C., Du, P., Wu, H., Li, J., Zhao, C., and Zhu, H. (2021). A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-net. *Comput. Electron. Agric.* 189:106373. doi: 10.1016/j.compag.2021.106373

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). ECA-net: efficient channel attention for deep convolutional neural networks. in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 13, 2020, 11531–11539.

Wspanialy, P., and Moussa, M. (2020). A detection and severity estimation system for generic diseases of tomato greenhouse plants. *Comput. Electron. Agric.* 178:105701. doi: 10.1016/j.compag.2020.105701

Yang, M., Yu, K., Zhang, C., Li, Z., and Yang, K. (2018). DenseASPP for semantic segmentation in street scenes. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, 3684–3692.

Yu, F., Koltun, V., and Funkhouser, T. (2017). "Dilated residual networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, 636–644.

Zagoruyko, S., and Komodakis, N. (2016). Wide residual networks. in *British Machine Vision Conference (BMVC)*, September 2016, 87.1–87.12.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. in *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, January 2017, 6230–6239.

Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., and Shimizu, M. (2014). Disease detection of Cercospora leaf spot in sugar beet by robust template matching. *Comput. Electron. Agric.* 108, 58–70. doi: 10.1016/j.compag.2014.07.004

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, 1–16. doi: 10.3389/fpls.2021.695749

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.

frontiers in Plant Science

ORIGINAL RESEARCH
published: 03 March 2022
doi: 10.3389/fpls.2022.810546

MSR-RCNN: A Multi-Class Crop Pest Detection Network Based on a Multi-Scale Super-Resolution Feature Enhancement Module

*Yue Teng[1,2], Jie Zhang[1], Shifeng Dong[1,2], Shijian Zheng[3] and Liu Liu[4]**

[1] Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Science, Hefei, China, [2] Science Island Branch, University of Science and Technology of China, Hefei, China, [3] Department of Information Engineering, Southwest University of Science and Technology, Mianyang, China, [4] Department of Computer Science and Engineering, Shanghai JiaoTong University, Shanghai, China

OPEN ACCESS

Edited by:
Po Yang,
The University of Sheffield,
United Kingdom

Reviewed by:
Marcin Wozniak,
Silesian University of Technology,
Poland
Sijia Yu,
Rutgers, The State University of New Jersey - Busch Campus,
United States

***Correspondence:**
Liu Liu
liuliu1993@sjtu.edu.cn

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection,
a section of the journal Frontiers in Plant Science

Received: *07 November 2021*
Accepted: *25 January 2022*
Published: *03 March 2022*

Citation:
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

Pest disaster severely reduces crop yield and recognizing them remains a challenging research topic. Existing methods have not fully considered the pest disaster characteristics including object distribution and position requirement, leading to unsatisfactory performance. To address this issue, we propose a robust pest detection network by two customized core designs: multi-scale super-resolution (MSR) feature enhancement module and Soft-IoU (SI) mechanism. The MSR (a plug-and-play module) is employed to improve the detection performance of small-size, multi-scale, and high-similarity pests. It enhances the feature expression ability by using a super-resolution component, a feature fusion mechanism, and a feature weighting mechanism. The SI aims to emphasize the position-based detection requirement by distinguishing the performance of different predictions with the same Intersection over Union (IoU). In addition, to prosper the development of agricultural pest detection, we contribute a large-scale light-trap pest dataset (named LLPD-26), which contains 26-class pests and 18,585 images with high-quality pest detection and classification annotations. Extensive experimental results over multi-class pests demonstrate that our proposed method achieves the best performance by 67.4% of mAP on the LLPD-26 while being 15.0 and 2.7% gain than state-of-the-art pest detection AF-RCNN and HGLA respectively. Ablation studies verify the effectiveness of the proposed components.

Keywords: agricultural pest detection, convolutional neural network, feature enhancement, Soft-IoU, wisdom agriculture

1. INTRODUCTION

The pest disaster is considered as the main reason for crop yield reduction, thus recognizing pests is necessary to guarantee crop yield. Manual pest recognition and location are time-consuming and laborious work. Traditional pest recognition methods prefer to design feature vectors to identify specific pest species, which lacks the generalization ability (Qing et al., 2012; Wang et al., 2012; Yaakob and Jain, 2012; Wen et al., 2015; Deng et al., 2018). Differently, deep learning-based methods using object detection as a ready-to-use approach cause unsatisfied performance due to the enormous gap between pest detection and generic object detection, which could be summarized into the differences in object characters and detection requirements.

The gaps of object characters include small-size, multi-scale, and high-similarly. Small size is the most distinguished property of general object detection. Taking the PASCAL VOC dataset (Everingham et al., 2010) and the LLPD-26 dataset we build as an example, the average size of pests (annotated by bounding boxes) is 1.58% of the general object bounding boxes. Existing methods fail to pay close attention to the small-size pests, which leads to insufficient recognition accuracy. The multi-scale property is another difference between pest detection and general object detection. The object size distribution is wide in pest detection tasks (e.g., the size of Gryllotalpa Orientalis Burmeister is 32 times larger than that of Nilaparvata Lugens Stal in our LLPD-26 dataset). Existing pest detection methods usually use feature fusion of adjacent layers to solve the multi-scale problem, but this fusion method is not sufficient to fully integrate information from different feature layers. The high similarity of interclass is also a crucial challenge (such as Mythimna Separata and Helicoverpa Armigera). Due to the low discrimination ability of high-similarly pests, the performance of the existing methods makes it unsuitable for practical application and remains to be improved.

Furthermore, position attention is more crucial for pest detection than the high-value Intersection over Union (IoU) compared to general object detection. Different prediction bounding boxes with the same IoU value have diverse performance, as shown in **Figure 1**. All the predicted bounding boxes (red boxes) in **Figure 1** have the same IoU value, but it is clear that the pest detection results are more accurate than the general object detection because there are lesser irrelevant pixels of other categories enclosed (as shown in **Figure 1D**). The result of **Figure 1A** is more accurate than the result of **Figure 1B** because **Figure 1A** contains all of the pest pixels. Therefore, detection bounding boxes with low IoU hardly cause trouble for pest detection since it excludes other class pixels. Existing methods usually adopt the hard IoU threshold to determine positive and negative samples. By doing so, it could cause some high-quality bounding boxes to be taken as negative samples.

In summary, this study focuses on reducing the gaps between general object detection and pest detection in two dimensions (pest bounding box character and detection target) to improve the performance of pest detection. In pest bounding box dimension: (1) Existing pests detection methods and general object detectors usually utilize FPN (Lin et al., 2017a) to improve the multi-scale feature extraction ability by top-to-down adjacent feature fusion method, but the incomplete fusion limits the performance of detectors. (2) High-similarly objects are recognized using channel attention (Hu et al., 2018) in the general detection field, but the single dimension attention is insufficient for pest detection. (3) The pattern of 5-layer feature maps is employed to detect objects, in which the top layer is used to recognize large-size objects and the down layer is used to recognize small-size objects, but the pest's size is far less than general objects (like dog and cat) resulting in the feature gradually disappearing with the convolution operation. In the pest detection target dimension, pest detections pay more attention to position rather than high-value IoU. Existing methods use a hard IoU threshold to distinguish positive and negative samples resulting in inadequate detection performance. To solve the defect of existing pest detection methods, we propose an MSR-RCNN to improve the detection performance of small-size, multi-scale, and high-similarly pests. The MSR module, the highlight of MSR-RCNN, is a plug-and-play component and can improve the performance of familiar detectors. We first use the super-resolution method to enhance small-size features. Multi-level features are fused at once by feature full fusion mechanism to promote the information transition and high-similarly pests are adequately recognized by feature full weighting mechanism to enhance feature expression ability. In addition, SI is a new design to distinguish different predict bounding boxes with the same IoU value and make networks more suitable for pest detection. Furthermore, to promote the development of pest detection and verify the feasibility of our methods, we construct a large-scale light-trap pest dataset (named LLPD-26) including 18,585 images and 26 classes. Abundant experiments on the LLPD-26 show that our methods can effectively detect multi-class pests and attain start-of-the-art (SOTA) performance.

The main contributions are listed as follows:

- We propose a novel pest detection network (named MSR-RCNN) to solve the defect that existing methods lack the targeted improvement of pest objects in three dimensions: small-size, multi-scale, and high-similarly. The highlight of our MSR-RCNN is the multi-scale super-resolution (MSR) feature enhancement module that can improve the performance of familiar detectors by plug-and-play pattern. The MSR module consists of the super-resolution component, the feature full fusion mechanism, and the feature full weighting mechanism. The three parts focus on improving the performance of small-size, multi-scale, and high-similarly pests.
- Since pest detection focus on the position rather than high-value IoU, we design a SI to differentiate the performance of different prediction result with the same IoU. The SI generates high-quality bounding boxes for network training and employs suitable results to test for pest detection. By using the Soft-IoU, our MSR-RCNN is more fit for pest detection tasks. Meanwhile, the performance of the network is improved without other costs.
- To more accurately monitor and detect multi-class crop pests, we construct a large-scale light-trap pest dataset (named LLPD-26) including 18,585 images and 26 classes. The most-species and largest-number characters of LLPD provide conditions for accurately detecting pests. In addition, adequate experiments on the LLPD-26 verify that our MSR-RCNN outperforms other SOTA methods.

2. RELATED STUDY

2.1. Deep Learning-Based Object Detection

Pest detection is a specific task of general object detection. In recent years, Convolutional Neural Network (CNN) is widely applied in the object detection fields. The deep learning-based object detection networks divide into one-stage networks and

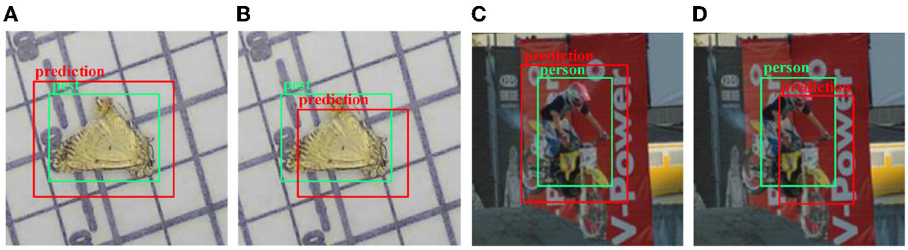


FIGURE 1 | The schematic diagram of different prediction bounding boxes with the same Intersection over Union (IoU). **(A)** The prediction box contains all object pixels. **(B)** The prediction box contains almost all object pixels. **(C)** The prediction box contains pixels of another category (motorbike). **(D)** Most of the pixels in the prediction box are other categories (motorcycles).

two-stage networks. As one of the most famous networks in the one-stage, Redmon et al. (2016) utilized the whole image as the input and directly obtained the prediction result using 24 convolution layers and 2 full connection layers. Subsequently, some enhanced versions of YOLO were proposed one after another (Redmon and Farhadi, 2017, 2018; Bochkovskiy et al., 2020). Lin designed Retinanet to solve the problem of positive and negative sample imbalance with the Focal Loss, thus improving the detection accuracy (Lin et al., 2017b). The FCOS avoided the anchor mechanism with the pattern of point regression resulting in reducing the number of hyperparameters. Meanwhile, low-quality predictions were filtered out through the proposed Center-ness branch (Tian et al., 2019). Two-stage networks require the selective search (Uijlings et al., 2013) or region proposal network (RPN) to generate region proposal first, and then the R-CNN network (Girshick et al., 2014) is used to refine the proposal box (Girshick, 2015). Faster R-CNN (Ren et al., 2017) proposed RPN based on the Fast R-CNN and established the baseline of the two-stage detector. Pang et al. designed the Cascade R-CNN network to continuously optimize the detection results by gradually increasing the IoU threshold (Cai and Vasconcelos, 2018). Libra R-CNN used concat to merge feature layers, but the essence of the feature fusion method was reducing the video memory for the non-local mechanism (Pang et al., 2019). FPN (Lin et al., 2017a) and PANet (Liu et al., 2018) used feature fusion of adjacent layers to solve the multi-scale problem, but the incomplete fusion method did not meet the requirement of pest detection. TridentNet used dilated convolution (Yu and Koltun, 2015) to improve the capability of multi-scale feature extraction (Li et al., 2019). The ThunderNet used Context Enhancement Module (CEM) module to integrate multi-scale information and adopted the Spatial Attention Module (SAM) to enhance feature representation (Qin et al., 2019). OHEM (Shrivastava et al., 2016) and Snip/Sniper (Singh and Davis, 2018; Singh et al., 2018) improved the performance of the network by using selective backpropagation. We use the two-stage framework as the baseline because the two-stage methods are usually more accurate than the one-stage methods, especially for small-size object detection.

2.2. Pest Detection Method Based on CNN

Due to the rapid development of CNN-based object detection, many researchers transplant deep learning-based methods to agricultural applications (Kamilaris and Prenafeta-Boldú, 2018; Dhaka et al., 2021; Hasan et al., 2021). In the pest recognition and detection field, Liu et al. (2016) used a global contrast region-based approach to construct a rice insect classification dataset named Pest_ID and used a CNN to identify the insects. Wang et al. (2017) applied LeNet and AlexNet to classify pest images. Thenmozhi and Reddy (2019) used transfer learning to explore the results of AlexNet, ResNet, LeNet, and VGG on three pest datasets. Yue et al. (2018) proposed a super-resolution method based on deep learning to solve the difficulty of insect recognition. Ayan et al. (2020) combined different CNN networks into a unified pest identification network and automatically selected the combination weight to carry out pest identification *via* the genetic algorithm. Shen et al. (2018) proposed an improved Faster R-CNN network with the inception structure to identify common grain pests. Liu et al. (2019) designed a detection network combining Faster R-CNN and channel-spatial to detect the light-trap pests. Jiao et al. (2020) proposed an anchor-free network (AF-RCNN) to identify and locate pests of 24 types. Liu et al. (2020) used global and local activation features to detect the 16-class pest dataset. The above methods ignore the gaps between object detection and pest detection and use insufficient improvement for pest detection, which led to an unsatisfied performance. Therefore, we design an MSR-RCNN to improve the performance of pest detection.

3. MATERIALS AND METHODS

3.1. Data Collection

We use the light-trap device to automatically collect the pest images in different periods. The data collection devices are from the Intelligent Machines Institute, Chinese Academy of Sciences, and distributed in the field environment of Anhui Province. The dataset includes 18,585 JPEG images with the resolution of 2,592×1,944 and is annotated by agricultural experts. Each pest object corresponds to a unique category and bounding box coordinate, and each image has multiple pests. To ensure effectiveness, we divide the data into 14,868 images of the train set and 3,717 images of the test set.

3.2. MSR-RCNN Pest Detection Network

To accurately detect 26-class pests, we design an MSR-RCNN network including a backbone network (ResNet50), MSR feature

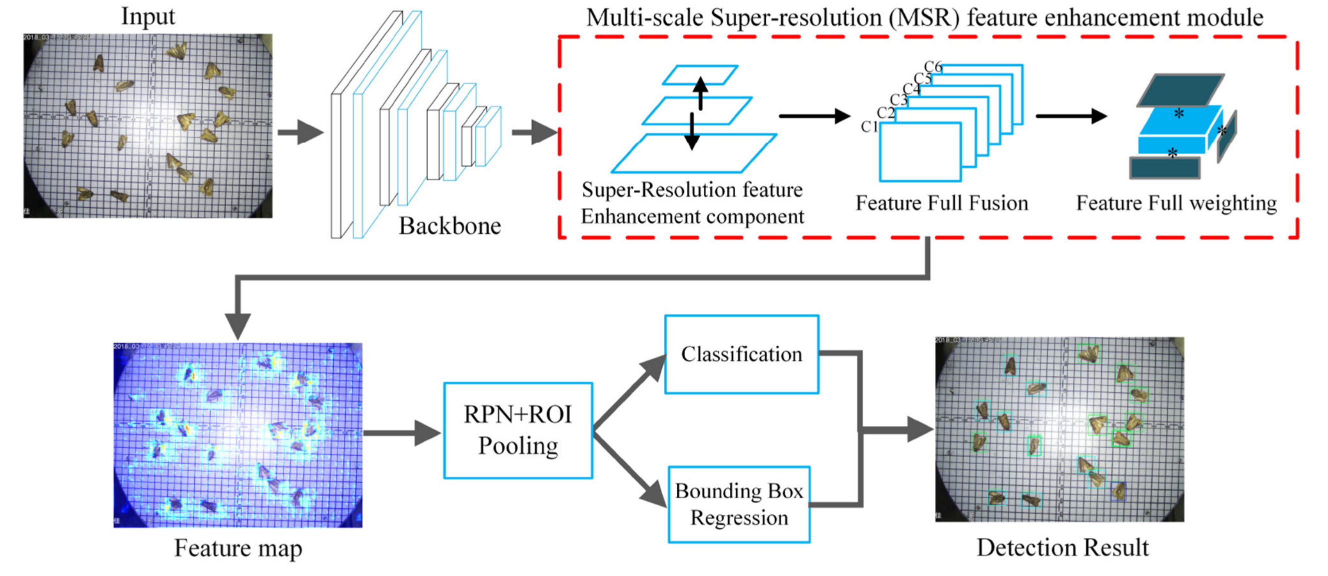


FIGURE 2 | The overall framework of the MSR-RCNN.

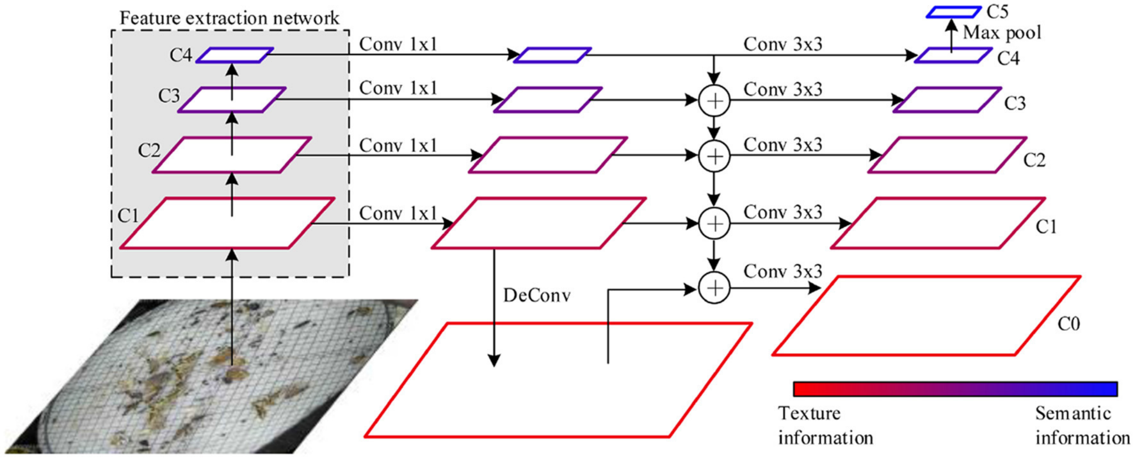


FIGURE 3 | The super-resolution feature enhancement component.

enhancement module, RPN, and bounding box regression and classification networks (RCNN). We use ResNet50 (He et al., 2016) as the backbone network to extract image features. The MSR feature enhancement module is utilized to improve the feature expression ability of the backbone in three dimensions: small-size, multi-scale, and high-similarly. With the MSR module, enhanced features are obtained for pest detection. The RPN (Ren et al., 2017) is used to obtain the region of interest (ROI) and the ROI Align (He et al., 2017) is employed to resize the ROI to the unitive size. Classification branch and bounding box regression are applied to obtain the final detection results, as shown in **Figure 2**.

3.3. MSR Feature Enhancement Module

Since small-size, multi-scale, and high-similarly pest characters of pests, we design the MSR feature enhancement module to improve the detection performance using a super-resolution component, a feature full fusion mechanism, and a feature full weighting mechanism. The super-resolution component from the MSR module obtains the six-layer feature map for the recognition of small size objects. Then, the full feature fusion mechanism integrates all features at once for the recognition of multi-scale objects. Since high-similarly pests in the LLPD-26 dataset are difficult to identify, we design the feature full weighting mechanism in the MSR module to enhance the fine-grained expression ability. The red part of

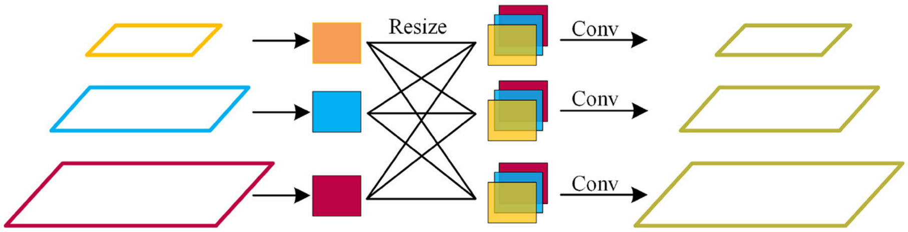


FIGURE 4 | The feature full fusion mechanism.

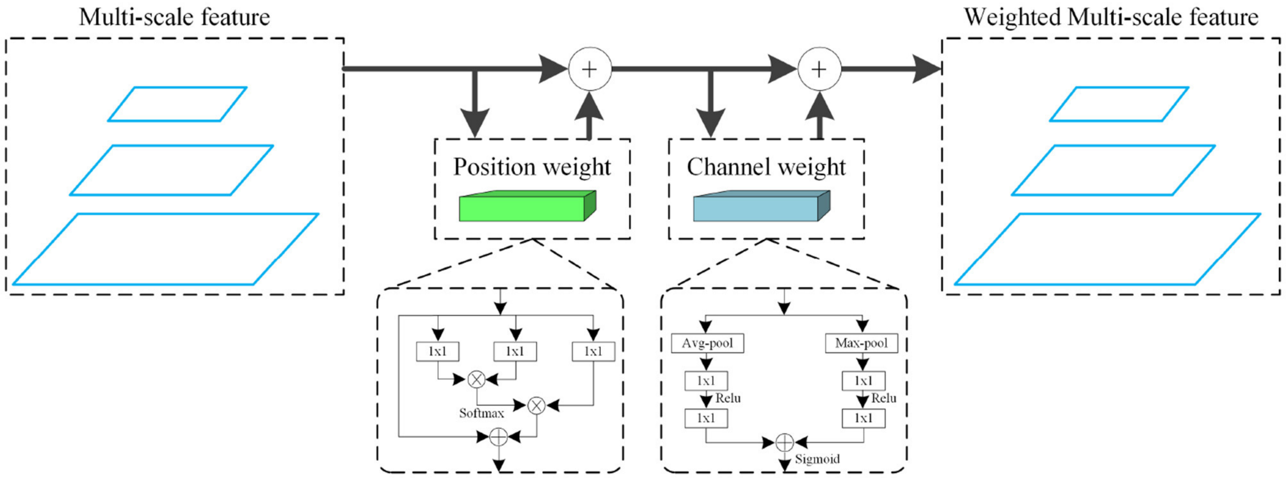


FIGURE 5 | The feature full weighting mechanism.

Figure 2 shows the overall framework of the MSR we devised.

3.3.1. Super-Resolution Feature Enhancement Component

Feature pyramid network (FPN) (Lin et al., 2017a) uses 5 layer feature maps to recognize objects, in which the top-level features include semantic information to detect large-size objects and the low-level features include texture information to detect small-size objects. However, the small-size pest features gradually disappear in the process of convolution operation resulting in misleading information transfer in the top-to-down feature fusion. Inspired by zooming in to identify pests in the manual annotation process, we design the super-resolution feature enhancement component to improve small-size feature extraction ability by using deconvolution to obtain fine-grained pest features.

To ensure the full utilization of features, we select the feature maps after each Resnet50 block (a total of 4) as the input of the super-resolution component. We use 1 x 1 convolution kernels for each layer feature to change the number of channels to 256. Duo to the size of pest objects is small, we deconvolve the feature map after the first block of the Resnet50 network to enhance texture information, which refers to the way people zoom in on images for small-size object recognition. In this way, we have 5-layer feature maps, four layers from the feature extraction network, and one layer from deconvolution operation. We use the bilinear interpolation method to add the upper layer features and apply the lower layer features to carry out adjacent layer feature fusion. The 3 x 3 convolution kernel is utilized to enhance the feature representation capability. Max pooling operation is carried out for top layer feature to enhance semantic information. After the above process, we have 6-layer feature maps, in which the top layer feature obtained by max-pooling has sufficient semantic information, and the bottom layer feature obtained by deconvolution has rich texture information. **Figure 3** shows the super-resolution feature enhancement component designed in this study.

3.3.2. Feature Full Fusion

The feature full fusion mechanism is used to improve the performance of multi-scale pest detection. By fusing the information of different feature layers, the defects are avoided in existing methods, which only combine adjacent layers or use a single feature layer to detect pests (Jiao et al., 2020; Liu et al., 2020). The inspiration for our design comes from the process of people looking at images. People often think of an image as a 2D image because the human eye treats multiple channels (usually RGB, 3-channel) at once. Similarly, the feature full fusion mechanism combines the 6-layer features from the super-resolution component at once. We fuse 6-layer feature maps into

TABLE 1 | The overall performance comparison.

Method	MSR	SIoU	*AP*	AP_{50}	AP_{75}	*mRecall*
General object detection						
Faster R-CNN (Ren et al., 2017)			35.4	62.3	37.7	50.5
Cascade R-CNN (Cai and Vasconcelos, 2018)			36.0	62.6	38.5	50.2
Libra R-CNN (Pang et al., 2019)			37.4	65.2	40.2	52.8
FCOS (Tian et al., 2019)			33.3	57.4	36.2	**55.2**
RetinaNet (Lin et al., 2017b)			27.9	48.8	29.4	53.1
Pest detection						
AF-RCNN (Jiao et al., 2020)			33.1	58.6	34.6	48.8
HGLA (Liu et al., 2020)			37.0	65.6	38.3	52.0
Ours						
MSR-RCNN	√		38.0	66.9	40.0	52.4
MSR-RCNN	√	√	**38.4**	**67.4**	**40.6**	52.0

TABLE 2 | Compare results by category on our LLPD-26 dataset using AP_{50}.

Class number	General object detection				Pest detection		Ours	
	Faster R-CNN	Cascade R-CNN	Libra R-CNN	RetinaNet	AF-RCNN	HGLA	MSR	MSR+SI
1	16.1	19.2	21.7	4.5	12.8	20.1	**21.1**	20.4
2	58.7	58.9	63.5	54.4	58.7	63.3	66.1	**67.2**
3	70.2	67.9	70.1	60.9	65.5	71.7	72.8	**74.0**
4	69.6	69.4	70.9	58.0	66.0	72.8	72.3	**72.8**
5	84.9	85.2	85.0	80.7	83.5	86.1	**86.2**	85.8
6	72.1	71.1	74.4	66.0	70.4	76.2	**76.4**	77.4
7	72.5	71.9	73.4	62.4	70.9	74.0	**74.9**	74.5
8	62.0	60.6	66.7	57.5	59.4	66.1	65.5	**70.5**
9	47.5	47.5	50.9	43.0	47.3	51.9	**53.6**	53.5
10	70.9	70.5	74.2	59.6	68.5	74.2	77.2	**78.8**
11	79.3	78.2	80.3	73.2	76.1	81.6	81.0	**81.7**
12	27.7	26.9	26.7	0.10	25.5	**32.7**	29.5	32.3
13	55.3	58.3	54.6	41.5	53.4	55.4	56.8	**59.7**
14	66.7	64.5	66.4	57.3	62.0	67.4	67.5	**69.9**
15	39.8	45.3	47.3	8.10	33.1	45.2	48.0	**51.3**
16	40.2	45.2	51.7	7.50	33.0	50.7	49.6	**51.5**
17	57.9	65.1	66.8	15.0	55.4	70.8	**70.9**	70.6
18	56.1	58.7	60.5	35.9	55.0	58.0	**64.9**	63.3
19	56.6	58.4	64.9	54.3	57.7	61.9	65.1	**69.4**
20	83.0	82.7	82.1	78.1	80.6	83.7	83.3	**83.8**
21	89.5	89.5	89.5	86.9	87.5	90.0	90.0	**90.1**
22	93.1	92.4	94.4	93.8	91.7	94.4	**94.6**	94.4
23	59.9	51.7	63.2	54.1	54.1	61.2	**66.0**	63.9
24	72.8	73.3	74.9	64.1	71.4	74.8	74.0	**75.2**
25	53.3	49.7	54.8	1.20	14.8	50.0	**59.2**	56.4
26	64.8	70.4	65.0	49.6	68.2	70.2	**73.0**	63.1
Mean	62.3	62.8	65.2	48.8	58.6	65.6	66.9	**67.4**

The parts in bold represent the best performance.

five layers to improve network efficiency. Specifically, for each of the 6-layer feature maps, we use the bilinear interpolation method to resize them to five sizes, in which the resolutions are 200×272, 100×136, 50×68, 25×34, and 13×17, respectively. We stack features of the same size and use a 1×1 convolution to unify channels to 256. The stacked feature maps are added to the

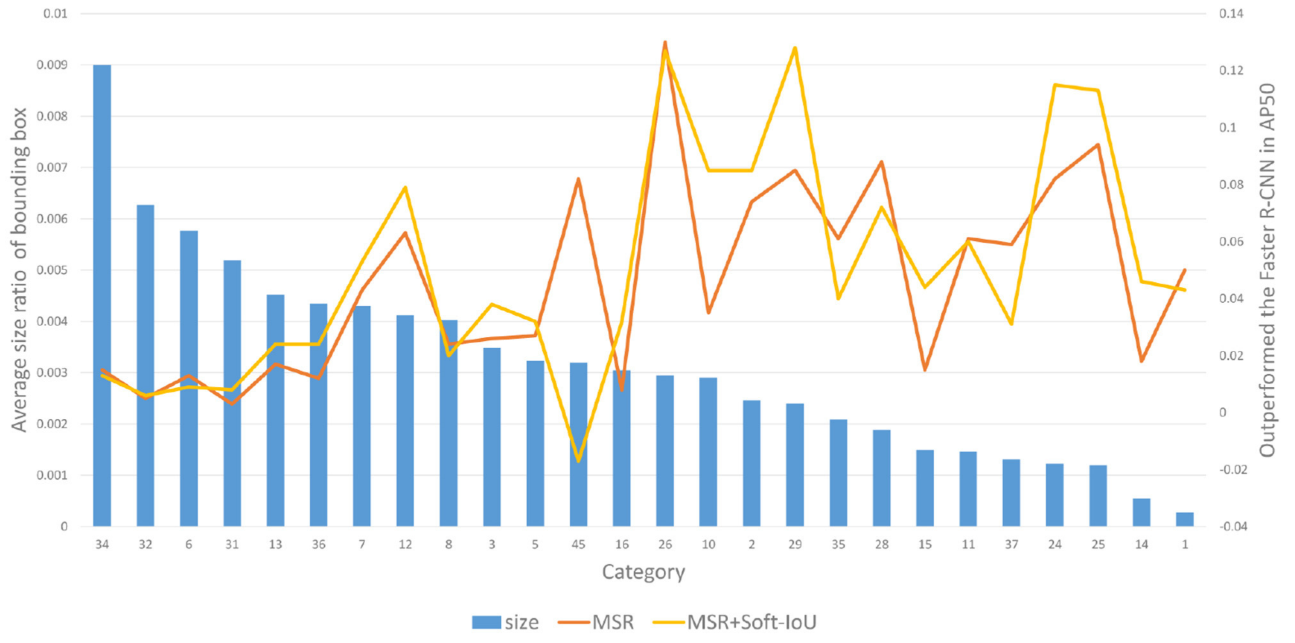


FIGURE 6 | Improved performance of our MSR-RCNN on pest data of different sizes.

C1∼C5 feature maps of the original feature maps. It is important to note that our feature full fusion is substantially different from the full connection layer, although it is very similar. This is because our feature full fusion module preserves the translation invariance of the pixels. This also leaves enough information for the next feature full weighting module. **Figure 4** shows the feature full fusion mechanism.

3.3.3. Feature Full Weighting

Due to the high-similarly pests in the LLPD-26 (e.g., *Cnaphalocrocis medinalis and Pyrausta nubilalis, Mamestra brassicae Linnaeus and Scotogramma trifolii Rottemberg*), fine-grained identification is required to improve the performance of detection. We design the feature full weighting for feature reinforcement learning. This could optimize the detection performance of similar pests from two dimensions (depth and location). For the feature map (W, H, and C) of each layer, our weighting method weights channel C and points (x, y) in the feature map, where W is the width, H is the height, and C is the channel number of the feature map. We use Formula (1) to describe our weighting method.

$$W(X) = \alpha\pi_L(X)g(X) + (1-\alpha)\pi_C(X)X \qquad (1)$$

Where $\pi_L(\cdot)$ represents the local weighting function, $\pi_C(\cdot)$ represents the channel weighting function, X represents the feature map, $W(X)$ represents the weighted feature map, and α is the scale factor. Formula (2) and Formula (3) give the specific forms of $\pi_L(\cdot)$ and $\pi_C(\cdot)$, respectively.

$$\pi_L(x_i) = \sum_{\forall j \in X} \theta_L(x_j)^{\mathrm{T}}\phi_L(x_i) \qquad (2)$$

$$\pi_C(X) = ReLu(\theta_C(avg(X))) + ReLu(\phi_C(max(X))) \qquad (3)$$

Among them, x_j represents the point on the feature map excluding the point X_i, $\theta(\cdot)$ and $\phi(\cdot)$ represent the learnable function for feature X, $avg(\cdot)$ and $max(\cdot)$ represent global average pooling and global maximum pooling, respectively. To guarantee the end-to-end pattern, we use a convolution operation to carry out the feature full weighting, as shown in **Figure 5**.

3.4. Soft-IoU

In general object detection (such as PASCAL VOC), IoU_{50} is used as the threshold to determine positive and negative samples. However, for pest detection, different bounding boxes with the same IoU value have different performances. Therefore, we design a SI with the position suppression method to optimize the training and test processes. Specifically, the calculation method of SI is shown in Formula (4):

$$SI(A,B) = \beta \cdot \lceil 1 - \frac{E(A_{center}, B_{center})}{max(A_{diagonal}, B_{diagonal})} \rceil \cdot \frac{A \cap B}{A \cup B} \qquad (4)$$

Where $E(\cdot)$ represents the Euclidean distance, A_{center} and B_{center} represent the center point of bounding box A and B, respectively, $A_{diagonal}$ and $B_{diagonal}$ represent the diagonal distance of bounding box A and B, respectively, $Max(\cdot)$ represents the maximum function, and β is the scaling factor. To ensure the stability, we adjust the IoU no more than 0.1 times the original IoU. Due to the high-quality positive samples contributing to training the network finely, β is selected as 0.9. In the test phase, $\beta = 1.1$ because we expect the bounding box as shown in **Figure 1A** to output the results as a positive sample.

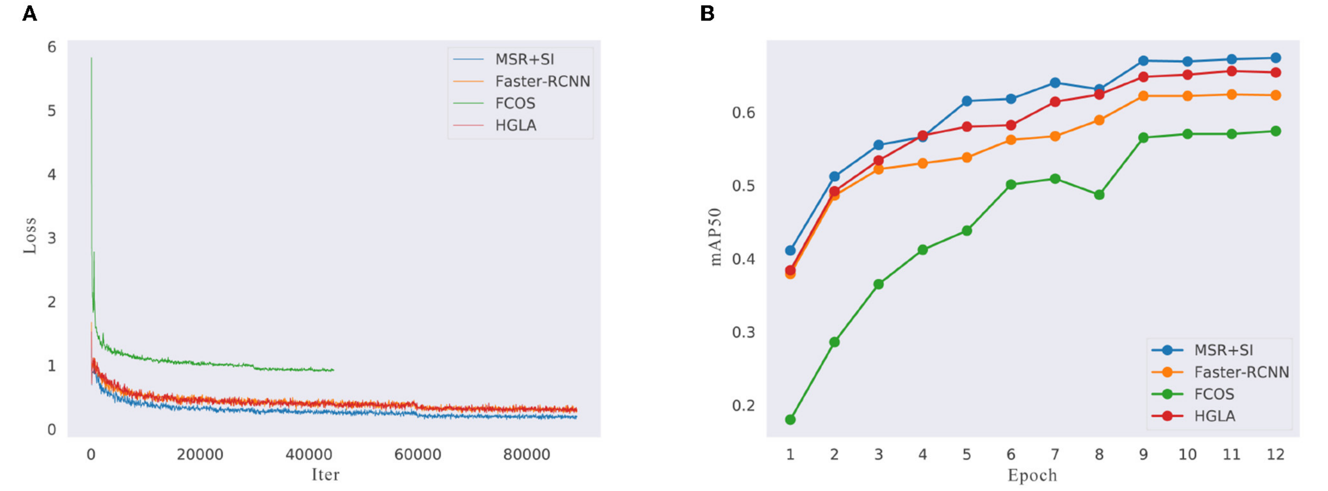


FIGURE 7 | The training loss and mAP_{50}. **(A)** The comparison of training loss. **(B)** The comparison of test accuracy.

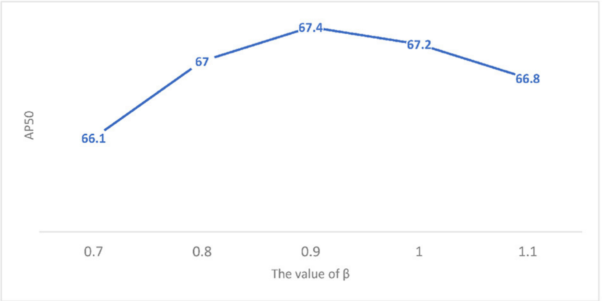


FIGURE 8 | Ablation of β in the Soft-IoU (SI).

TABLE 3 | MSR-RCNN network performance comparison results using different backbones.

	Resnet50	Resnet101	Resnext50	Resnext101
AP_{50}	**66.9**	66.1	66.3	66.7
AP_{75}	40.0	39.4	**40.3**	39.6
AP	**38.0**	37.4	**38.0**	37.8

The parts in bold represent the best performance.

4. EXPERIMENTS

4.1. Experiment Settings

We use the backpropagation and Stochastic Gradient Descent (SGD) to train our MSR-RCNN (LeCun et al., 1989). For the training of MSR-RCNN, each SGD mini-batch is constructed from a single pest image that contains 256 samples. Negative samples and positive samples are randomly selected in a ratio of 1 : 1 in each mini-batch. Gaussian distribution with a mean of 0 and a SD of 0.01 is used to initialize the parameters of the classification regression layer. In each SGD iteration, we use RPN to generate 1,000 potential regions to be sent to R-CNN for learning. We train a total of 12 epochs with a momentum of 0.9, among which the first 8 epochs have a learning rate of 0.0025, and the last 4 epochs are 0.00025. Our experiment is deployed on a Dell 750 server with NVIDIA Titan RTX GPU (24G memory) using the Mmdetection2.0.0 (Chen et al., 2019) framework and Python 3.8. Unless otherwise stated, all comparison models in this study use the default parameters. Since the SmoothL1 Loss function is differentiable at zero, we use it to train the R-CNN network for more stable performance. Because the L1 Loss is a non-differentiable function at zero, we apply it in RPN network training to improve the robustness.

4.2. Experiment Results

4.2.1. Performance on Our LLPD-26

We compare the performance of our method with Faster R-CNN (Ren et al., 2017), Cascade R-CNN (Cai and Vasconcelos, 2018), Libra R-CNN (Pang et al., 2019), FCOS (Tian et al., 2019), Retinanet (Lin et al., 2017b), AF-RCNN (Jiao et al., 2020), and HGLA (Liu et al., 2020), as shown in **Table 1**. Among them, AF-RCNN and HGLA are the existing deep learning-based pest detection methods, MSR represents the MSR feature enhancement module proposed by us, SI represents the SI, AP_{50} represents the Average Precision (AP) with the IoU threshold of 50%, AP represents the mean AP with the IoU threshold at 50, 75, and 95%. The FPN (Lin et al., 2017a) is used in all comparison methods. Our MSR module is slightly inferior to Libra R-CNN in AP_{75} performance due to the high-quality training box provided by the balanced sampling approach of Libra R-CNN. In addition, since pest detection is more focused on point location performance than bounding box IoU performance, AP_{50} is more valuable than AP_{75}. With the SI training method, the MSR-RCNN outperforms other methods.

To compare the performance of the proposed method in detail, the AP_{50} results of each category are given in **Table 2**. We

TABLE 4 | The performance of MSR with various detection methods.

Method	MSR	*AP*	*AP*$_{50}$	*AP*$_{75}$	*mRecall*
Faster R-CNN (Ren et al., 2017)		34.8	61.8	36.1	51.5
Faster R-CNN + FPN (Lin et al., 2017a)		35.4	62.3	37.7	50.5
Faster R-CNN + MSR	√	**37.6**	**66.3**	**39.5**	**51.9**
Cascade R-CNN + FPN (Cai and Vasconcelos, 2018)		36.0	62.6	38.5	50.2
Cascade R-CNN + MSR	√	**37.8**	**65.8**	**40.2**	**52.1**
FCOS + FPN (Tian et al., 2019)		33.1	57.0	35.9	**55.3**
FCOS + MSR	√	**33.8**	**58.8**	**36.0**	54.8
RetinaNet + FPN (Lin et al., 2017b)		27.9	48.8	29.4	**53.1**
RetinaNet + MSR	√	**30.8**	**53.1**	**33.3**	52.7

The parts in bold represent the best performance.

TABLE 5 | Detection performance comparison on general object detection datasets.

Benchmark	Method	Backbone	*AP*	*AP*$_{50}$	*AP*$_{75}$	*AP*$_s$	*AP*$_m$
PASCAL VOC	Faster R-CNN*	Resnet50	-	81.0	-	-	-
	MSR-RCNN		-	81.8	-	-	-
COCO	Faster R-CNN*	Resnet50	37.4	58.1	40.4	21.2	41.0
	MSR-RCNN		37.5	59.8	40.0	21.7	41.4

*Where * represents the method of reproduction using MMdetection.*

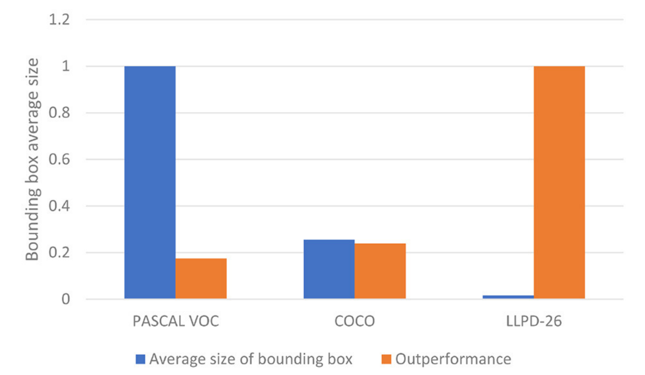


FIGURE 9 | The performance comparison between MSR-RCNN and Faster R-CNN on different datasets.

emphasize the best results for each class with bold to show the best performance. It can be found that our network outperforms other methods.

4.2.2. Ablation Experiments

4.2.2.1. Category Performance Improvement Comparison

Figure 6 shows the performance improvement of our MSR-RCNN compared with Faster R-CNN. Among them, the blue bar chart represents the size of the pest, and the line chart describes the performance improvement of the method for Faster R-CNN. Our methods (MSR and SI) mainly improve the detection performance of small-size objects. For medium-size pests, the performance of Soft-IoU is improved significantly.

4.2.2.2. The Training Loss and AP

To explain the improvement of our network in more detail, we present the training loss diagram of MSR-RCNN, Faster R-CNN, FCOS, and HGLA, as shown in **Figure 7**. Faster R-CNN represents two-stage methods, FCOS represents one-stage methods, and HGLA represents pest detection methods. Referring to the parameter setting of MMdetection, the batch size of FCOS is 4 samples, thus the loss iter only has half the other methods. It is clear that compared with other networks, our MSR-RCNN has more excellent data fitting ability and is capable of more complex work. In addition, our MSR-RCNN convergence rate is the fastest.

4.2.2.3. The Beta Value

For the β in Formula (4), an ablation study is performed and the results are shown in **Figure 8**. When the β is less than 0.9, the detector performance is affected because a large number of positive samples change into negative samples, resulting in the imbalance between positive and negative samples. When the β is greater than 0.9, the training performance of the model is misled due to the addition of too many low-quality detection boxes.

4.2.2.4. The Backbone of Our MSR Pest Detection Network

We choose ResNet50 as the backbone of the MSR-RCNN After a detailed comparison of the common backbone network. **Table 3** shows the performance comparison of our MSR-RCNN in different backbone networks. Why the result of ResNet50 is better than ResNet101? This reason is that the object size is generally small in our dataset. Therefore, with the deepening of the network layer, the features of small-size objects gradually

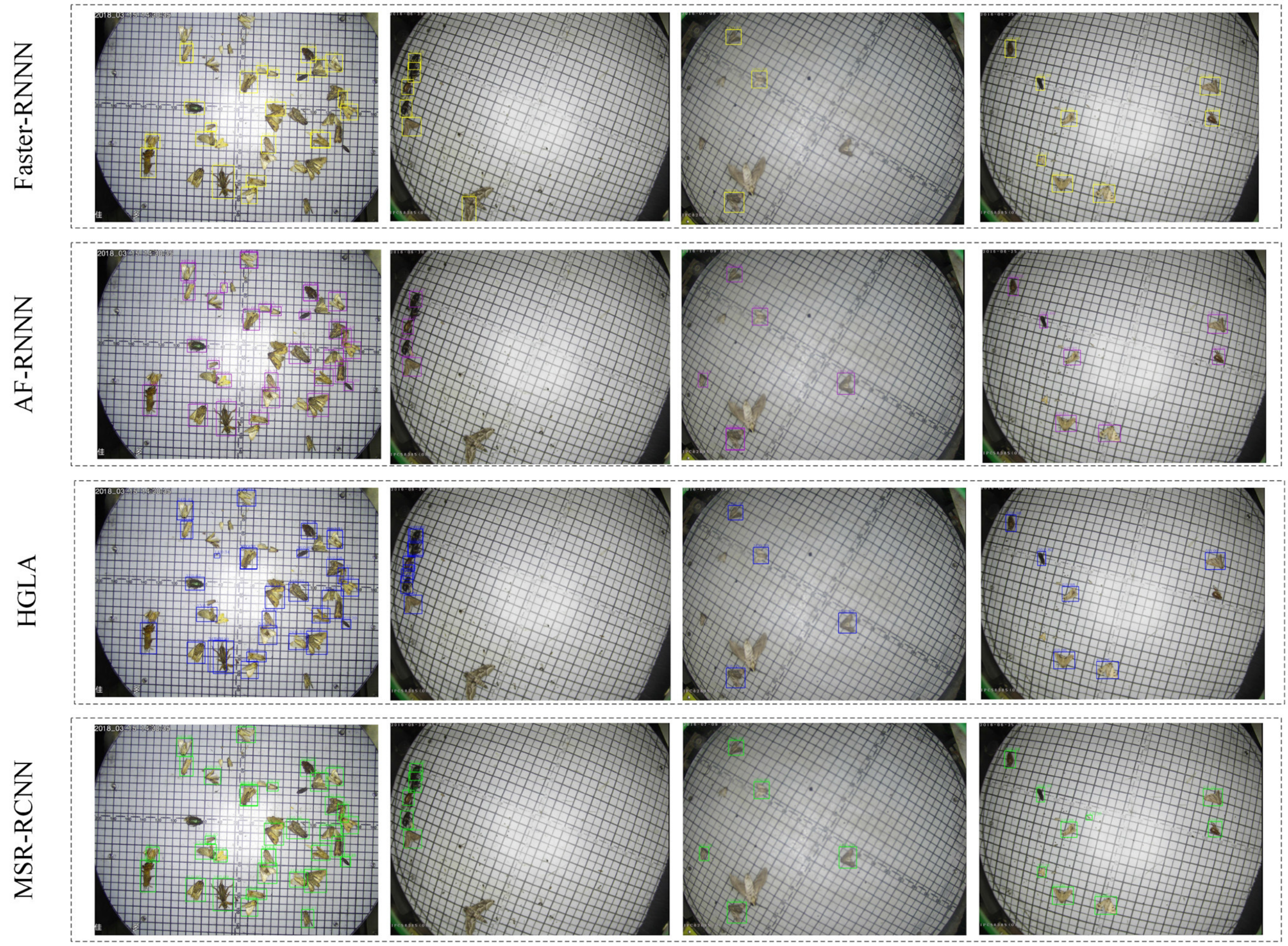


FIGURE 10 | Visualization results.

disappear in the continuous convolution operation. The top-to-down feature fusion method transmits blurry semantic information resulting in decreasing performance. To be fair, ResNet50 is used as the backbone extraction network for all comparative experiments in this study, unless otherwise stated.

4.2.2.5. MSR Module With Various Networks

We compare the performance of our MSR module with Faster R-CNN, Cascade R-CNN, FCOS, and RetinaNet, as shown in **Table 4**. The Faster R-CNN use the C4 feature map to detect pest. Due to the design of FPN (Lin et al., 2017a), all methods after 2017 use the multi-layer features detection pattern. Without bells and whistles, the MSR module effectively improves the pest detection performance under various networks. The experimental results show that the MSR module can improve the feature extraction capability and replace FPN in the pest detection field.

4.2.3. Generalization Capacity

We compare the performance on general object detection datasets (PASCAL VOC and COCO), as shown in **Table 5**. Where $*$ represents the results that we reproduced with MMDetection under the same parameter settings. Due to the Soft-IoU being designed for pest detection, we only present the performance of MSR-RCNN with the MSR module. Since MSR-RCNN is a small-size detection network for pest detection, we do not evaluate the performance of AP_l. The training set of PASCAL VOC 0712 is used to train networks and the test set of PASCAL VOC 2007 is used to verify the results. The experimental results show that our method can significantly improve the performance of IoU_{50} and small-size objects. This is highly consistent with the original intention of our MSR module.

In addition, **Figure 9** shows the performance comparison between our method and Faster R-CNN on different datasets, where the blue bar chart represents the normalized relative average size of the objects in several datasets, the yellow bar chart shows the normalized relative AP improved by our MSR-RCNN method compared to Faster R-CNN. With the increase of the object average size, the improvement of the performance becomes more and more obvious.

4.3. Qualitative Results

To visually observe the accuracy, we visualize the detection results of Faster R-CNN, AF-RCNN, HGLA, and MSR-RCNN (ours), as shown in **Figure 10**. Among them, the first column shows the dense distribution pest images, the second and fourth columns show the sparse distribution pest images, and the third column shows the image detection results when the camera has water mist caused by temperature change. The visualization shows that HGLA has many overlapped bounding boxes, AF-RCNN and Faster R-CNN mainly exhibit missed bounding boxes and false results (**Figure 10** columns 1 and 2). For columns 3 in **Figure 10** (low-quality images caused by equipment reasons), all of the detection results are degraded, but our MSR-RCNN is the least weakened. This is owed to our feature super-resolution module. Although the MSR-RCNN wrongly identifies the rice planthopper in the fourth column images (class 1 is identified as class 14), other methods did not find the existence of minimum-sized pests (**Figure 10** columns 4). The visualization results show that our MSR-RCNN outperforms other methods.

5. CONCLUSION

This study aims to bridge the gap between generic object detection and pest detection, in which the challenges lie in object characters and IoU adaptation. Therefore, we propose an MSR-RCNN that is targeted at detecting agricultural pests of 26 categories. Specifically, we build a large-scale light-trap pest dataset LLPD-26. For tackling the detection difficulty on small-size, multi-scale, and high-similarly pests, the MSR-RCNN adopts a MSR model that includes a super-resolution component, a feature fusion mechanism, and a feature weighting mechanism. In addition, motivated by the higher importance of pest positions, we propose a SI strategy to improve the adaptability of the network. The experimental results show that the proposed method can effectively detect multiple classes of pests. Ablation experiments verify the MSR model can improve the performance of other detectors in the plug-and-play form. Future study will focus on few-shot pest detection research and real-world application deployment.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary materials, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

YT contributed to the conception and design of software, analysis of the data and writing, and revising the manuscript. SD and SZ carried out compared method by using AF-RCNN and HGLA detector in experimental part. JZ and LL contributed to write and revise the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This study is supported in part by the national natural science foundation of China (no. 31671586) and the demonstration of intelligent management and control technology for the whole cycle of agricultural production (no. KFJ-STS-QYZD-167-02).

ACKNOWLEDGMENTS

The authors would like to thank Jia Duo company for providing data support. Besides, thanks to all the authors cited in this article and the referee for their helpful comments and suggestions.

REFERENCES

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

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

Cai, Z., and Vasconcelos, N. (2018). "Cascade r-cnn: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 6154–6162.

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.

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

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Ijaz, M. F., Woźniak, M., et al. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21, 4749. doi: 10.3390/s21144749

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis*. 88, 303–338. doi: 10.1007/s11263-009-0275-4

Girshick, R. (2015). "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

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* (Columbus, OH: IEEE), 580–587.

Hasan, A. M., Sohel, F., Diepeveen, D., Laga, H., and Jones, M. G. (2021). A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric*. 184, 106067. doi: 10.1016/j.compag.2021.106067

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2961–2969.

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* (Las Vegas, NV: IEEE), 770–778.

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7132–7141.

Jiao, L., Dong, S., Zhang, S., Xie, C., and Wang, H. (2020). Af-rcnn: an anchor-free convolutional neural network for multi-categories agricultural pest detection. *Comput. Electron. Agric.* 174, 105522. doi: 10.1016/j.compag.2020.105522

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

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1, 541–551. doi: 10.1162/neco.1989.1.4.541

Li, Y., Chen, Y., Wang, N., and Zhang, Z. (2019). "Scale-aware trident networks for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6054–6063.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2980–2988.

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., Xie, C., Wang, R., Yang, P., Sudirman, S., Zhang, J., et al. (2020). Deep learning based automatic multi-class wild pest monitoring approach using hybrid global and local activated features. *IEEE Trans. Ind. Inform.* 17, 7589–7598. doi: 10.1109/TII.2020.2995208

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* (Salt Lake City, UT: IEEE), 8759–8768.

Liu, Z., Gao, J., Yang, G., Zhang, H., and He, Y. (2016). Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Sci. Rep.* 6, 1–12. doi: 10.1038/srep20410

Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., and Lin, D. (2019). "Libra r-cnn: towards balanced learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 821–830.

Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., et al. (2019). "Thundernet: towards real-time generic object detection on mobile devices," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6718–6727.

Qing, Y., Jun, L., Liu, Q.-,j., Diao, G.-,q., Yang, B.-,j., Chen, H.-,m., et al. (2012). An insect imaging system to automate rice light-trap pest identification. *J. Integr. Agric.* 11, 978–985. doi: 10.1016/S2095-3119(12)60089-6

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* (Las Vegas, NV: iEEE), 779–788.

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. *arXiv [Preprint]*. arXiv:1804.02767.

Redmon, J., and Farhadi, A. (2017). "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 7263–7271.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). "Faster R-CNN: Towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39 (IEEE), 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Shen, Y., Zhou, H., Li, J., Jian, F., and Jayas, D. S. (2018). Detection of stored-grain insects using deep learning. *Comput. Electron. Agric.* 145, 319–325. doi: 10.1016/j.compag.2017.11.039

Shrivastava, A., Gupta, A., and Girshick, R. (2016). "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 761–769.

Singh, B., and Davis, L. S. (2018). "An analysis of scale invariance in object detection SNIP," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 3578–3587. doi: 10.1109/CVPR.2018.00377

Singh, B., Najibi, M., and Davis, L. S. (2018). SNIPER: Efficient multi-scale training. *arXiv [Preprint]*. arXiv:1805.09300.

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

Tian, Z., Shen, C., Chen, H., and He, T. (2019). "FCOS: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 9626–9635. doi: 10.1109/ICCV.2019.00972

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *Int. J. Comput. Vis.* 104, 154–171. doi: 10.1007/s11263-013-0620-5

Wang, J., Lin, C., Ji, L., and Liang, A. (2012). A new automatic identification system of insect images at the order level. *Knowl. Based Syst.* 33, 102–110. doi: 10.1016/j.knosys.2012.03.014

Wang, R., Zhang, J., Dong, W., Yu, J., Xie, C. J., Li, R., et al. (2017). A crop pests image classification algorithm based on deep convolutional neural network. *Telkomnika* 15, 1239–1246. doi: 10.12928/telkomnika.v15i3.5382

Wen, C., Wu, D., Hu, H., and Pan, W. (2015). Pose estimation-dependent identification method for field moth images using deep learning architecture. *Biosyst. Eng.* 136, 117–128. doi: 10.1016/j.biosystemseng.2015.06.002

Yaakob, S. N., and Jain, L. (2012). An insect classification analysis based on shape features using quality threshold artmap and moment invariant. *Appl. Intell.* 37, 12–30. doi: 10.1007/s10489-011-0310-3

Yu, F., and Koltun, V. (2015). "Multi-scale context aggregation by dilated convolutions," in *4th International Conference on Learning Representations (ICLR)*. Available online at: http://arxiv.org/abs/1511.07122

Yue, Y., Cheng, X., Zhang, D., Wu, Y., Zhao, Y., Chen, Y., et al. (2018). Deep recursive super resolution network with laplacian pyramid for better agricultural pest surveillance and detection. *Comput. Electron. Agric.* 150, 26–32. doi: 10.1016/j.compag.2018.04.004

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.

ORIGINAL RESEARCH
published: 02 June 2022
doi: 10.3389/fpls.2022.895944

Global Context-Aware-Based Deformable Residual Network Module for Precise Pest Recognition and Detection

Lin Jiao[1,2], Gaoqiang Li[1], Peng Chen[1,3]*, Rujing Wang[2,3,4]*, Jianming Du[2], Haiyun Liu[2,4] and Shifeng Dong[2,4]*

[1] National Engineering Research Center for Agro-Ecological Big Data Analysis and Application, Information Materials and Intelligent Sensing Laboratory of Anhui Province, School of Internet, Anhui University, Hefei, China, [2] Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, China, [3] Institutes of Physical Science and Information Technology, Anhui University, Hefei, China, [4] Science Island Branch, University of Science and Technology of China, Hefei, China

OPEN ACCESS

Edited by:
Jucheng Yang,
Tianjin University of Science and Technology, China

Reviewed by:
Feiniu Yuan,
Shanghai Normal University, China
Oleg Shchekochikhin,
Kostroma State Technological University, Russia

***Correspondence:**
Lin Jiao
ljiao@ahu.edu.cn
Peng Chen
pengchen@ustc.edu
Rujing Wang
rjwang@iim.ac.cn

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *14 March 2022*
Accepted: *29 April 2022*
Published: *02 June 2022*

Citation:
Jiao L, Li G, Chen P, Wang R, Du J, Liu H and Dong S (2022) Global Context-Aware-Based Deformable Residual Network Module for Precise Pest Recognition and Detection. Front. Plant Sci. 13:895944. doi: 10.3389/fpls.2022.895944

An accurate and robust pest detection and recognition scheme is an important step to enable the high quality and yield of agricultural products according to integrated pest management (IPM). Due to pose-variant, serious overlap, dense distribution, and interclass similarity of agricultural pests, the precise detection of multi-classes pest faces great challenges. In this study, an end-to-end pest detection algorithm has been proposed on the basis of deep convolutional neural networks. The detection method adopts a deformable residual network to extract pest features and a global context-aware module for obtaining region-of-interests of agricultural pests. The detection results of the proposed method are compared with the detection results of other state-of-the-art methods, for example, RetinaNet, YOLO, SSD, FPN, and Cascade RCNN modules. The experimental results show that our method can achieve an average accuracy of 77.8% on 21 categories of agricultural pests. The proposed detection algorithm can achieve 20.9 frames per second, which can satisfy real-time pest detection.

Keywords: deep learning, convolutional neural network, deformable residual network, agricultural pest, target detection

INTRODUCTION

Automatic insect recognition has attracted more and more attention in the field of agricultural engineering. Conventional pest management in farmland has relied mainly on periodic spraying plans based on schedules. With the increasing attention to environmental impact and pest control cost, integrated pest management (IPM) (Bernardo, 1993) has become one of the most effective and accurate pest management methods. It abandons the conventional spraying procedure and depends more on the actual existence or possibility of field insects. The use of insect attractants and traps is commonly adopted to monitor agricultural pest in the farmland. Growers and IPM consultants regularly monitor the pest situation of farmland by manually counting harmful insects on traps, and control agricultural pests according to specific insect distribution. However, it is time-consuming and inefficient. Therefore, automatic identification and counting of pests is the important step of IPM, which makes a major contribution for producers with large farmland.

As described in the study by Guo et al. (2021), the process of frequently used automatic recognition and counting methods can be described as follows: collecting insect pest images using trapping devices followed by automated counting *via* computer vision-based detection methods. Thus, the precise pest detection will be decided by computer vision-based detection algorithms. Wen and Guyer (2012) developed image-based orchard insect identification and classification methods by using the local features model, global features model, and the combination model, respectively. The method is more robust and can work on field insect images considering the messy background, missing insect features, and varied insect size and pose. Because each target of the sample case has different colors and distinctive body shapes, Hassan et al. (2014) proposed an automatic insect identification framework that can identify grasshoppers and butterflies by manipulating insects' color and their shape feature. Yalcin (2015) used multiple feature descriptors, i.e., Hu moment, elliptic Fourier descriptors, radial distance function, and local binary patterns, to identify and classify the insect images under complex background and illumination conditions. We know that the insect pest recognition accuracy of traditional approaches heavily depends on the hand-designed features by various algorithms. However, precise and proper features need to be carefully designed and selected for high accuracy, leading to expensive works and expert knowledge. It will be even worse when the background is complex.

Convolutional neural networks (CNNs) are effective in the fields of image recognition and classification due to the powerful ability of feature extraction. The framework of region-based CNN was developed to improve the detection accuracy (Girshick et al., 2014). CNN modules were used to automatically extract the feature representations from images, ignoring hand-crafted features. Two-stage object detection methods are the mainstream detection framework (Lin et al., 2017a; Ren et al., 2017; Cai and Vasconcelos, 2018). Specifically, the region proposal generation algorithms, such as Selective Search (Uijlings et al., 2013), EdgeBox (Zitnick and Dollár, 2014), and RPN (Ren et al., 2017), AF-RPN (Jiao et al., 2020), are applied to generate a set of region candidates (region of interests, ROIs) in the first stage, and then, these region proposals are used for obtaining multi-class labels and refining the bounding boxes using the R-CNN network. CNN-based object detection algorithms have been applied to pest detection in precision agriculture. Gomez Selvaraj et al. (2019) use Faster R-CNN detector with ResNet50, InceptionV2, and single-shot detector (SSD) with MobileNetV1 to detect banana disease and pest, and detection results show that deep CNN is a robust and easily deployable strategy for banana pest recognition. He et al. (2020) used a two-stage detection framework, Faster RCNN, to detect brown rice plant hopper, and compared it with a one-stage detection method, YOLO V3 (Redmon and Farhadi, 2018). Experimental results demonstrate that the performance of the two-stage detection algorithm significantly outperforms the one-stage detector. Wang et al. (2021) proposed a sampling-balanced region proposal network (S-RPN) and attention-based deep residual network for detecting multi-classes pests with a small size, achieving good performance compared with other state-of-the-art detectors. Jiao et al. (2020) developed a two-stage end-to-end agricultural detection method named AF-RCNN to recognize and localize multi-classes pest targets, achieving 56.4% mAP and 85.1 mRecall on a 24-types pest dataset. However, there are pose-variant, serious overlap, dense distribution, and interclass similar pests in our experimental dataset, leading to poor performance of pest feature extraction. Thus, the accurate and robust pest detection system still faces great challenges.

The hypothesis of this study is that the features of agricultural pests can be obtained by machine learning through images analysis, while they traditionally need professional knowledge of the expert. However, deep learning-based pest detection methods still face some challenges according to the aboded description. For example, there are pose-variant, serious overlap, dense distribution, and interclass similar pests in our experimental dataset, leading to poor performance of pest feature extraction. Thus, the accurate and robust pest detection system still faces great challenges. It is necessary to propose a new method to address the precise recognition of pest with pose-variant, serious overlap, dense distribution, and interclass similar pests. A deep CNN is applied to automatically extract rich feature information from pest images with multi-pose, high similarity, and high overlap. A feature extractor module is used to enhance the features of region-of-interest of pest by merging the global information of pest image. The objectives of this work are to (1) develop a deformable residual block (DRB) network to extract detailed feature information of multi-class pest with pose-variant, serious overlap, dense distribution, and interclass similar pests; (2) propose a global context-aware module to get high-quality feature of region-of-interests of pests; and (3) introduce an end-to-end two-stage pest detection algorithm to accomplish the identification and detection of 21-types of agricultural pest.

MATERIALS AND METHODS

In this part, the whole framework of our agricultural pest detection network is first demonstrated. Second, the materials used in this study are presented. Third, the proposed DRB network (DRB-Net) is described in detail. Finally, the region proposal generation algorithm and the global context-aware feature extraction module are introduced, respectively.

Agricultural Pest Detection Framework

In this part, the overview of the whole detection framework is shown in **Figure 1**. A pest collection equipment is used to obtain a large number of pest images and then these pest images are labeled by professional experts. Pest images are input into DRB-Net for extracting deformable feature information, and feature pyramid network (FPN) is applied to extract multi-scale fusion pest features. These extracted features are input to region proposal network (RPN) to generate a set of pest proposals, and then a global context-aware feature (GCF) extractor is developed to produce region-of-interest (RoI) with global context information. Following R-CNN (Girshick et al., 2014), two-stage

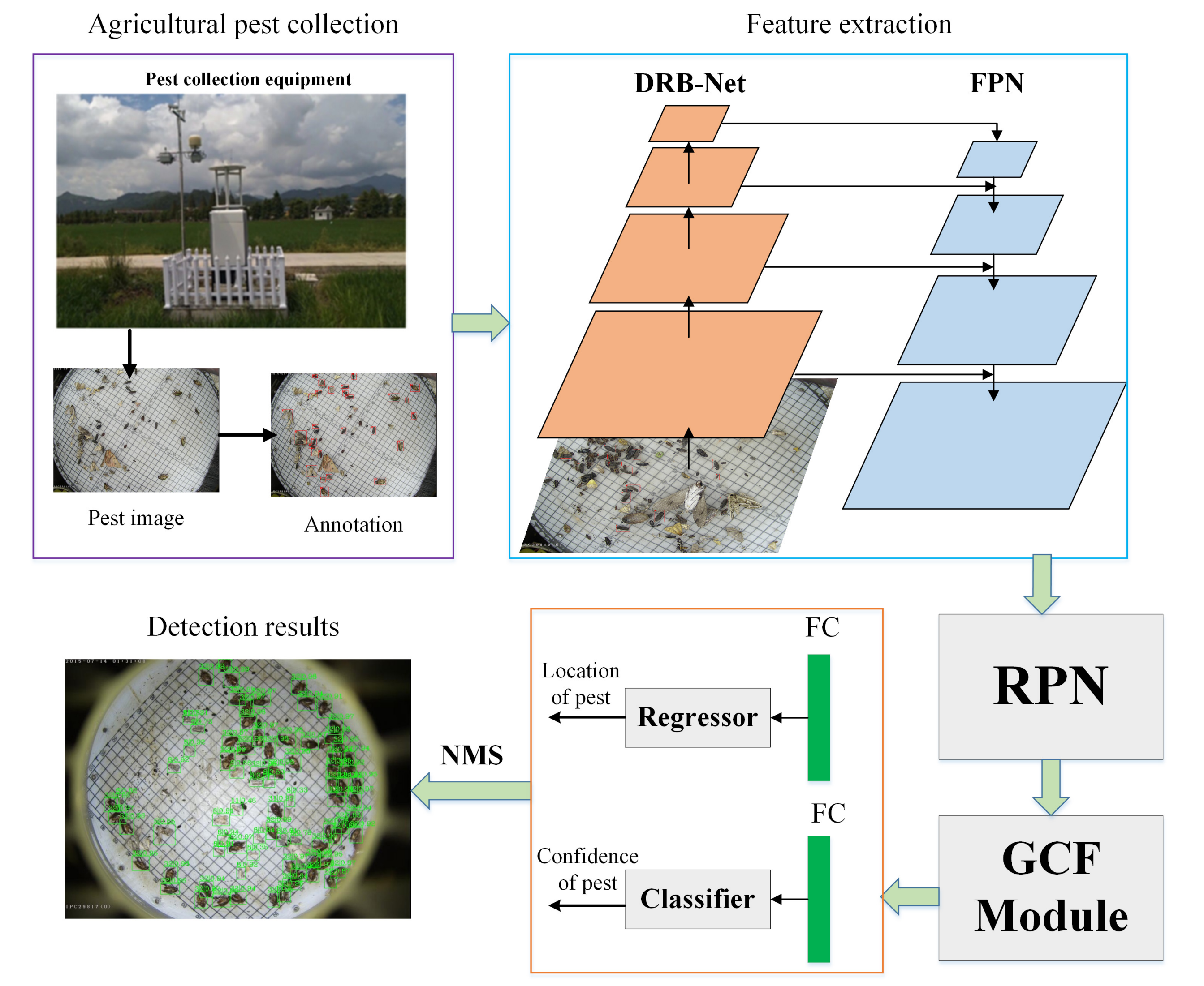


FIGURE 1 | Whole framework of agricultural pest detection. FC represents the fully connected layer.

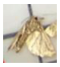
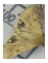
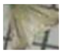
CNNs are used for specific-class classification and localization of each RoI *via* an end-to-end way. Finally, the NMS (Non-Maximum Suppression) algorithm (Rosenfeld and Thurston, 1971) is adopted to filter redundant bounding boxes, and obtain pest detection results.

Materials

In this study, the experimental images are collected by an automatic device that uses a multispectral light trap for attracting crop pests. HD camera above the tray of this device is set to take images, which were saved in a JPG format with $2,592 \times 1,944$ pixels. In this work, the width and height of the pest images are resized to 800×600 for high efficiency. The dataset contains 24,412 images and 21 types of pests. **Table 1** shows details of our collected agricultural pest dataset, including the scientific names, the pest images, the number of pest instances and pest images, and the average relative scale of each pest instance. In order to train and evaluate the performance of the CNN-based objector, all pest images are randomly split into train set (15,378 images), validation set (6,592 images), and test set (2,442 images), respectively.

To recognize the object of an image using deep CNN, the class and localization of each pest instance needs to be labeled. In this study, these pest instances are hand-annotated by several pest experts using LabelImg software, which is provided by the Computer Science and Artificial Intelligence Laboratory at MIT. Generally, rectangular bounding boxes are used to annotate the location of a pest instance, which can be represented as (x_1, y_1, x_2, y_2), here (x_1, y_1) is the coordinate of top-left and (x_2, y_2) is the coordinate of bottom-right. **Figure 2** shows some examples of agricultural pest images. Pose variations of the same types of pest will decrease the precise recognition, as presented in **Figure 2A**. Besides, the distribution of pest targets is seriously dense and worse is that the pest targets are overlapped, as shown

TABLE 1 | Details of 21 types of agricultural pest, including the pest images, number of pest instances of each category, number of pest images of each category, and the average relative scale of each pest instance.

Classes	Image	Number of instances	Number of pest images	Average relative scale (%)
Cnaphalocrocis medinalis (CM)		1,224	932	0.1214
Chilo suppressalis (CS)		1,285	454	0.1793
Mythimna separate (MS)		8,374	3,637	0.3978
Helicoverpa armigera (HA)		26,588	8,740	0.2814
Pyrausta nubilalis (PN)		15,739	5,294	0.2267
Athetis lepigone (AL)		28,932	7,200	0.1298
Spodoptera litura (SL)		1,896	1,543	0.4572
Spodoptera exigua (SE)		7,116	3,527	0.1377
Sesamia inferen (SI)		1,768	1,335	0.2776
Agrotis ypsilon (AY)		3,890	2,314	0.5703
Mamestra brassicae Linnaeus (MbL)		2,170	1,632	0.4259
Scotogramma trifolii Rottemberg (StR)		4,393	3,051	0.2816
Agrotis segetum (AS)		1,615	1,330	0.4024
Agrotis tokionis Butle (AtB)		465	351	0.6375
Holotrichia oblita Faldermann (HoF)		82	70	0.3348
Holotrichia parallela (HP)		11,325	3,002	0.2518
Anomala corpulenta (AC)		52,134	5,083	0.2466
Gryllotalpa orientalis Burmeister (GoB)		6,480	3,589	0.9530
Pleonomus canaliculatus (PC)		157	109	0.3281
Agriotes subrittatus Motschulsky (AsM)		6,161	1,729	0.1129
Melanotus caudex Lewis (McL)		677	224	0.1584

in **Figures 2B,C**, respectively. The appearance of two different categories of pest has a high similarity, for example, the class "HA" and "MS," as shown in **Figure 2D**.

Deformable Residual Block Network

As we know, a deep residual network is a common backbone for extracting features. For ResNet50 (He et al., 2016), it contains 16 residual blocks with 50 convolutional layers. The output feature map of each residual block in ResNet50 network has different resolutions. The details of the ResNet50 are reported in **Table 2**. For the same class pest instances with different poses and shapes, the common backbone cannot effectively extract the feature information of pest, leading to poor recognition of pest with different shapes and poses.

Inspired by previous work (Dai et al., 2017), it is known that deformable convolution can enhance the capability of CNNs of modeling geometric transformation of objects. The difference between traditional convolution and deformable convolution can be shown in **Figure 3**. It shows that the sampling locations of deformable convolution are irregular compared with the regular sampling of traditional convolution.

Additionally, from the aspect of mathematical description, the standard convolution can be defined as following:

$$y\left(p_0\right) \ = \sum\nolimits_{p_n \in R} w\left(p_n\right).x(p_0 + p_n) \tag{1}$$

where $y\left(p_0\right)$ denotes the output feature map for each location p_0; $\mathcal{R}$ represents the sampling space in the input feature map x; w is the learnable weight; p_n enumerates the location of sampling space R.

However, in deformable convolution, the sampling space is enlarged by adding the offsets, which can be defined by Equation (2):

$$y\left(p_0\right) = \sum\nolimits_{p_n \in R} w\left(p_n\right).x(p_0 + p_n + \triangle p_n) \tag{2}$$

where $\triangle p_n$ denotes the offset, which can be obtained by network learning. However, $\triangle p_n$ is typically fractional. The bilinear interpolation operation is used for obtaining the final offsets.

Therefore, to detect pose-invariant and shape-invariant pest instances, a deformable convolution module has been embedded into the deep residual network, which can extract multi-scale and deformable pest features. The architecture of DRB is presented in **Figure 4**. The deformable module is designed for extracting shape information of pest. Finally, the DRB is introduced into the residual blocks of ResNet50 backbone, achieving the effective extraction of deep deformable pest feature information.

As we know that low-level features usually have large spatial size and more-grained detail information, while high-level features tend to contain more semantic knowledge. Generally, low-level features are beneficial for the detection of small objects. To identify pest with different sizes, a multi-scale feature extraction network, i.e., FPN (Lin et al., 2017a) is adopted to fuse pest feature information from low-level and high-level feature maps.

Generation of Pest Region Proposal

In Faster RCNN (Ren et al., 2017), Ren et al. (2017) proposed the RPN to generate a set of region proposals. This region proposal is the region that contains the object instance. As shown in **Figure 5**, RPN model consists of two fully connected layers: classification

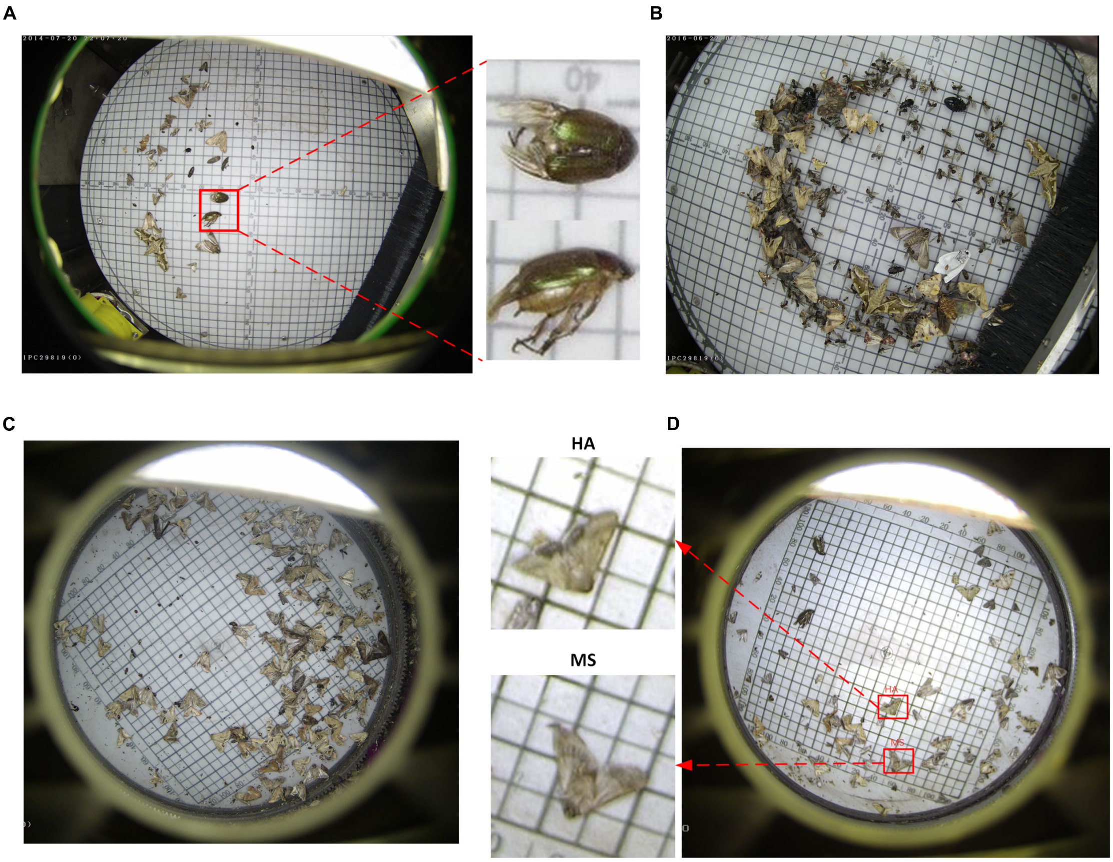


FIGURE 2 | Some examples of agricultural pest images. **(A)** Different shapes of the same class of pests. **(B)** Serious overlap. **(C)** Dense distribution. **(D)** High similarity between the classes "HA" and "MS."

layer and regression layer. The former outputs $2k$-dimension vector encoding the classification confidence (objects or not objects), and the latter outputs $4k$-dimension vector encoding the coordinates of bounding box. In this study, k denotes the number of anchor boxes in RPN. The parameter k is set to 1, leading to fewer parameters of RPN and improving the efficiency without decreasing the quality of pest region proposals. The stochastic gradient descent (SGD) (LeCun et al., 1989) method was used for end-to-end training, which allowed the convolutional layers to be shared between the RPN and the Fast R-CNN components. The feature maps from deformable FPN are propagated forward to pest proposal generation network, and then a set of pest proposals with classification scores and coordinates of bounding boxes is received as output.

TABLE 2 | Description of standard ResNet50.

Layer name	Setting of convolutional layers
Conv1	7 × 7, 64, stride 2
	3 × 3 max pool, stride 2
Conv2_x (block 1)	$\begin{bmatrix} 1\times 1, & 64 \\ 3\times 3, & 64 \\ 1\times 1, & 256 \end{bmatrix} \times 3$
Conv3_x (block 2)	$\begin{bmatrix} 1\times 1, & 128 \\ 3\times 3, & 128 \\ 1\times 1, & 512 \end{bmatrix} \times 4$
Conv4_x (block 3)	$\begin{bmatrix} 1\times 1, & 256 \\ 3\times 3, & 256 \\ 1\times 1, & 1,024 \end{bmatrix} \times 6$
Conv5_x (block 4)	$\begin{bmatrix} 1\times 1, & 512 \\ 3\times 3, & 512 \\ 1\times 1, & 2,048 \end{bmatrix} \times 3$
	Average pooling, 7 × 7, stride 1

However, these pest proposals may be reductant and of low quality. Generally, the NMS algorithm is adopted to decrease the overlapped bounding box candidates and improve the quality. Given a series of proposals with classification scores in an image, the IoU ratios between the bounding box with the highest score and its neighboring bounding boxes are calculated. The scores of neighboring bounding boxes will be suppressed when their IoU ratios are lower than the preset values. The process of NMS can be described mathematically as Equations (3 and 4):

$$s_i = \begin{cases} s_i \; IoU(B, b_i) < t \\ 0 \; IoU(B, b_i) \geq t \end{cases} \tag{3}$$

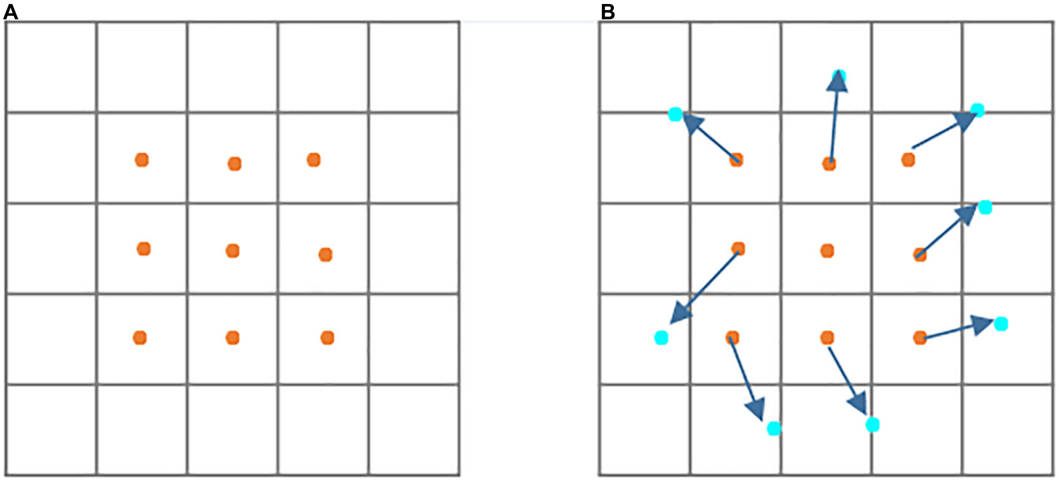


FIGURE 3 | Illustration of sampling location of traditional and deformable convolutions. **(A)** Regular sampling of traditional convolution. **(B)** Irregular sampling (indicated in deep blue arrows) of deformable convolution.

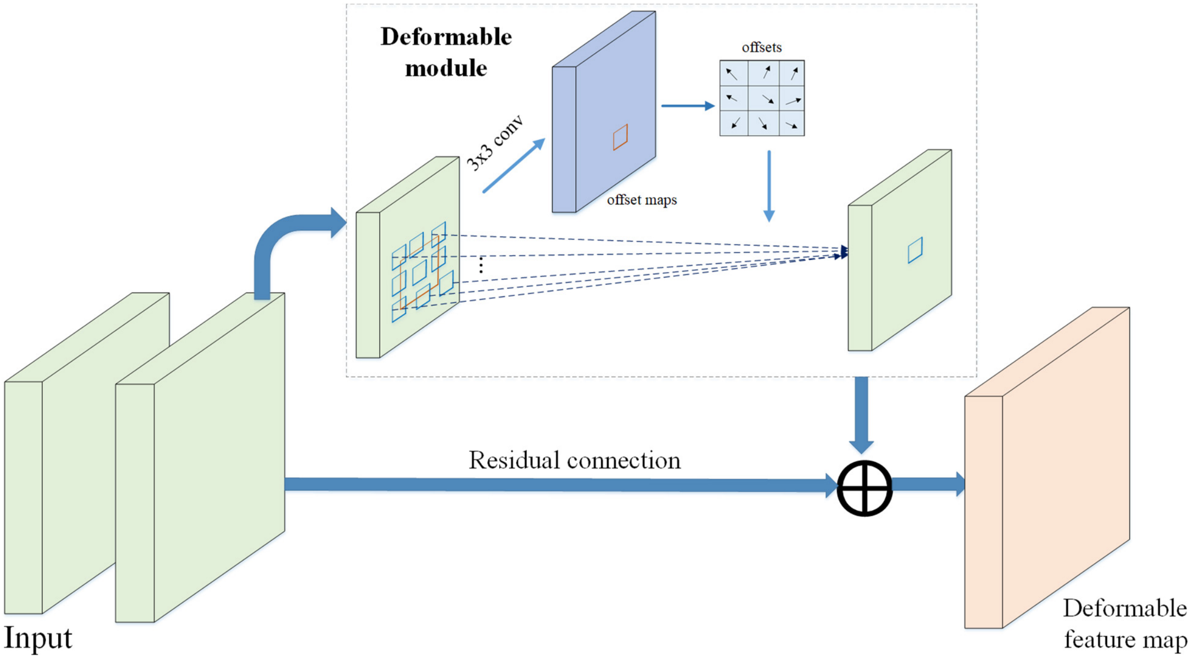


FIGURE 4 | Architecture of the deformable residual block.

$$IoU(B, b_i) = \frac{area(B \bigcap b_i)}{area(B \bigcup b_i)} \tag{4}$$

where B is the bounding boxes with the highest score, b_i represent the i-th neighboring boxes of B with confidence score S_i. t is the threshold value of IoU ratio, which is set to 0.7; area(B∩b_i) denotes the intersection of boxes with the highest scores and their neighboring boxes, and area(B∩b_i) is their union.

The low-quality bounding box candidates can be removed using the NMS algorithm. Notably, a different number of region proposals are used during training and testing. In our study, 1,000 proposals are selected according to their scores for network training and testing. Besides, the effect of different numbers of proposals is explored in the section of experiments.

Global-Context Feature Module

For the challenging scenarios in agricultural pest detection, such as cluttered background, foreground disturbance, simple integration of high-level, and low-level features may fail to detect the pest targets due to lacking the global context. A global context-aware feature module is designed in this work to extract rich information of agricultural pest, as shown in **Figure 6**. Given the full-image convolutional feature map in the FPN, the feature maps are pooled by global pooling, which can be implemented by an adaptive average pooling using the entire image's bounding box as the RoI. The pooled features are input into the post-RoI layer to get a global context pest feature. And the global feature is concatenated with the local RoI feature developed by RoI pooling. Therefore, additional global context information is accessible for each pest proposal, improving the recognition and localization of pest under complex scenes.

Unified Pest Detection Network

To detect the multi-categories pest, the RPN (Ren et al., 2017) and Fast R-CNN (Girshick, 2015) module are combined into a single network *via* an end-to-end way, as shown in **Figure 1**. These two networks can be separately trained. However, the separate training will lead to different convolutional layers. Therefore, according to the training procedure in Ren et al. (2017), joint training between RPN and R-CNN was performed, which allows for shared convolutional layers. In each SGD iteration, the forward pass generates pest proposals, which are then fed into the Fast R-CNN detector for training. The backward propagation happens as usual, and for the sharing convolutional layer, the backward propagated signals come from the combination of RPN losses and Fast R-CNN losses. Additionally, another advantage of the end-to-end training method is that it can reduce the training time compared with the separate training model.

Evaluation Metrics

To verify the performance of our proposed agricultural pest detection method, the metrics of average precision (AP) and recall are adopted. A true positive (TP) is when the network correctly identifies the pest target. A predicted box is viewed as

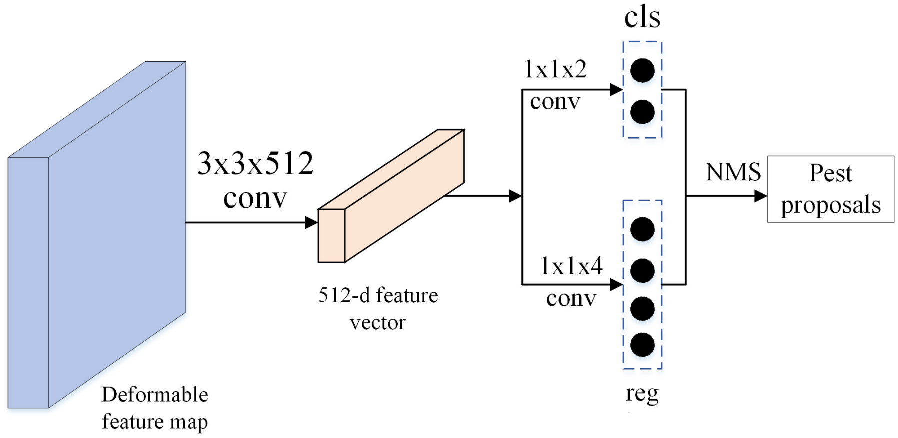


FIGURE 5 | Network structure of pest region proposal generation module.

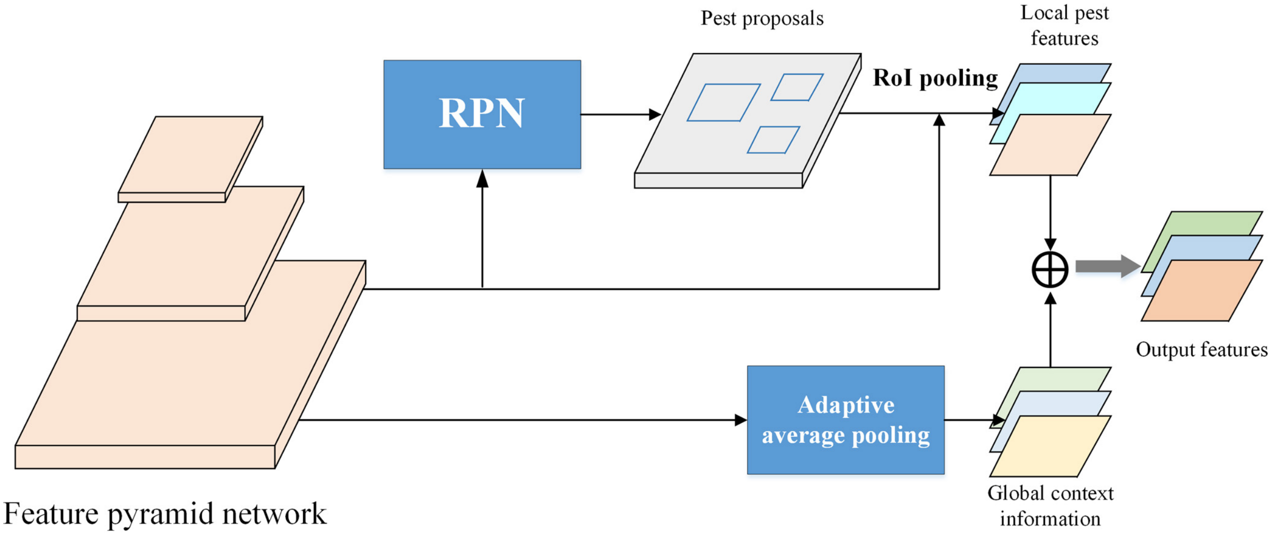


FIGURE 6 | Description of global context-aware feature module.

false positive (FP) when the model falsely identifies a pest target, for example, calling something an "*Agrotis ypsilon*" that is not an "*Agrotis ypsilon*." The precision (P) and recall (R) are defined as follows:

$$precision = \frac{\#TP}{\#TP + \#FP} \tag{5}$$

$$recall = \frac{\#TP}{GT} \tag{6}$$

In which $\#TP$, $\#FP$ present the number of TP and FP, respectively. Ground Truth (GT) denotes the total number of ground truth boxes.

The average precision (AP) can be calculated based on the shape of the precision/recall curve.

$$AP = \int_0^1 PdR \tag{7}$$

The mean AP (mAP) averaged over all object classes is employed as the final measure to compare performance over all object classes, and it is defined as follows:

$$mAP = \frac{1}{C}\sum\nolimits_{j=1}^{C} AP_j \tag{8}$$

where C is the number of classes, which is 21 in this study.

Additionally, the AP0.75 denotes the AP at IoU 0.75, which is applied to evaluate the detection accuracy of pest detection. The strict metrics, for example, mean AP and Average recall (AR) across IoU thresholds from 0.5 to 0.95 with an interval of 0.05, are used to further verify the performance of the proposed method. ARs, Arm, and ARl is the average recall of small, medium, and large pest target, respectively. In this study, the small, medium, and large pest target can be defined in **Table 3**.

TABLE 3 | Definition of the small, medium, and large pests.

	Min rectangle area (pixel)	Max rectangle area (pixel)
Small pest	0 × 0	32 × 32
Medium pest	32 × 32	96 × 96
Large pest	96 × 96	∞ × ∞

TABLE 4 | Detection results (AP) compared with other methods on pest dataset (unit: %).

	Method						
Class	SSD	YOLOv3	RetinaNet	FPN	YOLOF	Cascade RCNN	Our method
CM	68.7	64.7	68.2	70.0	63.9	69.6	**78.1**
CS	69.7	73.6	73.1	74.7	71.6	76.5	**80.0**
MS	79.7	77.3	75.3	82.3	79.6	82.0	**85.4**
HA	91.1	87.1	88.1	90.5	88.8	90.3	**91.6**
PN	79.6	77.0	76.7	82.0	79.9	82.7	**85.4**
AL	72.0	69.3	62.8	74.7	72.7	73.8	**78.9**
SL	81.4	73.8	78.3	83.2	81.8	84.2	**85.9**
SE	53.1	47.1	48.1	57.2	53.8	56.2	**64.9**
SI	77.1	73.0	79.1	82.6	76.9	81.5	**85.2**
AY	89.2	84.5	83.7	89.2	86.8	89.2	**91.4**
MbL	66.9	54.3	57.6	67.8	64.5	69.7	**77.0**
StR	58.2	55.6	52.5	61.4	55.8	59.9	**68.8**
AS	63.8	53.8	42.5	60.9	46.1	58.8	**68.2**
AtB	60.0	48.0	44.7	53.1	60.3	53.8	**64.0**
HoF	3.0	0.0	7.3	4.2	0.0	0.0	**16.8**
HP	93.0	89.4	87.8	90.8	88.8	90.8	**92.1**
AC	95.8	89.1	89.3	90.7	88.4	90.7	**91.6**
GoB	97.3	97.5	98.2	97.5	98.4	97.6	**97.6**
PC	54.2	44.1	43.4	53.1	42.0	52.7	**56.7**
AsM	79.0	81.6	75.2	81.9	74.9	82.0	**86.5**
McL	74.7	83.2	27.6	74.0	73.3	76.8	**87.5**
Average	71.8	67.8	64.7	72.5	70.0	72.3	**77.8**

The detection results of our method are shown in bold.

EXPERIMENTAL RESULTS AND ANALYSIS

Experimental Details

The proposed method and other state-of-the-art models are trained using the back-prorogation algorithm and SGD method, with momentum 0.9 and initialize learning rate to 0.0025 that will be dropped by 10 at the 8-th and 11-th epoch followed by Ren et al. (2017). The batch size is set to 4 during training. The proposed detection module is trained *via* an end-to-end way. These experiments are performed on a dell T3630 computer workstation with NVIDIA TITANX, 24G graphics card, and Intel core i9-9900K. Deep CNN was built based on Pytorch framework under Ubuntu18.02 operating system.

Comparison Results of Each Category of Agricultural Pest

Table 4 reports the detection results. It presents the AP of 21 pest classes performed by our method and other state-of-the-art models. **Table 4** suggests that that our method can achieve more precise recognition accuracy on all the categories. It is obvious that the proposed method significantly outperforms one-stage detectors, for example, 6.0% improvements for SSD (Liu et al., 2016), 10.0% improvements for YOLO (Redmon and Farhadi, 2018), and 13.1% improvements for RetinaNet (Lin et al., 2017b), and 7.8% inprovements for YOLOF (Chen et al., 2021). Additionally, the detection accuracy of our method is also higher than the multi-stage methods [e.g., FPN (Lin et al., 2017a) and Cascade RCNN (Cai and Vasconcelos, 2018)]. Specifically, it improves 5.3 points and 5.5 points compared with FPN and Faster RCNN, respectively.

However, **Table 4** also shows that the detection accuracy of the pest "HoF" is only 16.3%, which largely falls behind other categories of pests with adequate samples. This is because the number of samples of the pest "HoF" is only 70, leading to insufficient learning during network training. Therefore, the number of pest samples will significantly affect the detection results.

Table 4 summarizes that the "HoF" seems to be difficult to recognize on all detection models, while all the models could classify the "HA" pest. The proposed method can achieve 16.8% AP, obviously outperforming other methods. Especially, for the YOLO and Cascade RCNN detectors, the detection accuracy is 0.0%, which does not recognize this class of pests. The improvement of our method contributes to the introduction of the deformable residual network and global feature extractor, which can extract rich global pest features in deformed pest images.

Compared Results Evaluated by Strict Metrics

The stricter standards (e.g., AP0.5:0.9, AP0.75, and AR) are applied to evaluate the detection results. The AR is used to evaluate the localization accuracy of pest targets, and ARs, ARm, and ARl are the AR of small, medium, and large-scale pest, respectively. **Table 5** shows the compared detection results among SSD (Liu et al., 2016), RetinaNet (Lin et al., 2017b), YOLO (Redmon and Farhadi, 2018), Cascade RCNN (Cai and Vasconcelos, 2018), FPN (Lin et al., 2017a), YOLOF(Chen et al., 2021), and the proposed method. It is observed from **Table 5** that AP @IoU [0.5:0.95] and AP@IoU = 0.75 of our method can achieve 49.6 and 58.8%, respectively, outperforming other state-of-the-art detectors. This demonstrates that our method can not only improve the accuracy of classification but also localization.

Ablation Experiments

The proposed pest detection method has contributed two elements, including global-context feature (GCF) module and deformable residual block network (DRB-Net). To analyze the contribution of each component, the ablation experiments are shown in **Table 6**. In this study, the baseline is Faster R-CNN with FPN. We first add the GCF module to the baseline, as shown in the second row of **Table 6**. The DRB-Net leads to a gain of 2.5% AP. This is because of the addition of global context information, which is instrumental in the recognition of crop pest. The third row of **Table 6** demonstrates that the DRB-Net can effectively boost the performance from 75.0 to 76.6%. The improvements may be result from the extraction of agricultural pest with various scales and poses. Finally, we analyze the influence of multi-scale training. From the fourth row of **Table 6**, we can observe that multi-scale training can improve the accuracy of pest detection. This is because the multi-scale training enhances the diversity of training samples.

TABLE 5 | Compared results evaluated by strict evaluation criteria.

Method	SSD	RetinaNet	YOLOv3	Cascade RCNN	YOLOF	FPN	Proposed method
AP0.5:0.9	44.2	41.2	39.6	46.4	42.1	45.9	49.6
AP0.75	51.4	48.4	42.3	54.9	47.3	53.7	58.8
AR	61.3	61.5	51.3	58.0	58.3	59.3	62.0
ARs	47.7	51.6	40.2	43.5	48.1	45.3	51.1
ARm	64.0	65.6	53.9	60.1	61.2	63.0	61.9
ARl	45.0	45.0	50.0	30.0	35.0	35.0	50.0

Detection Efficiency

Aside from detection accuracy, the detection speed also needs to be considered. **Table 7** reports the results of the detection speed of the proposed method and other excellent detection models. The proposed model can run at a speed of 20.9 FPS, which outperforms Cascade RCNN (Cai and Vasconcelos, 2018). However, it underperforms other detection models, such as SSD (Liu et al., 2016), RetinaNet (Lin et al., 2017b), and YOLOv3 (Redmon and Farhadi, 2018). This is because the proposed pest detection network is a two-stage framework that uses RPN for generating pest proposals, leading to consumption of time. But one-stage detection models are proposal-free, directly regressing the bounding box of pest and classifying, resulting in higher efficiency. In summary, the precision of our method is higher than other methods, and the detection speed could satisfy the requirement of real-time detection; therefore, our method balances the pest detection efficiency and accuracy.

Analysis Experiments of Pest Proposals

As we know that the quality of pest proposals will decide the final detection accuracy of agricultural pest,

TABLE 6 | Ablation study on the major components.

GCF module	DRB-Net	Multi-scale training	mAP (%)
			72.5
✓			75.0
✓	✓		76.6
✓	✓	✓	77.8

TABLE 7 | Detection efficiency of agricultural pest using our method and other state-of-the-art models.

Method	Efficiency (FPS)	Accuracy
SSD	41.1	71.8
RetinaNet	21.4	64.7
YOLOv3	54.7	67.8
YOLOF	35.7	70.0
Cascade RCNN	17.2	72.3
FPN	22.0	72.5
Proposed method	20.9	77.8

TABLE 8 | Recalls of different number of pest region proposals generated by RPN with DRB-Net and without DRB-Net.

Number of proposals	10	50	100	1,000
With DRB-Net	55.1	89.0	95.2	95.2
Without DRB-Net	54.4	87.6	93.8	93.8

TABLE 9 | Recalls of pest proposals generated from RPN without DRB-Net and with under different IoU thresholds.

IoU thresholds	0.5	0.6	0.7	0.8	0.9
Without DRB-Net	93.8	92.5	85.5	58.7	8.4
With DRB-Net	95.2	94.1	87.7	61.7	13.3

Table 8 lists the recall of different numbers of pest proposals produced by RPN without and with DRB-Net. It shows that the quality is higher when using DRB-Net. For example, when using 50 proposals, the RPN with DRB-Net can achieve 89.0% recall, which obtains 1.4% improvements compared with RPN without DRB-Net. Thus, the introduction of DRB-Net contributes to the improvement of agricultural pest detection.

From the view of localization of pest, **Table 9** shows the recalls of pest proposal produced from RPN with and without DRB-Net under different IoU thresholds while using 100 proposals. It demonstrates that the performance of RPN with DRB-Net outperforms that without using DRB-Net. With the increase of IoU, the recalls of pest proposals will gradually decrease; however, the recall of RPN with DRB-Net can achieve 13.3, obtaining 4.9% improvements than without DRB-Net. This phenomenon suggests that the DRB-Net is the main factor to promote the quality of pest proposals.

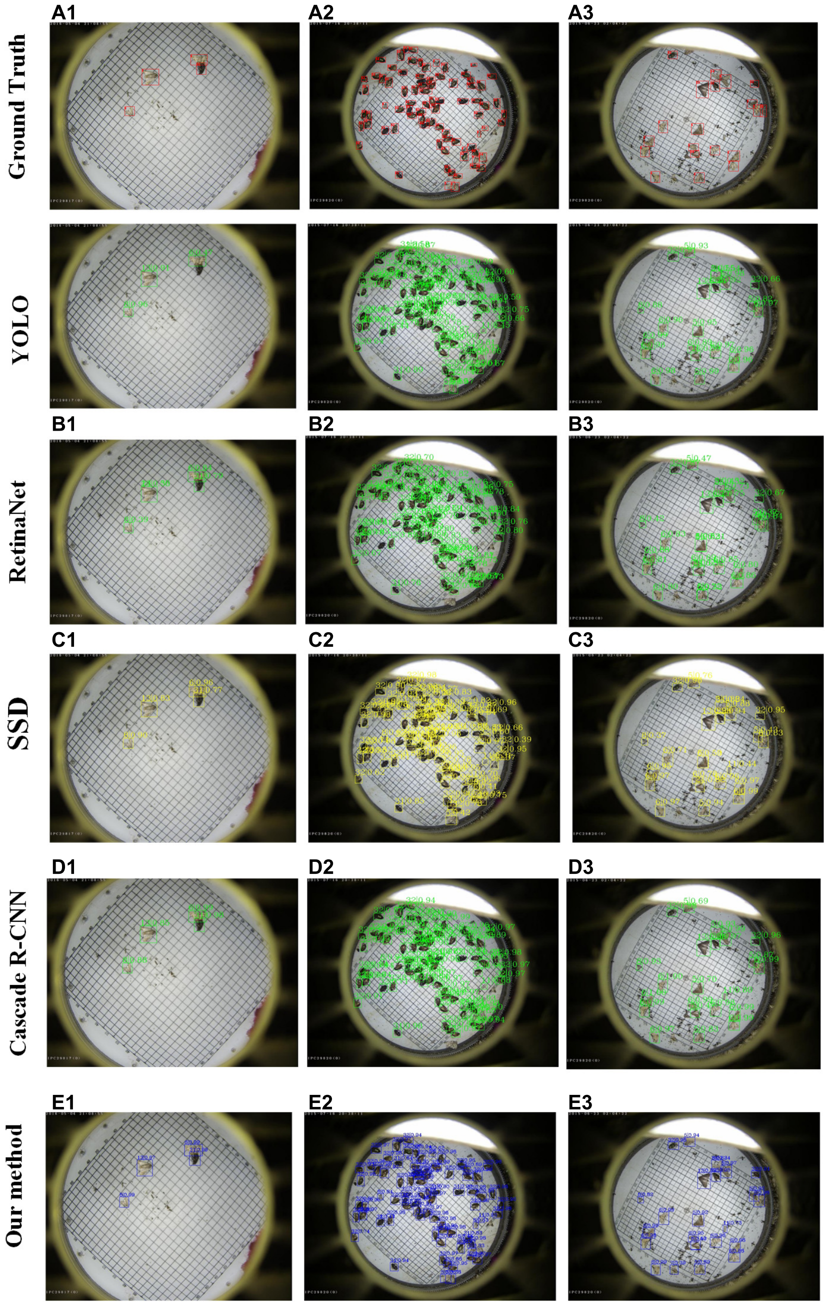


FIGURE 7 | Selected examples of agricultural pest detection results by using YOLO, RetinaNet, SSD, Cascade R-CNN, and our method.

Visualization of Agricultural Pest Detection Results

For visualization purpose, several examples of pest detection results are given in **Figure 7**. The row from the top to the bottom is expressed as the result of Ground truth, YOLO, RetinaNet, SSD, Cascade R-CNN, and our method. The detection results are marked by boxes with different colors. The proposed method could obtain good performance on the pest targets with sparse and dense distribution. For example, the class "HP" is undetected by using YOLO version 3 algorithm, as shown in **Figure 7** (a1), while the recognition accuracy can achieve 99.0% for the proposed method, as shown in **Figure 7** (d1). Additionally, for pest targets with dense distribution, our method has a higher precision of classification than other methods.

CONCLUSION

As we know, insect pests are one of the main factors affecting agricultural product yield. Precise recognition and localization of insect pests benefit to timely preventive measures to decrease economic losses. However, recent pest detection methods cannot effectively recognize and localize the pest targets. In this study, a deformable residual network is developed to extract deformable feature information of crop pest. Furthermore, a global context-aware extractor is designed to obtain global features of pest images, which are combined with local features, contributing to the improvement of the detection of pest targets. Quantitative experiments were conducted on the constructed large-scale multi-class pest dataset to evaluate the performance of the proposed method, demonstrating that the proposed method outperforms other state-of-the-art detectors in the view of pest localization and classification.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

LJ: conceptualization, methodology, software, investigation, and writing draft. GL: validation, formal analysis, visualization, and software. PC: validation and revised the manuscript. JD, RW, HL, and SD: writing and revising. All authors contributed to the article and approved the submitted version.

FUNDING

This study was supported by the Natural Science Foundation of Anhui Higher Education Institutions of China (No. KJ2021A0025), the National Natural Science Foundation of China (No. 62072002), the Major Special Science and Technology Project of Anhui Province (202003A06020016), and the Special Fund for Anhui Agriculture Research System.

REFERENCES

Bernardo, E. N. (1993). Adoption of the integrated pest management (IPM) approach in crop protection: a researcher's view. *Philipp. Entomol.* 9, 175–185.

Cai, Z., and Vasconcelos, N. (2018). "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: CVPR). 2575–7075. doi: 10.1109/CVPR.2018.00644

Chen, Q., Wang, Y. M., Yang, T., Zhang, X. Y., Cheng, J., and Sun, J. (2021). "You Only Look One-level Feature," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Electr Network* (Salt Lake City, UT: CVPR). 13034–13043.

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: ICCV). 764–773. doi: 10.1109/ICCV.2017.89

Girshick, R. (2015). "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: ICCV). 1440–1448. doi: 10.1109/ICCV.2015.169

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* (Salt Lake City, UT: CVPR). 580–587. doi: 10.1109/CVPR.2014.81

Gomez Selvaraj, M., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W., et al. (2019). AI-powered banana diseases and pest detection. *Plant Methods* 15:92. doi: 10.1186/s13007-019-0475-z

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* 12:705. doi: 10.3390/insects12080705

Hassan, S., Yusof, Z. H., and Shoon, L. W. (2014). Automatic classification of insects using color-based and shape-based descriptors. *Int. J. Appl. Control Electr. Electron. Eng.* 2, 23–35.

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* (Salt Lake City, UT: CVPR). 770–778. doi: 10.1109/CVPR.2016.90

He, Y., Zhou, Z. Y., Tian, L. H., Liu, Y. F., and Luo, X. W. (2020). Brown rice planthopper (*Nilaparvata lugens Stal*) detection based on deep learning. *Precis. Agric.* 21, 1385–1402. doi: 10.1007/s11119-020-09726-2

Jiao, L., Dong, S. F., Zhang, S. Y., Xie, C. J., and Wang, H. Q. (2020). AF-RCNN: an anchor-free convolutional neural network for multi-categories agricultural pest detection. *Comput. Electron. Agric.* 174:105522. doi: 10.1016/j.compag.2020.105522

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1, 541–551. doi: 10.1162/neco.1989.1.4.541

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: CVPR). 2117–2125. doi: 10.1109/CVPR.2017.106

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017b). "Focal loss for dense object detection," in *Proceedings of International Conference on Computer Vision* (Venice: ICCV). 2999–3007. doi: 10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). "SSD: single shot multibox detector," in *Proceedings of European Conference on Computer Vision* (Berlin: Springer). 21–37. doi: 10.1007/978-3-319-46448-0_2

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

Ren, S. Q., He, K. M., Girshick, R., and Sun, J. (2017). "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39 (New Jersy, NJ: IEEE). 1137–1149. doi: 10.1109/tpami.2016.2577031

Rosenfeld, A., and Thurston, M. J. I. T. C. (1971). Edge and curve detection for visual scene analysis. *IEEE Trans. Comput.* 20, 562–569.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M. (2013). Selective search for object recognition. *Int. J. Comput. Vis.* 104, 154–171. doi: 10.1007/s11263-013-0620-5

Wang, R., Jiao, L., Xie, C., Chen, P., Du, J., and Li, R. (2021). S-RPN: sampling-balanced region proposal network for small crop pest detection. *Comput. Electron. Agric.* 187:106290. doi: 10.1016/j.compag.2021.106290

Wen, C., and Guyer, D. (2012). Image-based orchard insect automated identification and classification method. *Comput. Electron. Agric.* 89, 110–115. doi: 10.1016/j.compag.2012.08.008

Yalcin, H. (2015). "Vision based automatic inspection of insects in pheromone traps," in *Proceedings of the 2015 Fourth International Conference on Agro-Geoinformatics* (Istanbul: IEEE). 333–338. doi: 10.1109/Agro-Geoinformatics.2015.7248113

Zitnick, C. L., and Dollár, P. (2014). "Edge boxes: locating object proposals from edges," in *Proceedings of European Conference on Computer Vision* (Berlin: Springer). 391–405. doi: 10.1007/978-3-319-10602-1_26

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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 03 June 2022
doi: 10.3389/fpls.2022.812506

Optical Identification of Fruitfly Species Based on Their Wingbeats Using Convolutional Neural Networks

***Ioannis Kalfas*[1]*, Bart De Ketelaere*[1]*, Tim Beliën*[2] *and Wouter Saeys*[1]*

[1]Department of Biosystems, Faculty of Bioscience Engineering, MeBioS, KU Leuven, Leuven, Belgium, [2]Zoology Department, pcfruit vzw, Sint-Truiden, Belgium

The spotted wing Drosophila (SWD), *Drosophila suzukii*, is a significant invasive pest of berries and soft-skinned fruits that causes major economic losses in fruit production worldwide. Automatic identification and monitoring strategies would allow to detect the emergence of this pest in an early stage and minimize its impact. The small size of *Drosophila suzukii* and similar flying insects makes it difficult to identify them using camera systems. Therefore, an optical sensor recording wingbeats was investigated in this study. We trained convolutional neural network (CNN) classifiers to distinguish *D. suzukii* insects from one of their closest relatives, *Drosophila Melanogaster*, based on their wingbeat patterns recorded by the optical sensor. Apart from the original wingbeat time signals, we modeled their frequency (power spectral density) and time-frequency (spectrogram) representations. A strict validation procedure was followed to estimate the models' performance in field-conditions. First, we validated each model on wingbeat data that was collected under the same conditions using different insect populations to train and test them. Next, we evaluated their robustness on a second independent dataset which was acquired under more variable environmental conditions. The best performing model, named "InceptionFly," was trained on wingbeat time signals. It was able to discriminate between our two target insects with a balanced accuracy of 92.1% on the test set and 91.7% on the second independent dataset. This paves the way towards early, automated detection of *D. suzukii* infestation in fruit orchards.

Keywords: insect recognition, convolutional neural network, pest management, automatic monitoring system, wingbeat analysis, wingbeat frequencies, optical sensing and sensor, deep learning

Edited by:
Jucheng Yang,
Tianjin University of Science and Technology, China

Reviewed by:
Jun Liu,
Weifang University of Science and Technology, China
Alvaro Fuentes,
Jeonbuk National University, South Korea

****Correspondence:***
Wouter Saeys
wouter.saeys@kuleuven.be

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *10 November 2021*
Accepted: *18 May 2022*
Published: *03 June 2022*

Citation:
Kalfas I, De Ketelaere B, Beliën T and Saeys W (2022) Optical Identification of Fruitfly Species Based on Their Wingbeats Using Convolutional Neural Networks. Front. Plant Sci. 13:812506. doi: 10.3389/fpls.2022.812506

INTRODUCTION

Drosophila suzukii (Matsumura), the spotted wing *Drosophila* (SWD), is a major invasive fruit pest which is native to Western Asia, but has spread to many countries around the world. It was first spotted in Southern Europe in 2008 (Rasquera, Spain) and in the following years it spread to the majority of European countries across a wide range of environmental conditions and climates (Mortelmans et al., 2012; Asplen et al., 2015). Unlike the majority of other Drosophilidae, *D. suzukii* lays its eggs in healthy ripening fruits rather than damaged or overripe ones, thus creating special problems to growers. The host range of SWD includes mainly soft-skinned fruits and it is quite broad, having now been documented in cherries, peaches, nectarines, plums, persimmons, strawberries, grapes, blackberries, blueberries, raspberries, pluots, figs, and several other fruit crops, as well as a wide variety of non-crop host plants (Walsh et al., 2011; Kenis et al., 2016; Tait et al., 2021). Damage in fruit production by SWDs ranges from negligible to 80% crop loss (Dreves et al., 2009; Lee et al., 2011; Walsh et al., 2011; Asplen et al., 2015; Potamitis and Rigakis, 2015; Klick et al., 2016; Farnsworth et al., 2017; Yeh et al., 2020). A study looking into revenue losses due to SWD infestation and fruit rejections found that gross revenues decreased by 37% for raspberries and 20% for strawberries in California, United States (Goodhue et al., 2011). The spread of *D. suzukii* is quite fast since it is introduced or re-introduced to habitats worldwide *via* global fruit trade and it then moves quickly from one region to another by flying (Rota-Stabelli et al., 2013). Consequently, knowledge of SWD (or similar) population sizes at any given time would be very useful to growers of host crops and parties directly or indirectly affected by the subsequent economic losses since it would provide the ability to assess new possible infestations or the severity of existing ones.

Most traditional monitoring methods require a frequent human intervention to either sample larvae in fruits or identify and count trapped insects. These labor-intensive procedures are time consuming and can be inefficient when dealing with rapid pest invasions. In the case of SWD, their population can double in size in only 4 days (Emiljanowicz et al., 2014) and a single female can produce approximately 3,000 adult descendants within a couple of months (Tochen et al., 2014). Moreover, SWD flies are known to utilize a variety of non-crop hosts and alternative habitats (Dalton et al., 2011; Burrack et al., 2013; Atallah et al., 2014), which makes manual monitoring methods progressively more challenging and inefficient as the number of necessary inspection areas and field types increase. Besides, the high activity season of the SWD varies and lasts quite long, ranging from early July until late December according to studies conducted in the eastern part of the United States (Pelton et al., 2016; Guédot et al., 2018) as well as Europe (Clymans et al., 2019; Tait et al., 2021). Hence, a necessity for more automated monitoring systems of pest insect populations arises.

Automatic monitoring systems of pests can generate timely warnings in real-time and prompt farmers to act if needed. This could also help control the use of insecticides, which create severe negative effects on public health and the environment (Wilson and Tisdell, 2001; European Commission, 2019). By relying on data-derived metrics of pest population sizes, insecticide use could be applied only under certain infestation conditions and not as a precautionary measure. In the past years, several automatic insect traps have been developed (Jiang et al., 2008, 2013; Shieh et al., 2011; López et al., 2012; Lampson et al., 2013; Potamitis et al., 2015; Lima et al., 2020a). The two main approaches that prevail in designing insect monitoring devices are: (1) imaging of trapped insects; and (2) recording a sensor reading of the insect upon entry.

In the first approach, the insects are commonly trapped on a sticky surface which is imaged by a camera. Then, the trapped insects on that surface are counted and identified by using simple computer vision and artificial intelligence (AI) algorithms (Espinoza et al., 2016; Nieuwenhuizen et al., 2018; Lima et al., 2020a). Image-based traps are frequently combined with Convolutional Neural Network (CNN) classifiers and object detectors (Li et al., 2021). For example, Roosjen et al. (2020) used images taken from an unmanned aerial vehicle (UAV) and fed them to CNNs to detect SWD individuals trapped on sticky plates. They demonstrated a rather low area under the precision-recall curve (AUC) of 0.086 for female SWDs and 0.284 for male. When using static images instead, they detected female SWDs with a promising AUC of 0.506 and male SWDs with AUC of 0.603. Thus, despite the success of CNN models in classifying images or detecting objects, systems that employ CNNs still struggle to address challenges that arise in the field, such as varying illumination, blurry images due to insect movement, orientation or crowding, and uncalibrated systems (out of focus cameras, poor color calibration, white balancing, etc.). To overcome some of these challenges, practitioners often apply data augmentation by creating replicas of their original data with visual differences that simulate various real conditions. This way, CNN models learn features that distinguish their target insects from others in multiple different settings. Still, classifying small insects in images remains a challenge even for such complex models, especially for insects that do not have prominent or unique features.

In sensor-based insect traps, often an infrared or optical sensor is placed inside a lure trap to count the number of times a target insect enters, or to capture its wingbeat pattern or produced vibrations to classify it (van Roy et al., 2014; Potamitis and Rigakis, 2015; Potamitis et al., 2017; Lima et al., 2020b; Kalfas et al., 2021; Rigakis et al., 2021). Sensor-based traps are paired with lures, and they can either record events that likely belong to a target insect or capture more complex patterns on which prediction models are built. In two example cases, researchers built a detection system for Red Palm Weevil infestations in trees using bioacoustics signals produced by this insect (Ilyas et al., 2009; Hussein et al., 2010). Bioacoustic signals like calling or courtship sound signals are also recorded using microphones or similar audio recorders to classify insect species (Mankin, 1994; Chesmore, 2001; Raman et al., 2007; Zamanian and Pourghassem, 2017), but these devices are sensitive to wind noise or ambient sounds when deployed in the field. In two different studies, Potamitis et al. (2014, 2015) embedded an optoelectronic sensor in a McPhail-type trap and were able to count and classify fruitfly species by measuring the insects' wingbeat. Optoelectronic sensors provide several benefits for recording insect biometric data compared to microphones and cameras since they are not influenced by the environmental conditions or the target's distance from the sensor while recording data (Potamitis et al., 2018). Wingbeat data captured from optical sensors have already been used successfully to classify insect species and with the recent advances in the field of Machine Learning (ML) it has become possible to build high-performing classification systems (van Roy et al., 2014; Chen et al., 2014; Potamitis and Rigakis, 2015; Fanioudakis and Potamitis, 2018). However, strict validation procedures are crucial to avoid that over-optimistic results are obtained with these powerful machine learning techniques. In a previous study involving a rigorous validation strategy, we have shown that CNNs are able to classify wingbeat data of mosquitoes on the genus level, but were less successful at the species level (Kalfas et al., 2021).

Both the *D. melanogaster* (DM) and the *D. suzukii* (SWD) occur in similar habitats with presence of soft-skinned fruits and overlapping high activity seasons. However, unlike SWD, DM poses no considerable threat to fruit crops since it will mainly attack overripe fruit that are already unfit for sale. Hence, a system that can accurately discriminate between the two *Drosophila* genera will be very valuable to estimate the need for crop protection at any given time. Both insect types are very small in size and range between 2 and 4.5 mm in body length, and 2 and 3.5 mm in wing length (Walsh et al., 2011). On average, DMs are slightly smaller than SWDs, but there is substantial overlap between both populations. Using optical sensor recordings of the wingbeats, we aim to overcome the limitations that an in-field camera system would have, dealing with such small insects with similar appearance. As no reports were found on the discrimination of these highly similar inspect species from the Dropsophila genus based on their wingbeat signals, the aim of this study was to train and strictly validate CNN classifiers to discriminate wingbeat signals of the SWD pest from the DM as a stepstone towards automatic in-field pest monitoring.

MATERIALS AND METHODS

Insect Stock Culture

The *D. suzukii* culture used in the laboratory experiments originated from multiple collections of adults in a private garden (Gentbrugge, Belgium, 51°1.522′N, 3°46.093′E). The *D. melanogaster* culture was received from the "Expertise Unit on Educational Provision" (Faculty of Bioscience Engineering, KU Leuven, Belgium). The laboratory colonies were maintained in polystyrene Drosophila vials (Greiner Bio-One™ Insect Breeding Conical Container, 217,101) on a cornmeal-yeast-agar diet (42 g/l fresh yeast, *Saccharomyces cerevisiae*, Algist Bruggeman; 55 g/l white table sugar, Suikerraffinaderij Tienen; 90 g/l crushed cornmeal, Aveve; 2 g/l Ethyl 4-hydroxybenzoate 99%, Alfa Aesar; 9 g/l agar powder, VWR chemicals and 910 g/l tap water). The vials were stoppered using foam stoppers (Greiner Bio-One™ Ceaprenstop, diameter 36 mm, 330,070) and kept in a plant-growth chamber at 22 ± 1°C, 60 ± 11% RH, and a 16:8 l:D photoperiod.

Sensor Design

The wingbeat sensor consists of two main parts: (a) a sensing head and (b) a microelectronic device that handles how the signals are stored (**Figure 1**). The sensing head consists of two boards placed opposite to each other, which act as a light emitter and receiver, respectively. As an insect flies between the two boards, it occludes the emitted light with its body and wings. The light receiving board then records a pattern of varying light intensity values which constitutes the wingbeat signal in the time domain. The microelectronic device measures the Root Mean Square (RMS) value of the live signal and contains software that defines the sampling frequency, triggering and storing of wingbeat events (in an embedded SD card). For more details regarding the wingbeat sensor device we refer to Potamitis and Rigakis, 2015 and Kalfas et al., 2021.

Experimental Setup and Data Collection

All wingbeat data were recorded in a laboratory or a climate room by placing an optoelectronic sensor inside spacious insectary cages where either *D. melanogaster* or *D. suzukii* insects were free to fly in (**Figure 1**). The same sensor device was placed in each insect cage sequentially for a period of 2–3 weeks (**Figure 2**) until sufficient data were collected for each population, considering that the number of valid signals would be fewer than the total number of signals per population after our data cleaning process. We reared two separate populations per *Drosophila* species (four insect populations in total) and tried to limit the number of insects in each population to around 200–300 individuals. We did not select insects based on their age or sex and new insects kept on hatching from larvae in the food media during the entire experiment. The vials with the food media were replaced once the food was depleted and no new eggs seemed to appear inside.

To collect a dataset of wingbeat signals under controlled conditions (Controlled dataset in **Table 1**), all insect cages were placed in a "climate room" to have stable environmental conditions. The average temperature in this room was 22 ± 0.6°C and the average relative humidity was 64 ± 5%. During this controlled experiment 99,154 wingbeat signals were recorded across all populations. False triggers and weak signals (with a noisy Power Spectral Density) were filtered out by employing a data cleaning procedure which is explained in "Selected Data Types and Data Cleaning." The numbers of retained signals are summarized in **Table 1**.

A second set of wingbeat signals was compiled from data acquired in a different lab environment 6 months prior to the controlled dataset (**Table 1**). Data collection for this dataset lasted from late July until middle of October 2020, starting with the SWD class. The collection process of the DM class was initiated in August, but it was interrupted due to being provided with a non-flying variant of DMs. The process restarted late in September with a stock of wild DMs, but it was hindered by the environmental and room conditions at that time; hence the low numbers of DM wingbeat signals collected. Temperature and humidity were not controlled and varied according to the room environmental conditions, which were on average 23 ± 1°C and 55 ± 9% RH. After applying the higher mentioned filtering procedure, a total of 22,744 wingbeat signals were retained in this dataset; 21,572 of those belong to the SWD class and 1,172 belong to the DM class.

Selected Data Types and Data Cleaning

The time profiles of the wingbeats collected by the optoelectronic sensor device were digitized using a sampling frequency of

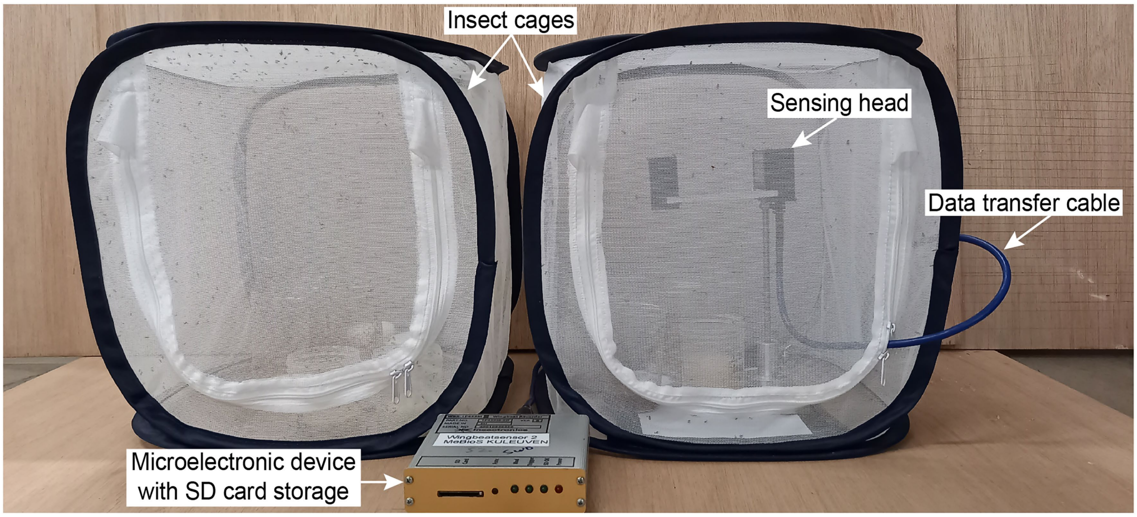


FIGURE 1 | Photograph of the laboratory setup with two insect cages and the wingbeat sensor. The wingbeat sensor consists of a sensing head, a data transfer cable and a microelectronic device with an SD card storage.

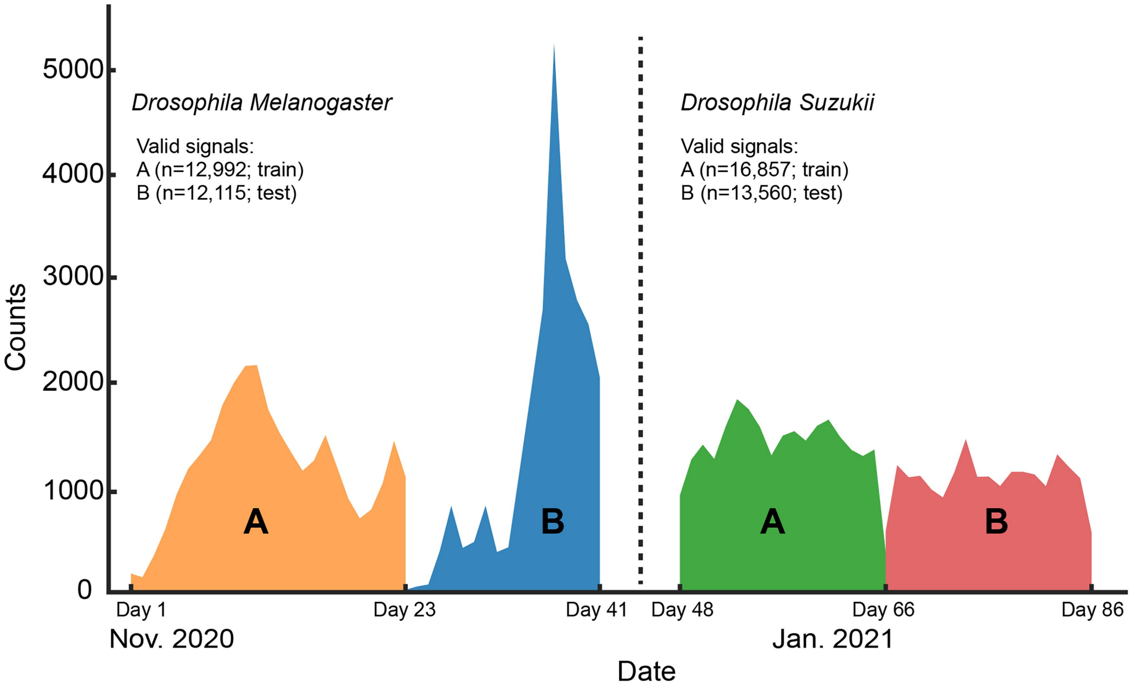


FIGURE 2 | Histogram of the signal counts collected on each day for the whole length of the controlled environment experiment. The number of valid signals per Drosophila species and the data split (train or test) they belong to are shown in the legend.

8 kHz. According to the Nyquist-Shannon sampling theorem (Shannon, 1949), this value should be sufficient to cover the main wingbeat frequencies of most insects, which were estimated to be < 1 kHz (Byrne et al., 1988), and their respective overtones in fine detail. The recorded signals consist of 5,000 light intensity measurements across 0.625 s. The intra-class variability for the two insects' wingbeat signals is high due to the various flight patterns that insects perform while flying through the sensor, while the inter-class difference seems small in both time (see **Figure 3**) and frequency domains (see **Figure 4**).

The three data types that were analyzed and classified in this research are: (1) wingbeat time signals, (2) their frequency content, and (3) time-frequency content (see **Figure 5**). The frequency content of the wingbeat time signals was calculated using the Welch power spectral density (PSD) method with a "Hanning" window (FFT size of 8,192 samples, segment length of 5,000 samples and 2,500 samples overlap). The spectrogram of the wingbeat time signals is calculated as the frequency-over-time representation (FFT size of 8,192 samples, hopping length of 5, window length of 600).

A strict data cleaning procedure was employed to remove weak signals or false triggers captured by the sensor. A preprocessing bandpass filter was first applied to all signals ("low-cutoff": 140 Hz, "high-cutoff": 1500 Hz). Then, two metrics were employed to evaluate the validity of wingbeat signals: (a) a "PSD-score" defined as the sum of a wingbeat signal's L2-normalized PSD values and (b) the number of peaks detected in its PSD, measured in V^2/Hz. The peaks were detected using Scipy-library's "find_peaks" function (McKinney, 2010) with the following settings:

- "prominence" = 0.001,
- "height" = 0.04,
- "width" = 1,
- "distance" = 5.

A wingbeat signal was considered valid if its PSD-score was between 3.5 V^2/Hz and 12 V^2/Hz, and it had more than 1 but fewer than 15 peaks in its PSD. These threshold choices for the two metrics were found to substantially reduce the number of weak or noisy signals without discarding too much data. In theory, a clean wingbeat signal PSD is expected to contain five peaks in total—one peak at the main wingbeat frequency (max<300 Hz; see **Figure 4**) and a single peak for each of the occurring harmonics. In practice, however, more peaks might occur in a high-resolution PSD (see **Figure 6**). Therefore, a ceiling of maximum 15 peaks is considered to be a safe threshold to keep signals with three times more peaks in their PSD than the theoretically "cleanest" signal and remove noisier signals. Lowering this threshold did not have a significant impact on the resulting signals, so further optimization is possible, but its increase is not recommended. Examples of a valid *D. melanogaster* wingbeat signal and one that was rejected by the above procedure are shown in **Figure 6**. The bandpass-filtered wingbeat signals were then fed to the classification models as waveforms of 5,000 dimensions or as PSD and spectrogram transformations. Both the PSD and spectrogram data were converted to decibel (dB) scale and only the values within the preprocessing filter's range (i.e., 140 to 1,500 Hz; 1,360 dimensions) were retained. The spectrogram images were downscaled to 295 × 400 pixel dimensions, maintaining the same aspect ratio of the original spectrograms, while allowing computational efficiency during training.

TABLE 1 | The number of signals for the two datasets used in this study (Controlled and Remote-Uncontrolled) and the data splits we applied.

	Controlled dataset		Remote-uncontrolled dataset
	Train and validation	Test	Test
DM signals	12,992	12,115	1,172
SWD signals	16,857	13,560	21,572

Data Splitting and Performance Evaluation

The aim of this research was to design an experiment where it would be possible to validate our trained models in a strict way and uncover their "true" performance in field conditions. To this end, we used data from two different datasets. The "Controlled" dataset, where data was collected under controlled environmental conditions and a "Remote-Uncontrolled" dataset where environmental variables were not controlled, and the data acquisition was 6 months earlier than for the controlled dataset (**Table 1**).

It should be noted that in our experimental setting a single insect can produce multiple similar signals within a population, because it can fly through the sensor multiple times while in the enclosure. When a random validation strategy would be applied, these highly similar datapoints could end up in different data splits and lead to over-optimistic estimates for the model performance (Kalfas et al., 2021).

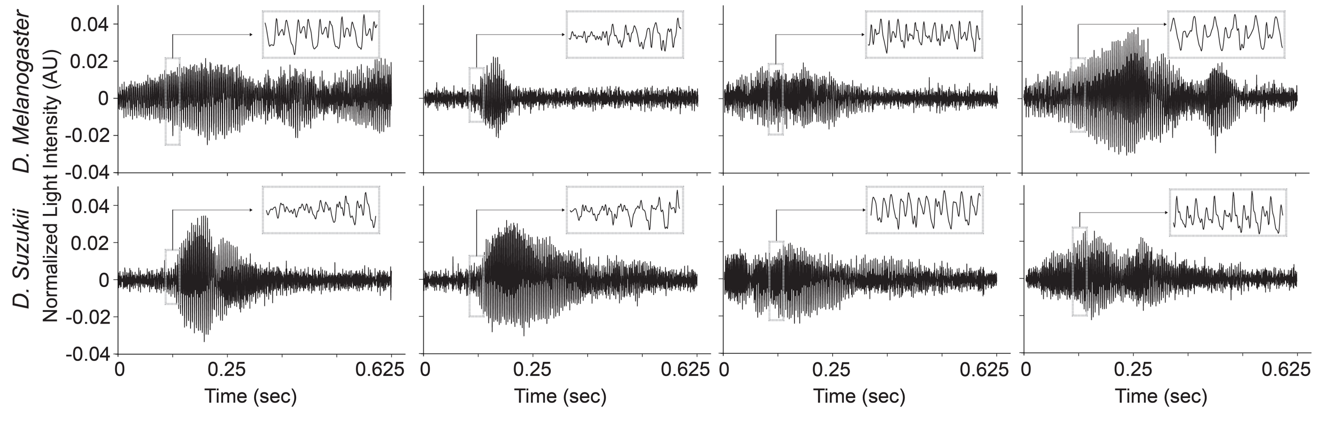


FIGURE 3 | Illustrations of different wingbeat time signals of *Drosophila suzukii* and of *Drosophila melanogaster*.

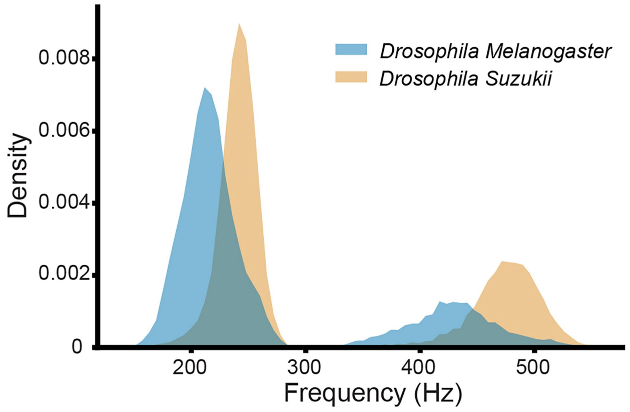


FIGURE 4 | Histograms of the main wingbeat frequencies and the first harmonics of *D. melanogaster* and *D. suzukii* wingbeat signals from the controlled dataset.

Using separate populations for training and testing, we aimed to tackle this problem and uncover the models' "true" classification performance which would emerge in field conditions. Hence, for the controlled dataset we created two separate insect populations for each of the two fruitfly species we are classifying (**Figure 2**). For each insect species, the population with the higher number of samples was chosen for training our models ("A" groups; **Figure 2**) and the other is used for testing ("B" groups; **Figure 2**). The training set was further split into training and validation sets which consist of 80 and 20% of its randomly sampled data, respectively. This validation set was used for hyper-parameter tuning of the models during training and model checkpoint selection. The remote uncontrolled dataset, which contains different insect populations, was used as an additional, truly external test set.

To evaluate the classification performance, we calculated the balanced accuracy and *F*1-score metrics on the test sets. The balanced classification accuracy in this binary setting is defined as the average of the proportion of correct predictions of each class individually, or the average of recall obtained on each class (best value equals to 1 and worst value is 0). The recall is defined as:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

where TP is the number of true positives and FN the number of false negatives. To calculate the *F*1-score, we first define precision as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

where TP is the number of true positives and FP the number of false positives. Finally, the *F*1-score is defined as:

$$F1 - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

which constitutes the harmonic mean between precision and recall. Time required to train or perform inference is measured and compared across models. For the latter, we take the average of five runs given a single batch of size 1. For the model with the highest classification performance, we report its confusion matrix for the test sets derived from the Controlled and Remote-Uncontrolled datasets.

Model Architectures and Training

Custom and state-of-the-art models from literature were chosen to fit 3 different types of wingbeat data, i.e., wingbeat time signals, their frequency (PSD) and time-frequency representations (spectrograms). For the wingbeat time and frequency signals, two models were trained: a custom 8-layer CNN—which we named "DrosophilaNet," and a variation of the state-of-the-art model for time-series and 1-dimensional data classification known as "InceptionTime"(Fawaz et al., 2019)—which we named "InceptionFly." DrosophilaNet consists of 8 blocks of Convolutional ("type": 1D-Convolution, "activation": ReLU), Batch-Normalization and Max-Pooling layers ("window": 2) that progressively create lower dimensional representations of the original data and feed their output to an Average Pooling, a Dropout layer ("drop rate": 0.2) and a Linear classification layer ("activation": Sigmoid) with 1 output unit. The number of filters in the convolutional layers increased in powers of 2, starting from 16 in the 1st block, to 2,048 in the 8th block, while the kernel size was fixed to a value of 3.

InceptionTime consists of residual blocks which in turn consist of multiple "inception modules" each. The residual blocks' input is transferred *via* skip connections to be added as input to the next block. Inception modules in each block reduce the input's dimensionality using a bottleneck layer and then extract hierarchical features of multiple resolutions by applying convolution filters of various lengths in parallel. These features are pooled, convolved, batch-normalized and fed to a ReLU activation function. For our InceptionFly, we used two residual blocks composed of three inception modules each. All inception modules had a fixed number of 32 convolutional filters using kernel sizes of: 6, 12, and 24. The two residual blocks were followed by an Average Pooling layer and a Linear classification layer ("activation": Sigmoid).

The spectrogram images were modeled with DenseNet121 (Huang et al., 2017), which is a popular CNN model for image classification tasks that was already tested and known to perform well in a similar task of classifying mosquito spectrogram images (Kalfas et al., 2021), while ranking first among other popular CNN models in a different study (Fanioudakis and Potamitis, 2018). We removed the top layer of DenseNet121 to replace it with a Linear fully-connected layer with 512 units ("activation": ReLU), a Dropout layer ("drop rate": 0.2) and a Linear classification layer ("activation": Sigmoid) with 1 output unit. Its input layer dimensions were modified

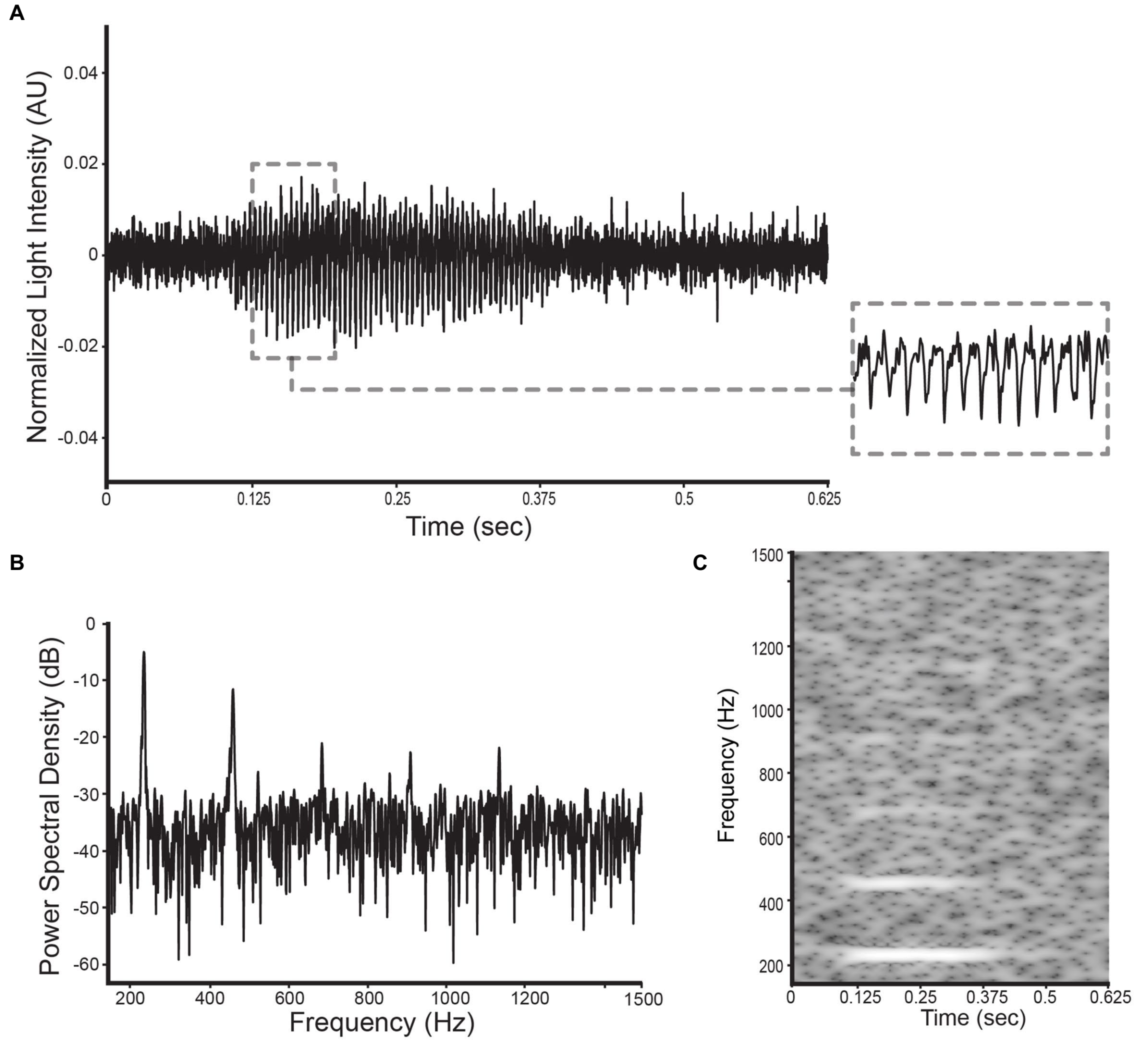


FIGURE 5 | Illustration of the selected datatypes used in this study for a *D. suzukii* signal: **(A)** the wingbeat signal, **(B)** its power spectral density, and **(C)** its spectrogram.

to match our spectrogram data dimensions (295 × 400) and the rest of the model's architecture remained intact. A summary of our data processing pipeline and an illustration of the model architectures used, are presented in **Figure 7**.

The training procedure for all neural network models was designed with the following settings:

- Training epochs: 100.
- Batch size: 32.
- Loss: categorical cross-entropy.
- Optimizer: Adam.

To help the neural networks to converge faster and reach high classification rates we used Cyclical Learning Rates (CLR; Smith, 2017) with the following settings for the CLR scheduler:

- Base learning rate: 0.0001.
- Max learning rate: 0.01.
- Cycle momentum: False.
- Mode: triangular.

The training procedure was allowed to run for 100 epochs while saving a model checkpoint (with the model's parameters) in each epoch. In the end, we selected the model checkpoint that showed the maximum validation accuracy. This accuracy is different from the balanced accuracy score we report on the model performance and is defined as the set of labels predicted by the model for each training datapoint, that exactly match the corresponding ground truth labels.

All models output a single probability score, ranging from 0 to 1, based on the Sigmoid activation of their last Linear classification layer. Probability scores below 0.5 are mapped to DM predictions, while scores greater or equal to 0.5 indicate a SWD prediction. Thus, in this binary classification setting the DM is considered the "negative class" and SWD the "positive class." We fine-tuned the selected models' decision thresholds

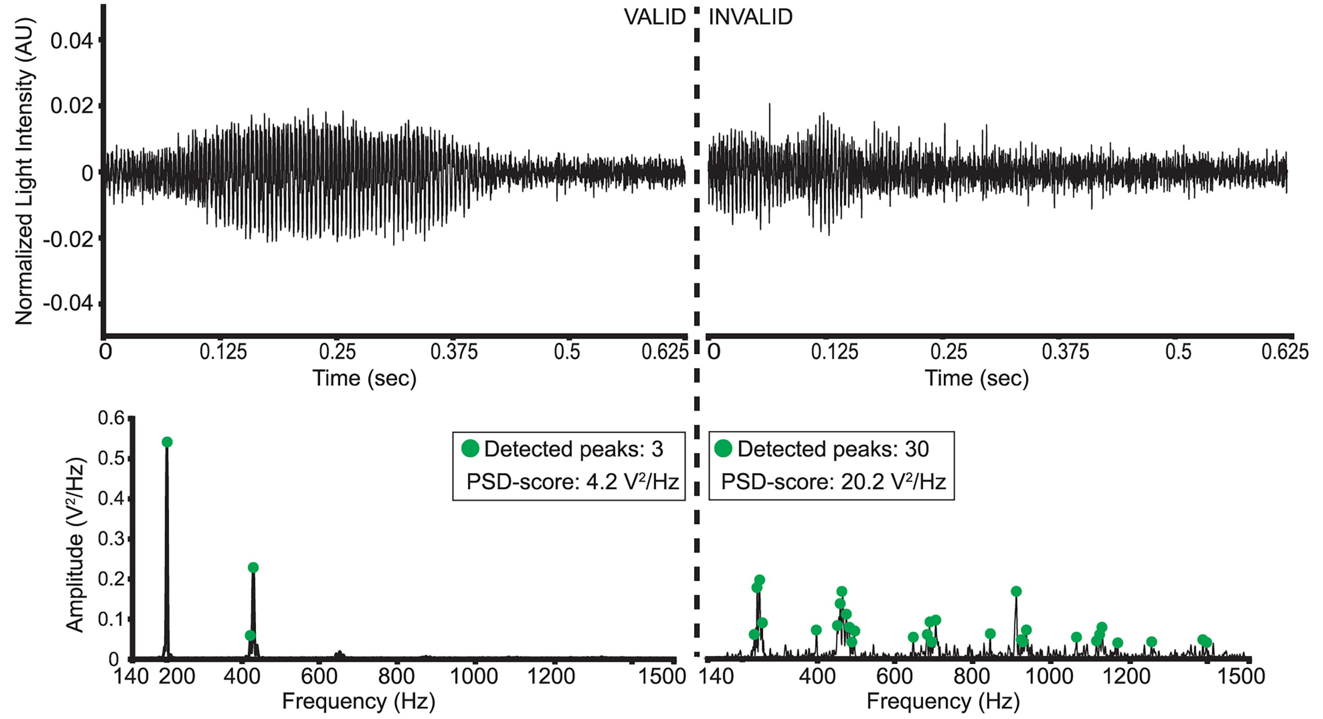


FIGURE 6 | Illustration of a "valid" and "invalid" *D. melanogaster* wingbeat signal and their respective PSD's.

by choosing the threshold that maximized the respective model's balanced accuracy score on the validation data (Fernández et al., 2018).

While training our models we experimented with custom data-augmentation techniques to increase model robustness and guide the neural networks in learning the important distinguishing features of the input data. Since all analyses begin with the wingbeat time signals—which are either modeled directly or transformed into frequency (PSD) or time-frequency (spectrogram) representations—we designed data transformations that might be applied on them as an "online" pre-processing step. First, a "Random-Roll" operation was applied that shifts the raw signal forwards or backwards in time by a number (of samples) randomly chosen from a range between 500 (0.0625 s) and 4,500 (0.5625 s). The part of the time signal that goes out of the original length because of shifting forwards (or backwards) is attached at the beginning (or the end) of the time signal. This augmentation technique helps in producing signals for various insect flights. Second, a "Random-Flip" operation was applied which mirrors the signal in the time dimension and third a "Random-Noise" operation was applied which adds Gaussian noise in a randomly selected part of the signal, which acts like signal "time masking" (Bouteillon, 2019). Each of the above operations had a 50% chance to be applied to any given input signal during training. As these 50% changes were applied independently, combinations of these operations were also possible.

All experimental scripts to train, evaluate and visualize our results were written in Python3, using the Pytorch library (version 1.8.1), Scikit-learn (version 0.24.1), and other scientific computing libraries (McKinney, 2010; Oliphant, 2010; Pedregosa et al., 2011; Mcfee et al., 2015). The code was executed on a single GPU (Nvidia RTX 5000; 16 GB RAM) laptop computer.

RESULTS AND DISCUSSION

Wingbeat Signals

As illustrated in **Figure 4**, the main wingbeat frequencies and the first harmonics of SWD and DM overlap. This makes it difficult to use these features for efficiently classifying between SWD and DM (Chen et al., 2014; Genoud et al., 2018). There is also no clear distinction between the two sexes of either insect species in terms of their wingbeat frequencies. This is not unexpected since visually, the sexes of both *Drosophila* species are very similar. Having a highly similar wing and body shape is expected to result in highly similar wingbeat recordings, which is confirmed by the wingbeat time signals for SWD and DM in **Figure 3**. Sex and age have been reported to influence the wingbeat recordings (Chen et al., 2014; Genoud et al., 2018). However, such information was not included in this study as for each *Drosophila* species both male and female flies of varying age were placed in the cages with the optical sensor, as would be the case in the field.

Our data cleaning procedure retained 55,524 valid wingbeat signals in the controlled dataset. Out of those, 29,849 were used for training and validation (SWD: $n = 16,857$; DM: $n = 12,992$), and the remaining 25,675 signals formed the test

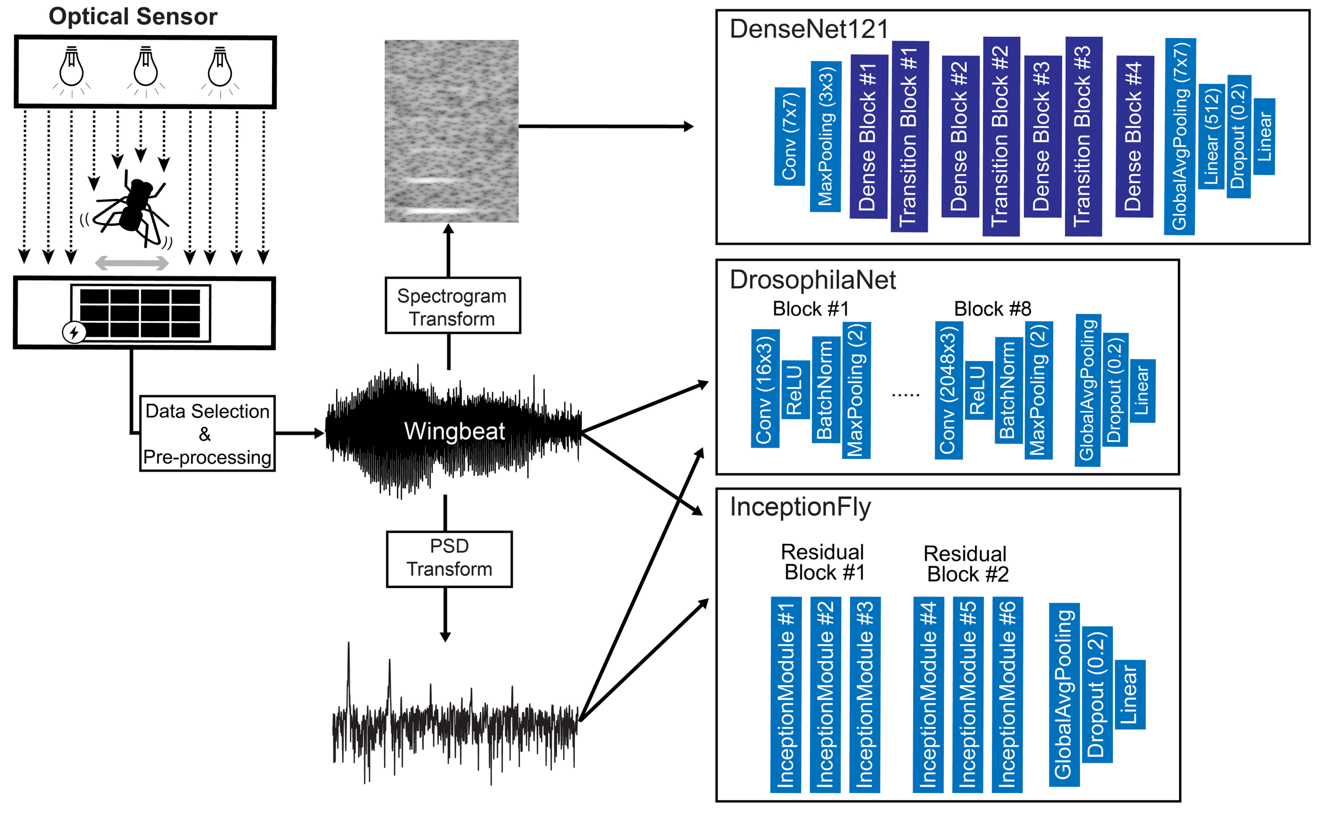


FIGURE 7 | Diagram of the data processing and modeling procedures including an illustration of the optical wingbeat sensor and the model architectures used in this study. For more information on the "Dense Block" and "Transition Block" layers, see Huang et al. (2017).

set (SWD: $n = 13{,}560$; DM: $n = 12{,}115$). For the remote uncontrolled dataset, the data cleaning procedure retained 22,744 valid wingbeat signals. Out of those, 21,572 belonged to SWD and 1,172 to the DM class. The low number of DM wingbeat signals in the Remote-Uncontrolled dataset can be attributed to unfavorable external conditions during this experiment. The experimental setup was in the same room as other machinery that raised the temperature and dried up the air during the morning hours of the same time period. This motivated us to use climate chambers for the collection of the Controlled dataset. Notably, the data acquired from the DM cages contained a considerably higher number of invalid signals compared to the SWD data. This may partly be attributed to the higher activity levels of DMs that lead to falsely triggering the sensor more often, e.g., by crawling on the sensor head. SWD insect population sizes seemed more stable throughout the length of the experiment, in contrast to DM populations which seemed to fluctuate.

Classifier Performance

The performance of all classifiers is summarized in **Table 2**. Their precision-recall curves for both datasets are shown in **Figure 8**. The best performing model was InceptionFly with wingbeat time signals. Trained with the Controlled dataset, it classifies wingbeat signals from the Controlled test set with a balanced accuracy score of 92.1% and $F1$-score of 0.93. DrosophilaNet performed similarly with a balanced accuracy of 91% and $F1$-score of 0.92. Using either InceptionFly or DrosophilaNet with PSD input data provided inferior classification results with balanced accuracies of 78.7 and 81.8%, and $F1$-scores of 0.67 and 0.84, respectively. Densenet121 trained with spectrograms provided a balanced accuracy of 87% and $F1$-score of 0.80 in the Controlled test set. This is in line with our previous work, where "InceptionTime" outperformed all other models on either wingbeat time signals, frequency signals or time-frequency signals (Fawaz et al., 2019; Kalfas et al., 2021). However, in this study, we found that DrosophilaNet had similar performance while being faster to train and perform inference with, compared to InceptionFly. In **Table 2** and **Figure 8**, we note that DrosophilaTime is more capable to model PSD data in both datasets, while it trains and performs inference on it faster, too. In **Figure 9**, the training and validation accuracy curves are plotted for the top two models in classification performance – InceptionFly and DrosophilaNet trained with wingbeat time signals. Despite InceptionFly reaching a higher validation accuracy, DrosophilaNet converges faster in the training set, while showing signs of high validation accuracies from the 10th epoch onwards. This makes it a good candidate for being deployed in the field where fast training and inference

TABLE 2 | Model performance for selected data types on the two test datasets (controlled and remote-uncontrolled).

Input	Model	Decision threshold	Total training time	Inference time	Controlled dataset		Remote-uncontrolled dataset	
					Balanced accuracy	*F*1-score	Balanced accuracy	*F*1-score
PSD (1360 × 1)	DrosophilaNet	0.694	**31 min**	**4 ms**	81.8%	0.84	83%	0.90
	InceptionFly	0.674	1.25 h	5.5 ms	78.7%	0.67	79.5%	0.86
Wingbeat signal (5000×1)	DrosophilaNet	0.744	53 min	4.8 ms	91%	0.92	91%	**0.97**
	InceptionFly	0.737	3.3 h	7.8 ms	**92.1%**	**0.93**	**91.6%**	0.96
Spectrogram (295×400)	DenseNet121	0.646	36.6 h	29.8 ms	87%	0.80	88.1%	0.95

Classification performance is measured using the balanced accuracy and F1-score. The models' fine-tuned decision thresholds are reported along with the total training time (measured in minutes or hours) and the inference time which was estimated for a batch size of 1, by taking the mean inference time of five runs for each model (measured in milliseconds). The best score for each performance metric is shown in bold.

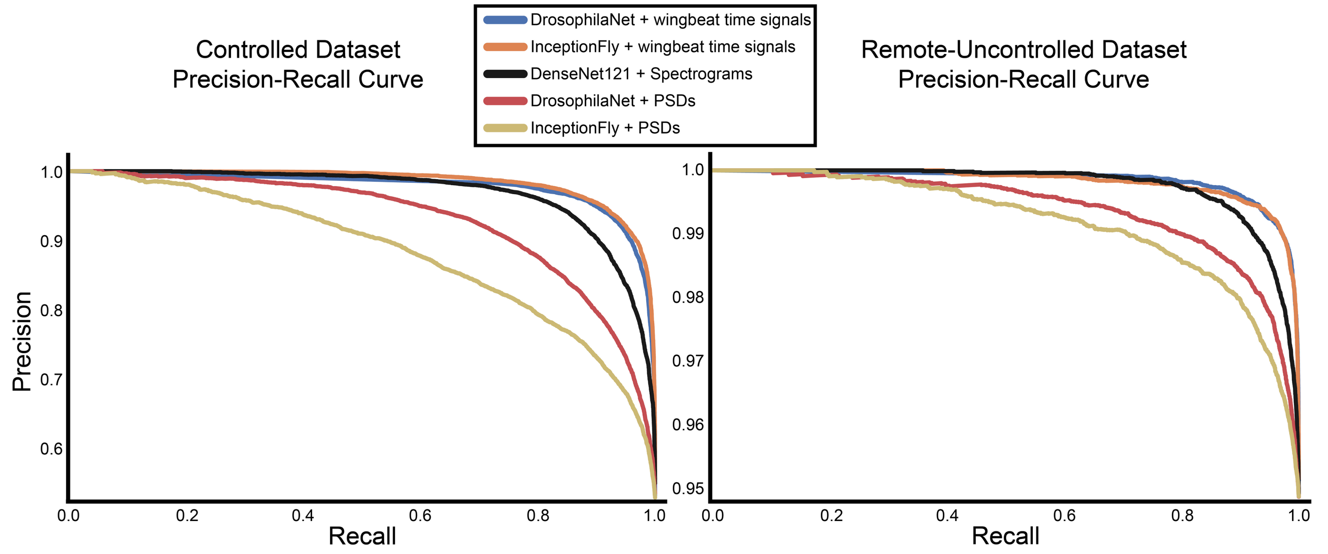


FIGURE 8 | Precision-recall curves for all models for the controlled and remote-uncontrolled datasets.

are critical. However, it could be interesting to investigate simpler variants of InceptionFly—fewer filters or smaller kernel sizes – that could improve its training and inference time performance.

The Remote-Uncontrolled dataset was used as an additional test set to evaluate our models' robustness. In **Figure 10**, the best model's confusion matrix and classification performance using wingbeat time signals on this dataset are illustrated. Data belonging to this dataset were collected months in advance, in different environmental conditions – which were expected to be closer to in-field conditions, and from different insect populations compared to those included in the training set. Still, InceptionFly trained on the Controlled dataset was able to classify wingbeat time signals in this Remote-Uncontrolled dataset with a balanced accuracy score of 91.6% and *F1*-score of 0.96. DrosophilaNet was again a close second with a balanced accuracy of 91% and a slightly higher *F*1-score of 0.97.

The two classification performance metrics used in this study—balanced accuracy and *F1*-score—are both reliable metrics for binary classification problems, but they are not equally sensitive to how the model performs on both classes. The *F*1-score is more sensitive to a model's performance in the positive class (SWD), while balanced accuracy equally considers both classes (SWD and DM) when evaluating model performance. This means that a higher *F*1-score is expected when a model accurately classifies many SWD signals regardless of making more mistakes in the DM predictions. On the other hand, the balanced accuracy metric assigns equal weight to SWD and DM mistakes. This explains the high *F*1-scores for the class-imbalanced Remote-Uncontrolled dataset. From a pest monitoring perspective, one could argue that it is more important to classify SWD correctly, but a robust model should also be sensitive to the DM classification performance for both the Controlled and Remote-Uncontrolled dataset. Therefore, we report both metrics.

Models trained on wingbeat time signals outperformed models using either PSD or spectrograms as input on both datasets. This suggests that important information for classifying

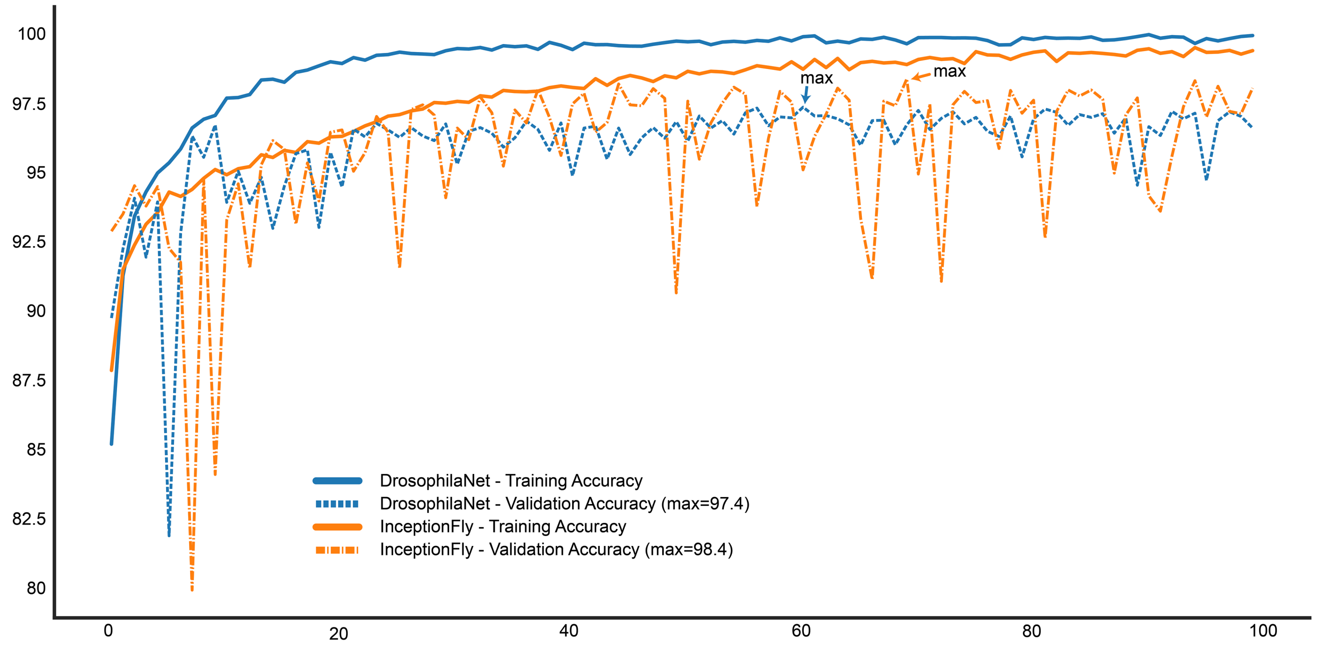


FIGURE 9 | Training and Validation accuracy curves for the top two performing models: InceptionFly and DrosophilaTime.

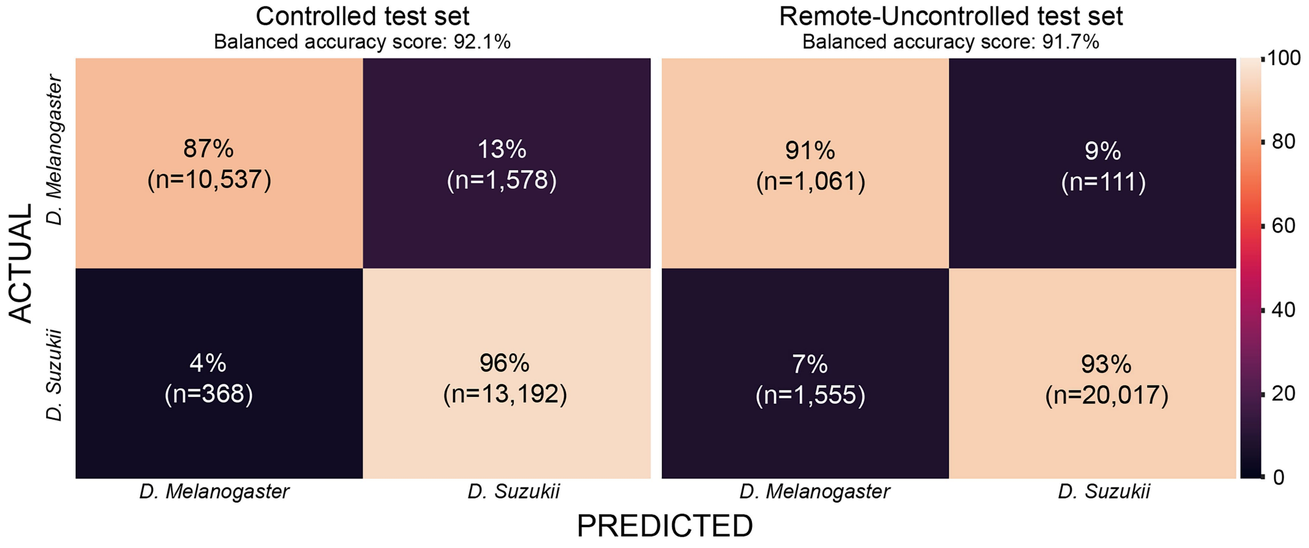


FIGURE 10 | Confusion matrix for InceptionFly trained with wingbeat time signals for our two datasets.

the wingbeats of these two highly similar insect species is present in the time dimension. It is hypothesized that micro-movements of the insects' wings are captured by the artificial neurons of InceptionFly or DrosophilaNet, which helps them classify wingbeats more accurately. This information is likely averaged out in the PSD and spectrograms. Higher resolution spectrograms could lead to better classification results, but that would create higher computational costs with even longer training and inference times. Besides, DenseNet121 was already the slowest among all models requiring 36.6 h to train and 29.8 ms to perform inference on a single datapoint, which is, respectively, 12 and 4 times longer than for the best performing model InceptionFly (see **Table 2**).

Towards Deployment in the Field

To obtain more insight in the cases were the algorithms resulted in misclassifications, we analyzed the temperature, relative humidity and timestamp of all misclassified wingbeat recordings. However, no clear correlations were found between these parameters and the models' classification performance. To obtain a better understanding of where the model fails and in what aspects the wingbeat patterns of the two species differ, it is recommended to investigate the role of the sampling frequency on classification performance of deep CNNs and to focus on the models' explainability.

The results reported here were obtained without applying any of the aforementioned data augmentation techniques since no significant performance change was noted when using these. Similar classification results were reached with all different data types used in this research when employing one or a combination of all considered data augmentation techniques. Data augmentation is expected to have a stronger effect when used with much smaller amounts of data since it would help to capture all different variations of the input data that would remain unseen given less data. An interesting follow-up study could help to identify the classification performance of wingbeat models and the effect of data augmentation starting from few data and increasingly adding more. The non-deterministic nature of neural networks would need to be taken into account when performing such experiments, since slight performance changes are expected after every training procedure.

The confusion matrix for InceptionFly trained with wingbeat time signals indicates a strong classification ability for this model (**Figure 10**). InceptionFly seemed to perform better for the SWD class compared to the DM class, since for the Controlled test set, only 4% of all SWD samples were misclassified as DM compared to 13% for the DM samples. For the Remote-Uncontrolled test set the misclassification rates were more balanced with 7 and 9%, respectively. The in-field performance of InceptionFly is expected to be close to its performance on the Remote-Uncontrolled dataset, but some challenges are expected still due to variation in the wingbeat frequencies in response to variable environmental conditions (Unwin and Corbet, 1984). Therefore, special attention needs to be given to performance monitoring and error analysis when the model is deployed in the field, especially for signals collected in extreme environmental conditions that were not covered in our two datasets.

CONCLUSION

Fruit production is increasingly challenged by the *D. suzukii* fruitfly which lays its eggs in healthy ripening fruits rather than damaged or overripe ones. Fruit growers demand automatic monitoring tools to efficiently protect fruit crops against this pest. To this end, we combined an optical wingbeat sensor with convolutional neural networks and evaluate the possibility to discriminate the wingbeat signals acquired for *Drosophila suzukii* and *D. melanogaster* fruitflies. To our knowledge, no other studies have previously built classification models for these two common pests. All models used in this work were validated in a strict way to uncover the "true" classification performance that can be expected in field conditions. A first validation involved classification of wingbeat signals collected in different enclosures under the same environmental conditions. Our best performing model, InceptionFly trained with wingbeat time signals was able to discriminate these wingbeat signals with an accuracy of 92.1%. Next, the model was also validated on wingbeat signals that had been collected independently under more variable environmental conditions. This validation was also successful with an accuracy of 91.7%. This shows that this model is sufficiently robust to be embedded in an automatic insect monitoring system that will operate in field conditions to provide accurate estimates of *D. suzukii* and *D. melanogaster* pest presence.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

IK: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft, writing—review and editing, and visualization. BK: conceptualization, methodology, resources, writing—review and editing, project administration, funding acquisition, and supervision. TB: resources, writing—review and editing, project administration, and funding acquisition. WS: conceptualization, methodology, resources, writing—review and editing, project administration, and supervision. All authors contributed to the article and approved the submitted version.

FUNDING

This research conducted with the financial support of VLAIO (Flanders Innovation & Entrepreneurship) (project HBC.2016.0795) and the Horizon 2020 Research and Innovation Programme, Grant Agreement no. 862563.

ACKNOWLEDGMENTS

We would like to thank Frank Mathijs of the MeBioS technical staff for his invaluable help to materialize many of the ideas for this project. Ludmila Peckova from the "Expertise Unit on Educational Provision" of KU Leuven for providing us stocks of wild *D. melanogasters*. We also thank Klaartje Bunkens and Praktijkpunt Landbouw Vlaams-Brabant for sharing their climate chambers to allow us to perform our experiments under controlled conditions. Finally, we would like to thank Ammar Ahmedi from pcfruit for sharing his valuable knowledge on our target insects.

REFERENCES

Asplen, M. K., Anfora, G., Biondi, A., Choi, D. S., Chu, D., Daane, K. M., et al. (2015). Invasion biology of spotted wing Drosophila (*Drosophila suzukii*): a global perspective and future priorities. *J. Pest Sci.* 88, 469–494. doi: 10.1007/s10340-015-0681-z

Atallah, J., Teixeira, L., Salazar, R., Zaragoza, G., and Kopp, A. (2014). The making of a pest: the evolution of a fruit-penetrating ovipositor in *Drosophila suzukii* and related species. *Proc. R. Soc. B Biol. Sci.* 281:20132840. doi: 10.1098/rspb.2013.2840

Bouteillon, E. (2019). Specmix: a simple data augmentation and warm-up pipeline to leverage clean and noisy set for efficient audio tagging. dcase. Community. Available at: https://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Bouteillon_27_t2.pdf

Burrack, H. J., Fernandez, G. E., Spivey, T., and Kraus, D. A. (2013). Variation in selection and utilization of host crops in the field and laboratory by *Drosophila suzukii* Matsumara (Diptera: Drosophilidae), an invasive frugivore. *Pest Manag. Sci.* 69, 1173–1180. doi: 10.1002/ps.3489

Byrne, D. N., Buchmann, S. L., and Spangler, H. G. (1988). Relationship between wing loading, Wingbeat frequency and body mass in homopterous insects. *J. Exp. Biol.* 135, 9–23. doi: 10.1242/jeb.135.1.9

Chen, Y., Why, A., Batista, G., Mafra-Neto, A., and Keogh, E. (2014). Flying insect classification with inexpensive sensors. *J. Insect Behav.* 27, 657–677. doi: 10.1007/s10905-014-9454-4

Chesmore, E. D. (2001). Application of time domain signal coding and artificial neural networks to passive acoustical identification of animals. *Appl. Acoust.* 62, 1359–1374. doi: 10.1016/S0003-682X(01)00009-3

Clymans, R., Van Kerckvoorde, V., Bangels, E., Akkermans, W., Alhmedi, A., De Clercq, P., et al. (2019). Olfactory preference of *Drosophila suzukii* shifts between fruit and fermentation cues over the season: effects of physiological status. *Insects* 10:200. doi: 10.3390/insects10070200

Dalton, D. T., Walton, V. M., Shearer, P. W., Walsh, D. B., Caprile, J., and Isaacs, R. (2011). Laboratory survival of *Drosophila suzukii* under simulated winter conditions of the Pacific northwest and seasonal field trapping in five primary regions of small and stone fruit production in the United States. *Pest Manag. Sci.* 67, 1368–1374. doi: 10.1002/ps.2280

Dreves, A., Walton, V., and Fisher, G. (2009). *A New Pest Attacking Healthy Ripening Fruit in Oregon*. Eugene: Oregon University.

Emiljanowicz, L. M., Ryan, G. D., Langille, A., and Newman, J. (2014). Development, reproductive output and population growth of the fruit fly pest *Drosophila suzukii* (Diptera: Drosophilidae) on artificial diet. *J. Econ. Entomol.* 107, 1392–1398. doi: 10.1603/EC13504

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

European Commission (2019). *The European Green Deal*. Brussels. European Commission.

Fanioudakis, L., and Potamitis, I. (2018). Deep networks tag the location of bird vocalisations on audio spectrograms. Available at: http://www.xeno-canto.org/. Accessed December 13, 2018.

Farnsworth, D., Hamby, K. A., Bolda, M., Goodhue, R. E., Williams, J. C., and Zalom, F. G. (2017). Economic analysis of revenue losses and control costs associated with the spotted wing drosophila, Drosophila suzukii (Matsumura), in the California raspberry industry. *Pest Manag. Sci.* 73, 1083–1090. doi: 10.1002/ps.4497

Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., et al. (2019). InceptionTime: finding AlexNet for time series classification. Available at: https://arxiv.org/abs/1909.04939. Accessed April 28, 2020.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from Imbalanced Data Sets. Vol. 10.* (Berlin: Springer), 978–983.

Genoud, A. P., Basistyy, R., Williams, G. M., and Thomas, B. P. (2018). Optical remote sensing for monitoring flying mosquitoes, gender identification and discussion on species identification. *Appl. Phys. B Lasers Opt.* 124, 1–11. doi: 10.1007/s00340-018-6917-x

Goodhue, R. E., Bolda, M., Farnsworth, D., Williams, J. C., and Zalom, F. G. (2011). Spotted wing drosophila infestation of California strawberries and raspberries: economic analysis of potential revenue losses and control costs. *Pest Manag. Sci.* 67, 1396–1402. doi: 10.1002/ps.2259

Guédot, C., Avanesyan, A., and Hietala-Henschell, K. (2018). Effect of temperature and humidity on the seasonal phenology of *Drosophila suzukii* (diptera: Drosophilidae) in Wisconsin. *Environ. Entomol.* 47, 1365–1375. doi: 10.1093/ee/nvy159

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks." in *Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*; 21–26 July 2017; IEEE, Honolulu, HI, United States

Hussein, W. B., Hussein, M. A., and Becker, T. (2010). Detection of the red palm weevil rhynchophorus ferrugineus using its bioacoustics features. *Bioacoustics* 19, 177–194. doi: 10.1080/09524622.2010.9753623

Ilyas, P., Ganchev, T., and Kontodimas, D. (2009). On automatic bioacoustic detection of pests: the cases of rhynchophorus ferrugineus and sitophilus ory zae. *J. Econ. Entomol.* 102, 1681–1690. doi: 10.1603/029.102.0436

Jiang, J. A., Lin, T. S., Yang, E. C., Tseng, C. L., Chen, C. P., Yen, C. W., et al. (2013). Application of a web-based remote agro-ecological monitoring system for observing spatial distribution and dynamics of *Bactrocera dorsalis* in fruit orchards. *Precis. Agric.* 14, 323–342. doi: 10.1007/s11119-012-9298-x

Jiang, J. A., Tseng, C. L., Lu, F. M., Yang, E. C., Wu, Z. S., Chen, C. P., et al. (2008). A GSM-based remote wireless automatic monitoring system for field information: a case study for ecological monitoring of the oriental fruit fly, *Bactrocera dorsalis* (Hendel). *Comput. Electron. Agric.* 62, 243–259. doi: 10.1016/j.compag.2008.01.005

Kalfas, I., De Ketelaere, B., and Saeys, W. (2021). Towards in-field insect monitoring based on wingbeat signals: the importance of practice oriented validation strategies. *Comput. Electron. Agric.* 180:105849. doi: 10.1016/j.compag.2020.105849

Kenis, M., Tonina, L., Eschen, R., van der Sluis, B., Sancassani, M., Mori, N., et al. (2016, 2004). Non-crop plants used as hosts by *Drosophila suzukii* in Europe. *J. Pest Sci.* 89, 735–748. doi: 10.1007/s10340-016-0755-6

Klick, J., Yang, W. Q., Walton, V. M., Dalton, D. T., Hagler, J. R., Dreves, A. J., et al. (2016). Distribution and activity of *Drosophila suzukii* in cultivated raspberry and surrounding vegetation. *J. Appl. Entomol.* 140, 37–46. doi: 10.1111/jen.12234

Lampson, B. D., Han, Y. J., Khalilian, A., Greene, J., Mankin, R. W., and Foreman, E. G. (2013). Automatic detection and identification of brown stink bug, *Euschistus servus*, and southern green stink bug, *Nezara viridula*, (Heteroptera: Pentatomidae) using intraspecific substrate-borne vibrational signals. *Comput. Electron. Agric.* 91, 154–159. doi: 10.1016/j.compag.2012.12.010

Lee, J. C., Bruck, D. J., Curry, H., Edwards, D., Haviland, D. R., Van Steenwyk, R. A., et al. (2011). The susceptibility of small fruits and cherries to the spotted-wing drosophila, Drosophila suzukii. *Pest Manag. Sci.* 67, 1358–1367. doi: 10.1002/ps.2225

Li, W., Zheng, T., Yang, Z., Li, M., Sun, C., and Yang, X. (2021). Classification and detection of insects from field images using deep learning for smart pest management: a systematic review. *Ecol. Inform.* 66:101460. doi: 10.1016/j.ecoinf.2021.101460

Lima, M. C. F., Leandro, M. E. D., Valero, C., Coronel, L. C. P., and Bazzo, C. O. G. (2020a). Automatic detection and monitoring of insect pests—a review. *Agriculture* 10:161. doi: 10.3390/agriculture10050161

Lima, M. C. F., Leandro, M. E. D., Valero, C., Coronel, L. C. P., and Bazzo, C. O. G. (2020b). Automatic detection and monitoring of insect pests—a review. *Agriculture* 10, 1–24. doi: 10.3390/agriculture10050161

López, O., Rach, M. M., Migallon, H., Malumbres, M. P., Bonastre, A., and Serrano, J. J. (2012). Monitoring pest insect traps by means of low-power image sensor technologies. *Sensors (Switzerland).* 12, 15801–15819. doi: 10.3390/s121115801

Mankin, R. W. (1994). Acoustical detection of *Aedes taeniorhynchus* swarms and emergence exoduses in remote salt marshes. *J. Am. Mosq. Control Assoc.* 10, 302–308.

Mcfee, B., Raffel, C., Liang, D., Ellis, D. P. W., Mcvicar, M., Battenberg, E., et al. (2015). "Librosa–audio processing Python library." in *Proceedings of the 14th Python in Science Conference*, 18–25. Available at: http://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcfee.pdf (Accessed May 5, 2022).

McKinney, W. (2010). Data structures for statistical computing in Python. *Proc. 9th Python Sci. Conf.* 1697900, 51–56. doi: 10.25080/Majora-92bf1922-00a

Mortelmans, J., Casteels, H., and Beliën, T. (2012). *Drosophila suzukii* (Diptera: Drosophilidae): a pest species new to Belgium. *Belgian J. Zool.* 142, 143–146.

Nieuwenhuizen, A., Hemming, J., and Suh, H. (2018). "Detection and classification of insects on stick-traps in a tomato crop using faster R-CNN." in *Proceedings of the Netherlands Conference on Computer Vision*; September 26–27, 2018.

Oliphant, T. E. (2010). Guide to NumPy. *Methods* 1:378. doi: 10.1016/j.jmoldx.2015.02.001

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., et al. (2011). Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830. doi: 10.1007/s13398-014-0173-7.2

Pelton, E., Gratton, C., Isaacs, R., Van Timmeren, S., Blanton, A., and Guédot, C. (2016). Earlier activity of Drosophila suzukii in high woodland landscapes but relative abundance is unaffected. *J. Pest Sci.* 89, 725–733. doi: 10.1007/s10340-016-0733-z

Potamitis, I., and Rigakis, I. (2015). Novel noise-robust Optoacoustic sensors to identify insects through Wingbeats. *IEEE Sensors J.* 15, 4621–4631. doi: 10.1109/JSEN.2015.2424924

Potamitis, I., Rigakis, I., and Fysarakis, K. (2014). The electronic McPhail trap. *Sensors (Switzerland).* 14, 22285–22299. doi: 10.3390/s141222285

Potamitis, I., Rigakis, I., and Fysarakis, K. (2015). Insect biometrics: Optoacoustic signal processing and its applications to remote monitoring of McPhail type traps. *PLoS One* 10:e0140474. doi: 10.1371/journal.pone.0140474

Potamitis, I., Rigakis, I., and Tatlas, N. A. (2017). Automated surveillance of fruit flies. *Sensors* 17:110. doi: 10.3390/s17010110

Potamitis, I., Rigakis, I., Vidakis, N., Petousis, M., and Weber, M. (2018). Affordable bimodal optical sensors to spread the use of automated insect monitoring. *J. Sensors.* 2018, 1–25. doi: 10.1155/2018/3949415

Raman, D. R., Gerhardt, R. R., and Wilkerson, J. B. (2007). Detecting insect flight sounds in the field: implications for acoustical counting of mosquitoes. *Trans. ASABE.* 50, 1481–1485.

Rigakis, I., Potamitis, I., Tatlas, N.-A., Potirakis, S. M., and Ntalampiras, S. (2021). TreeVibes: modern tools for global monitoring of trees for borers. *Smart Cities.* 4, 271–285. doi: 10.3390/smartcities4010017

Roosjen, P. P. J., 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, 2994–3002. doi: 10.1002/ps.5845

Rota-Stabelli, O., Blaxter, M., and Anfora, G. (2013). Drosophila suzukii. *Curr. Biol.* 23:R8, –R9. doi: 10.1016/j.cub.2012.11.021

Shannon, C. E. (1949). Communication in the presence of noise. *Proc. IRE* 37, 10–21. doi: 10.1109/JRPROC.1949.232969

Shieh, J. C., Wang, J. Y., Lin, T. S., Lin, C. H., Yang, E. C., Tsai, Y. J., et al. (2011). A GSM-based field Monitoring system for *Spodoptera litura* (Fabricius). *Eng. Agric. Environ. Food.* 4, 77–82. doi: 10.1016/S1881-8366(11)80016-9

Smith, L. N. (2017). "Cyclical learning rates for training neural networks." in *Proceedings—2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*; March 24, 2017; (IEEE), 464–472.

Tait, G., Mermer, S., Stockton, D., Lee, J., Avosani, S., Abrieux, A., et al. (2021). Drosophila suzukii (Diptera: Drosophilidae): a decade of research towards a sustainable integrated Pest management program. *J. Econ. Entomol.* 114, 1950–1974. doi: 10.1093/jee/toab158

Tochen, S., Dalton, D. T., Wiman, N., Hamm, C., Shearer, P. W., and Walton, V. M. (2014). Temperature-related development and population parameters for drosophila suzukii (Diptera: Drosophilidae) on cherry and blueberry. *Environ. Entomol.* 43, 501–510. doi: 10.1603/EN13200

Unwin, D. M., and Corbet, S. A. (1984). Wingbeat frequency, temperature and body size in bees and flies. *Physiol. Entomol.* 9, 115–121. doi: 10.1111/j.1365-3032.1984.tb00687.x

van Roy, J., De Baerdemaeker, J., Saeys, W., and De Ketelaere, B. (2014). Optical identification of bumblebee species: effect of morphology on wingbeat frequency. *Comput. Electron. Agric.* 109, 94–100. doi: 10.1016/j.compag.2014.09.014

Walsh, D. B., Bolda, M. P., Goodhue, R. E., Dreves, A. J., Lee, J., Bruck, D. J., et al. (2011). Drosophila suzukii (Diptera: Drosophilidae): invasive pest of ripening soft fruit expanding its geographic range and damage potential. *J. Integr. Pest Manag.* 2, G1–G7. doi: 10.1603/IPM10010

Wilson, C., and Tisdell, C. (2001). Why farmers continue to use pesticides despite environmental, health and sustainability costs. *Ecol. Econ.* 39, 449–462. doi: 10.1016/S0921-8009(01)00238-5

Yeh, D. A., Drummond, F. A., Gómez, M. I., and Fan, X. (2020). The economic impacts and Management of Spotted Wing Drosophila (Drosophila Suzukii): The case of wild blueberries in Maine. *J. Econ. Entomol.* 113, 1262–1269. doi: 10.1093/jee/toz360

Zamanian, H., and Pourghassem, H. (2017). "Insect identification based on bioacoustic signal using spectral and temporal features." in *2017 25th Iranian Conference on Electrical Engineering, ICEE 2017*; May 2–4, 2017; (1785–1790).

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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 04 July 2022
doi: 10.3389/fpls.2022.891785

Deep Metric Learning-Based Strawberry Disease Detection With Unknowns

*Jie You, Kan Jiang and Joonwhoan Lee**

Artificial Intelligence Lab, Department of Computer Science and Engineering, Jeonbuk National University, Jeonju, South Korea

There has been substantial research that has achieved significant advancements in plant disease detection based on deep object detection models. However, with unknown diseases, it is difficult to find a practical solution for plant disease detection. This study proposes a simple but effective strawberry disease detection scheme with unknown diseases that can provide applicable performance in the real field. In the proposed scheme, the known strawberry diseases are detected with deep metric learning (DML)-based classifiers along with the unknown diseases that have certain symptoms. The pipeline of our proposed scheme consists of two stages: the first is object detection with known disease classes, while the second is a DML-based post-filtering stage. The second stage has two different types of classifiers: one is softmax classifiers that are only for known diseases and the K-nearest neighbor (K-NN) classifier for both known and unknown diseases. In the training of the first stage and the DML-based softmax classifier, we only use the known samples of the strawberry disease. Then, we include the known (*a priori*) and the known unknown training samples to construct the K-NN classifier. The final decisions regarding known diseases are made from the combined results of the two classifiers, while unknowns are detected from the K-NN classifier. The experimental results show that the DML-based post-filter is effective at improving the performance of known disease detection in terms of mAP. Furthermore, the separate DML-based K-NN classifier provides high recall and precision for known and unknown diseases and achieve 97.8% accuracy, meaning it could be exploited as a Region of Interest (ROI) classifier. For the real field data, the proposed scheme achieves a high mAP of 93.7% to detect known classes of strawberry disease, and it also achieves reasonable results for unknowns. This implies that the proposed scheme can be applied to identify disease-like symptoms caused by real known and unknown diseases or disorders for any kind of plant.

Keywords: deep metric learning, unknown disease detection, strawberry disease detection, *K*-nearest neighbor, open set recognition

OPEN ACCESS

Edited by:
Lei Shu,
Nanjing Agricultural University, China

Reviewed by:
Sijia Yu,
Rutgers, The State University of New Jersey, United States
Zhiwei Ji,
Nanjing Agricultural University, China

****Correspondence:***
Joonwhoan Lee
chlee@jbnu.ac.kr

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *10 March 2022*
Accepted: *06 June 2022*
Published: *04 July 2022*

Citation:
You J, Jiang K and Lee J (2022) Deep Metric Learning-Based Strawberry Disease Detection With Unknowns. Front. Plant Sci. 13:891785. doi: 10.3389/fpls.2022.891785

INTRODUCTION

There has been much research into plant disease detection based on the deep object detection technique, and substantial advancements have been achieved in this field (Zhao et al., 2019). The object detection models for plant diseases have been developed in two directions: One is for better precision (Ren et al., 2015; Lin et al., 2017b; Tan et al., 2020) while the other is for faster response (Redmon and Farhadi, 2018; Zhang et al., 2018; Bochkovskiy et al., 2020). There are now many off-the-shelf object detection models that can be chosen for plant disease detection for a specific purpose (Xiao et al., 2021; Dananjayan et al., 2022).

In constructing a plant disease detector, researchers collect samples of known diseases and then successfully train a selected object detection model using these samples. However, there may be disease-like symptoms in the inference process that are not actually from the known diseases. One of the confidence levels for the predefined disease classes might be maximum but with a low value, which means that it can produce false detection, or just miss detection according to the detection threshold. To reduce the false detection rate, the detection threshold can be increased, but the real disease with obscure symptoms might be missed. This is an undesirable situation that leads to a large number of either false or missed detections depending on the detection threshold.

Open-set detection (Bastan et al., 2019; Fehérvári and Appalaraju, 2019; Mahdavi and Carvalho, 2021) could solve this problem, as it discerns the unknown diseases as they are in the inference process, although only known diseases are taken care of in the training process. Unfortunately, the technology is not yet mature enough to be practically utilized for fine-grained plant disease detection. The state-of-the-art performance is not that good, even for coarse-grained tasks of distinct objects that look different.

Another alternative method is the post-filtering approach that effectively reduces the erroneous detections involved in the detection process. Many post-filtering schemes can be chosen, but we selected DML-based classifiers (Li and Tian, 2018; Kaya and Bilge, 2019) to be used for known and known unknown diseases. DML produces the feature space in which each cluster of the class becomes compact by reducing the intra-cluster distances and increasing the inter-cluster distances.

Our proposed scheme is similar to the object detection of plant disease followed by simple post-filtering, but the prepared unknown samples are used to classify ambiguous samples into an unknown category. The post-filtering stage has two different types of classifiers: softmax classifiers for only known diseases and the *K*-NN classifier for known and unknown diseases. In training the first stage of the object detection model and the DML-based softmax classifier, we only used known samples of the strawberry disease. Then, the known unknown training samples are included to construct the *K*-NN classifier. The final decisions for known diseases are made based on the combined results of the two classifiers, while unknowns are detected solely from the *K*-NN classifier. **Table 1** summarizes the data type used to train the building blocks and their decisions in the inference process of our proposed scheme. Note that the DML-based post-filter can be used as a separate ROI classifier if the disease-like symptoms are manually annotated, as opposed to the automatic detection in the first stage. Therefore, the technology in our scheme can be exploited for both the detection and classification of plant diseases.

In the experiment, we adapt Faster R-CNN with Feature Pyramidal Network (FPN) for the object detection model and margin triplet loss for DML. To verify our scheme, we constructed a strawberry disease dataset and used it for the experiment. The contributions of this study can be summarized as follows:

(1) This study proposes a practical solution for detecting known and partly known unknown plant diseases that provide good detection performance. It achieves approximately 93.7% of mAP to known classes of strawberry disease, and it also achieves reasonable results for unknowns of real field data.

(2) The proposed scheme consists of two stages: the object detection stage and the DML-based post-filter stage. The object detection model can be freely chosen according to the design requirement because it can be separated from the following DML-based post-filter. In addition, the DML-based post-filter can be separated from the first stage, and it can also be exploited for the ROI-based classifier of known and unknown diseases. The separate DML-based *K*-NN classifier provides high recall and precision for both known and known unknown diseases.

RELATED WORKS

The proposed scheme consists of two consecutive stages of an object detection model, followed by add-on post-filtering. This section reviews the related works to our scheme, which include object detection for monitoring plant disease, DML to separate clusters of classes, and *K*-NN classifier for known unknown detection.

Object Detection Models for Plant Disease Monitoring

As mentioned previously, various object detection models are available for plant disease monitoring. They have been developed to achieve two objectives: better accuracy and higher speed. Faster R-CNN (Ren et al., 2015; Lin et al., 2017b) is a 2-stage model that is relatively slow but accurate. On the other hand, the YOLO family and SSD (Zhang et al., 2018) start from a single stage with detection performance that is fast but less accurate. However, there have been continuous developments aiming for better accuracy while sacrificing speed. For example, the recent version of the YOLO family (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018; Bochkovskiy et al., 2020) provides many design options according to different requirements. Moreover, a recent transformer model (Carion et al., 2020) for object detection has been announced, and it is ready to be further developed to compete with Convolutional Neural Network (CNN)-based models. In addition, diverse models have been developed to meet the needs of various applications, even if there are few application examples for plant disease detection (Lin et al., 2017a; Tan et al., 2020).

For plant disease detection, a model with better speed could be required, such as light YOLO v.5. A mobile robot can capture plant images in a greenhouse, and the board embedded in the robot can help automatically identify disease symptoms in the field. On the other hand, the captured images can be transmitted to a remote cloud site of a high-performance computing facility to be precisely scrutinized using an accurate but slow model. In this situation, Faster R-CNN or its variants, such as cascaded Faster R-CNN, would be a better choice. Note that the classification approach for monitoring diseases is hard to automatize (Kim et al., 2021); this is because the image-containing symptoms of the disease should be manually located to take pictures and then fed into the classification-based monitoring system. However, it is still an important way to identify known and unknown diseases or disorders. Kim et al. (2021) and Liu and Wang (2021) provide excellent reviews of deep learning-based disease detection and classification models.

Post-filtering and Deep Metric Learning

The post-filtering approach is a practical way to improve detection accuracy, and it can be added to plant disease detection. Because the additional post-filter can reduce false detections, the confidence threshold of the detection stage can typically be lowered to increase the recall, even if that increases the number of false detections. Fuentes et al. (2020) adapted the idea to their one-*versus*-all post-filtering approach in tomato disease detection, while Kim et al. (2021) shared a similar idea in their cascaded Faster R-CNN for strawberry disease detection.

In this study, we propose the use of DML to build a low-dimensional feature space of known disease classes, where the clusters are well separated, by increasing the inter-cluster distances while reducing the intra-cluster distance (Kaya and Bilge, 2019). Furthermore, Ji et al. (2021) proposed a framework in which the features are learned by a deep learning feature extractor and WDM-tSNE is applied to accurately cluster the feature space of plant disease. In general, metric learning is done to obtain a proper metric for classifying objects, which captures a mapping function from visual objects to a low-dimensional embedded feature space with respect to a predefined distance metric, such as Euclidian or L1 distance. There are two different metric learning structures with different losses: one is the Siamese structure that uses contrastive loss (Chopra et al., 2005) and the other is the triplet structure with triplet loss (Schroff et al., 2015). Janarthan et al. (2020) have adapted the former structure to citrus disease classification. In our scheme, we choose the latter triplet structure. The essence of the DML in our scheme is to obtain a mapping that will separate clusters of known classes well in the feature space to make sufficient room for the known unknown diseases. Better classification performance for known diseases can be obtained by applying the softmax classifier to the embedded features from the metric learning. However, for the unknowns, we used the *K*-NN classifier based on the DML-embedded features that could be lost or falsely detected when only the object detection is applied. Although the object classifier after the object detection produces better performance, it is difficult to include the known unknowns, because there could be a huge set of unknown unknowns that are only experienced in the inference process. In other words, previous methods could not well expect the unknown unknowns in the training process.

Open World Setting for Unknown Disease Recognition

Significant progress has been made with machine intelligence, which is another technique for continual and life-long learning for open-world recognition, even if it is premature for practical applications, especially fine-grained tasks (Schlachter et al., 2019a,b, 2020; Geng et al., 2020). In the most general problem settings of the open world, no type of unknown can be contained in the training dataset, that is, it only appears in the test environment. Joseph et al. (2021) identify the open-world detection problem in 3-dimensional space, where one axis is the direction of increasing problem difficulty, one axis is the direction of open-set learning, and the last axis is incremental learning. In terms of the first axis of problem difficulty, open-set identification is more difficult than classification alone. However, if there is no prior assumption of unknowns, as is the case in the traditional open-set recognition problem setting, then the resulting state-of-the-art classification performance is not that good. For example, the state-of-the-art performance for easy MNIST, SVHN, and CIFAR-10 dataset exceeds 90%, but for difficult CUB and ImageNet dataset does not reach 90% in terms of AUROC (Vaze et al., 2021). In open object detection, which is a much harder problem than classification, the technology is far from being practically applicable for difficult plant disease

TABLE 1 | Proposed data type scheme for known and unknown disease detection.

	Type of disease data	First object detection stage	Second stage DML-based post-filter		
			Softmax	*K*-NN	Combined**
Training	Known	Used	Used	Used	Not used
	Unknown	Not used	Not used	Used	Not used
Inference	Known	Detected	Classified	Classified	Classified
	Unknown	Possibly detected*	Not classified	Classified	Not classified

*Disease-like symptom can be detected in the first object detection model, but it is determined by the K-NN classifier.**This stands for the final decision of the combined softmax and K-NN classifiers for known disease.*

detection. Because incremental learning (Parisi et al., 2019) for continual and life-long learning (Parisi et al., 2019) is beyond the scope of our work, it is not reviewed in this article, although it is related to open-set recognition.

In this article, we release the constraints on the rigorous open-set problem setting. For example, we do not know the name of the disease for samples, but they certainly exhibit similar disease-like symptoms that may have originated from diseases or disorders. Compared to the samples of major diseases, such samples look diverse and the frequency of similar objects is rare. One point that we want to emphasize is that the classifier performance of the closed set data is positively correlated with that of the open-set data (Parisi et al., 2019). In our scheme, DML tries to make a better classifier for the closed disease dataset while simultaneously leaving large empty room to locate unknowns.

METHODS

Figure 1 shows a schema of the proposed scheme. Our scheme is divided into two stages: the object detection module and the deep metric learning module. In the training, the object detection module can be trained with known disease samples to find as many potential known disease positions with the object classifier as possible. Then, the feature embedding of the post-filter is trained by DML to separate the clusters of known classes well. In the deep metric learning module, we cannot consider the unknown disease-like samples, so the training of the post-filter is identical to that of the conventional method of object detection and its refinement. Note that we enlarged the bounding boxes of the object detection results and sent for post-filter training; this is done to allow for dislocation of the object detection results and to include more context information around disease. Then, the embedded features of bounding boxes of known diseases are extracted from the DML-learned network to build the softmax classifier. Once the DML-learned network and softmax classifier training is finished, the weight is frozen and DML-embedded features from known and known unknown samples are used to build the *K*-NN classifier.

In the inference process, known and unknown disease samples are fed into the trained object detector. Then, the extended bounding box around the symptom is given to extract DML-trained features to be categorized by the softmax and *K*-NN classifiers. In this study, the softmax classifier is only concerned with known diseases, while the *K*-NN classifier deals with both known diseases and unknowns. The overall classification category of known diseases can be made by the combined decision of softmax and *K*-NN classifiers.

Object Detection Model

As described in the previous section, there have been diverse object detection technologies for plant disease monitoring. In our scheme, we choose FPN-Based Faster R-CNN for accurate detection. According to the open-set object detection, it provides the best accuracy based on standard protocol (Dhamija et al., 2020). Note that our scheme cannot detect unknown unknowns, because these are inevitably ignored in the training of the building blocks of our scheme. The object classifier in the object detection module distinguishes the known diseases from the background and produces the classification probability for knowns. **Figure 2** shows the conventional FPN-Based Faster R-CNN, which can detect various sizes of objects due to the exploitation of the pyramidal feature structure (Lin et al., 2017b). In this study, we want to emphasize that a low detection threshold would be better so as not to ignore the disease-like symptoms that are from unknown diseases or disorders. The size of the input image was 224 × 224 pixels to fit the CNN backbone. The number of diseases in the object detection stage was eight, including an angular leafspot, anthracnose (fruit rot, runner), blossom blight, gray mold (fruit), leaf spot, and powdery mildew (fruit, leaf). Note that some diseases show symptoms at different parts, and these are treated as different categories, because the part images are quite different.

Deep Metric Learning for Embedded Features

Our scheme chooses the ResNet50 network with margin triplet and cross-entropy losses for DML. The embedded features are used to refine the softmax classifier. In general, there are many false detections of normal leaf, fruit, flower, and runner as one of the diseases in the first stage of object detection. In our post-filter, each one is also treated as a separate class for training DML. The false detection of normal parts as diseases can be corrected in the DML-based classifiers. Therefore, we have considered 12 known classes in the DML-learning (eight known diseases and four normal parts).

There are two losses involved in the DML of margin triplet loss for embedded features and cross-entropy loss for the softmax classifier. The margin triplet loss is defined as Schroff et al. (2015):

$$L_{tuplet} = max\,\{d(f(x_a),\ f(x_p)) - d(f(x_a),\ f(x_n)) + margin\},\ 0) \tag{1}$$

where,

$$d(x_i,\ x_j) = \frac{x_i \cdot x_j}{max(||x_i||_2 \cdot \left|\left| x_j \right|\right|_2,\ \ \varepsilon)} \tag{2}$$

In Eq. (1), $f(x_a)$, $f(x_p)$, and $f(x_n)$, respectively, represent the features of anchor, positive, and negative image samples after mapping $f()$, from the network in **Figure 3**. Here, $d()$ is the Euclidian distance. The value of the margin was set to 0.01, and εwas $1e^{-8}$, which is a very small value to avoid dividing by zero. The cross-entropy loss is

$$L_{ce} = \frac{1}{N}\sum_{n=1}^{N} log\left(\frac{exp\left(f\left(x_n\right)\right)}{\sum_{c=1}^{C} exp(f(x_c))}\right) \tag{3}$$

where N spans the size of the batch and C is the number of classes.

Figure 3 presents the training of the DML with the softmax classifier in our scheme. The size of the input image is 256 × 256 to meet the requirements of the first CNN layer of the shared network to obtain a proper mapping in **Figure 3**. Note that

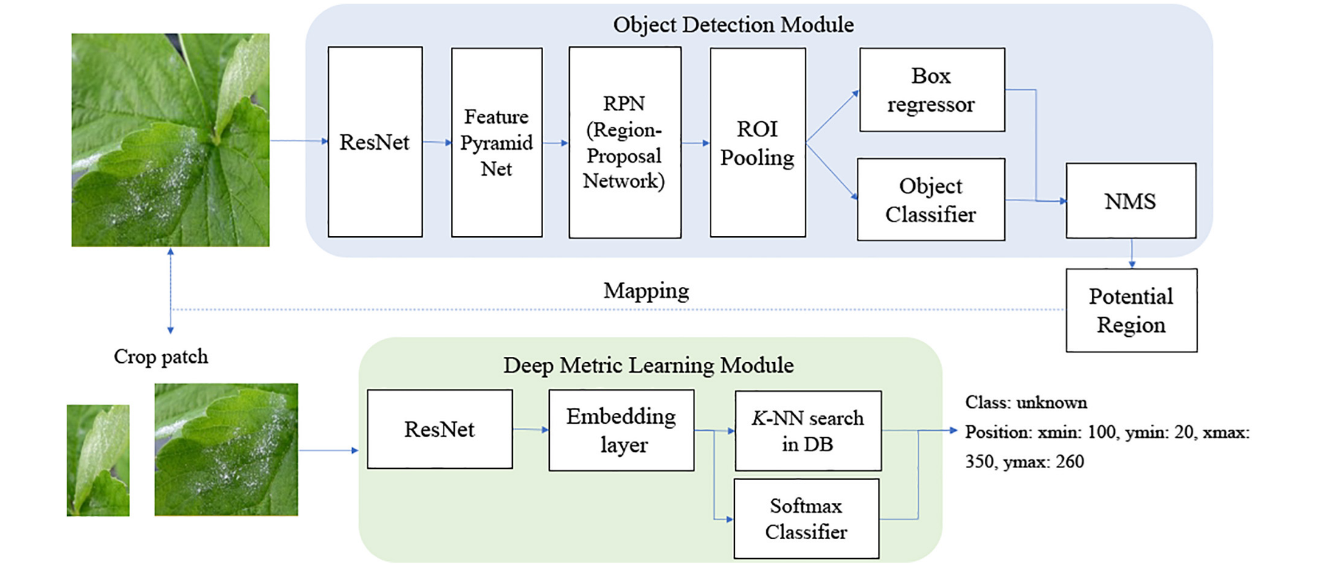


FIGURE 1 | Structure of overall scheme for inference.

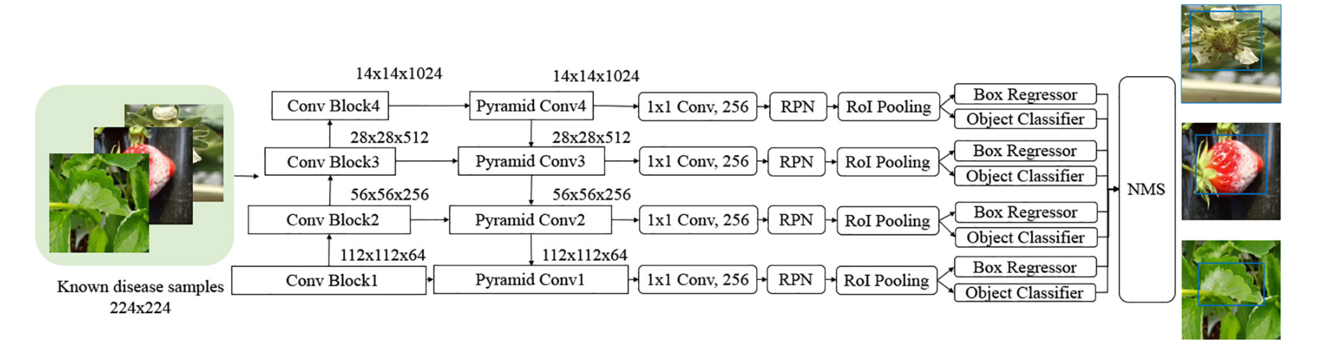


FIGURE 2 | Feature pyramidal network (FPN)-based Faster R-Convolutional Neural Network (CNN) for potential disease detection.

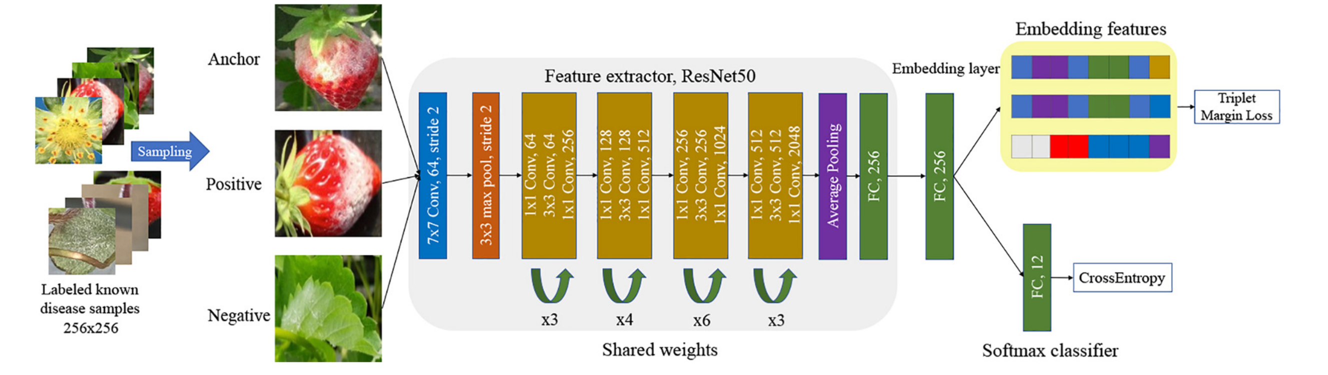


FIGURE 3 | Triplet network and loss with softmax classifier.

the extended bounding boxes from the object detection step are normalized to a uniform size. During the training, the feature extractor tries to minimize the margin triplet loss, which minimizes the Euclidian distance between a pair of the anchor and positive image, and maximizes the Euclidean distance between a pair of anchor and negative image, after

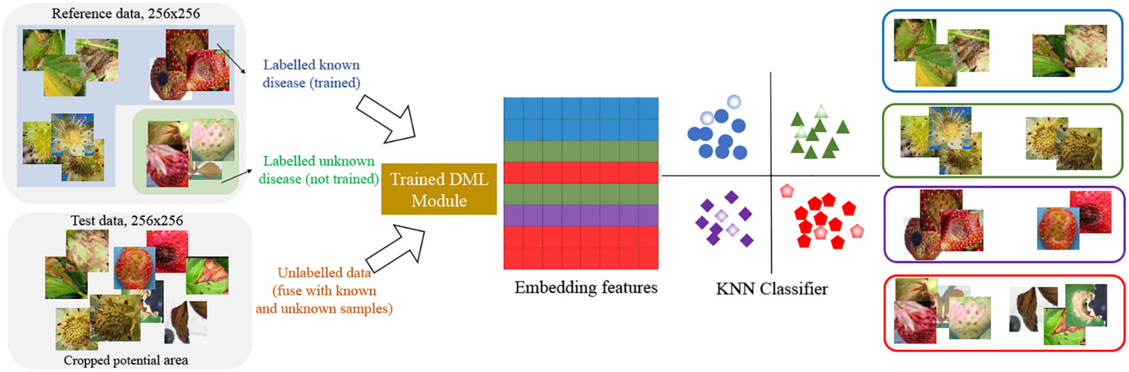

Reference data, 256x256
Labelled known disease (trained)
Labelled unknown disease (not trained)
Trained DML Module
Test data, 256x256
Unlabelled data (fuse with known and unknown samples)
Cropped potential area
Embedding features
KNN Classifier

FIGURE 4 | *K*-nearest neighbor (*K*-NN) classifier to categorize the disease classes with unknowns.

trainable mapping in ResNet50. In actuality, the same triplet networks sharing the weight parameters are simultaneously learned. Finally, the dimension of the embedded features that are used for the softmax classifier, and later the *K*-NN classifier is 256. We followed the method in Schroff et al. (2015) to sample semi-hard triplets to train the network. The semi-hard samples are the subset of all triplet samples, in which the distance between negative and anchor is further from the positive and anchor, $||f(x_i^a) - f(x_i^p)||_2^2 < ||f(x_i^a) - f(x_i^n)||_2^2$. This is a crucial step to speed up training and ensure the network convergence.

K-Nearest Neighbor (*K*-NN) Classifier for Categorizing the Diseases With Known Unknown Samples

In the second stage of our scheme, the *K*-NN classifier (Schroff et al., 2015) is built as a lazy learner. Here, the reference data includes known and known unknown samples with normal parts for the *K*-NN classifier. As a result, the number of classes in the *K*-NN classifier is 13, consisting of eight known diseases, four normal parts, and the class for known unknowns. In the experiment, we set $K = 13$ and chose a class randomly when the tie happens on multiple majority classes. **Figure 4** shows how the images are mapped into 256-dimensional embedded features and how to decide one of the class labels including unknowns in the *K*-NN classifier.

Note that there are duplicate classifiers in our scheme; one is from the softmax classifier and the other is the *K*-NN classifier. They both exploit DML-embedded 256-dimensional feature, but the softmax classifier does not take care of unknowns. As a result, there are 12 categories for the softmax classifier and one more unknown category for the *K*-NN classifier. There is no specific reason to make a different number of categories except for the fact that the softmax classifier is solely focused on known diseases to measure its performance in terms of average precision (AP) and mean AP (mAP), while the *K*-NN classifier considers both the known and unknown diseases.

The final classification of the known diseases and normal parts can be obtained by combining the two different decisions: one from the softmax classifier and the other from the *K*-NN classifier. There are typically no probabilities from the *K*-NN classifier, but we define the probability of the *j*-th class as:

$$p_j^{K-NN} = \frac{the\ number\ of\ nearest\ neighbors\ in\ class\ j}{K} \tag{4}$$

$$for\ j \in \{1, 2, \ldots, C\} \tag{5}$$

In the experiment, $C = 12$ without the unknown class. The probability can be combined with that from the softmax output to make the final decision. We simply multiply the two probabilities and take the class that has the maximum value, as in Eq. (5):

$$class\ label = \arg\max \left\{ p_j^{K-NN} \times p_j^{softmax} \right\} \tag{6}$$

where $p_j^{softmax}$ denotes the output probability of the softmax classifier. Therefore, the final decision rules for known diseases and unknowns can be summarized as follows:

TABLE 2 | Number of bounding boxes for the training and testing of disease objects.

Name	First stage		Second stage	
	Bounding boxes		Extended bounding boxes	
	Training	Test	Training (Aug)	Test
Angular leafspot	818	265	6,162	265
Anthracnose (fruit rot)	188	57	1,424	57
Anthracnose (runner)	237	166	30,897	166
Blossom blight	1,906	265	18,182	265
Gray mold (fruit)	1,468	224	13,069	224
Leaf spot	2,353	497	14,627	497
Powdery mildew (fruit)	405	161	2,626	161
Powdery mildew (leaf)	1,764	371	14,313	371
Normal (flower)	–		967	92
Normal (fruit)	–		1,842	104
Normal (leaf)	–		10,984	1,066
Normal (runner)	–		31,191	452
Unknowns	–		3,830*	862
Total	9,139	2,006	150,114	4,582

**Second stage unknown training data prepared for lazy classifier K-NN to find the unknown, which is unseen while training the feature extractor (ResNet).*

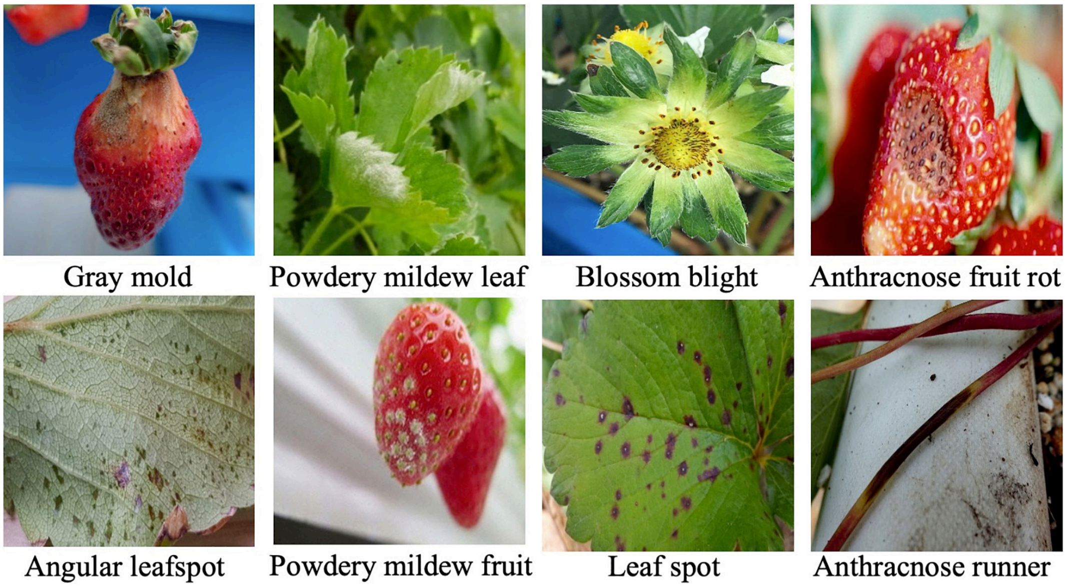


FIGURE 5 | Sample images for training disease detection.

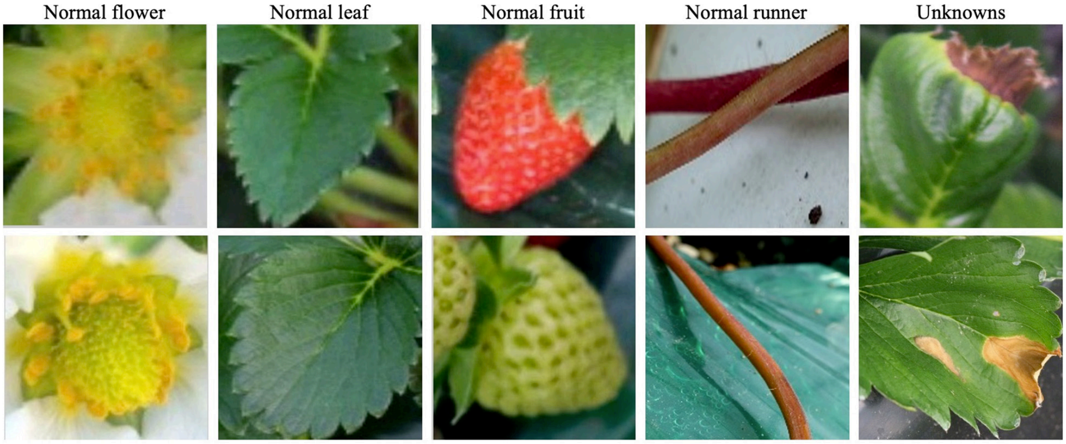


FIGURE 6 | Sample images of normal leaf, fruit, flower, and runner, with unknowns.

Rules

1) If the K-NN classifier decides the image sample is unknown, it is an unknown disease.

2) Otherwise, refer to Eq. (5) to decide the proper class and probability among known. classes.

EXPERIMENTAL RESULTS

Dataset for Experiment

For the experiments, an image dataset of strawberry diseases is constructed from the images taken by cellular phones in many greenhouses. The total number of images in the dataset is 7,230, and angular leafspot, anthracnose (fruit rot, runner), blossom blight, gray mold (fruit), leaf spot, and powdery mildew (fruit, leaf) disease images are included with normal images of flower, fruit, leaf, and runner. The disease images were taken by a cellular phone without any additional treatment to provide a more realistic appearance.

Training Feature Pyramidal Network (FPN)-Based Faster R-Convolutional Neural Network (CNN) Object Detector for Disease Monitoring

For the training, the diseases and their bounding boxes enclosing the symptoms were annotated. The number of bounding boxes for each disease used for training and testing are,

TABLE 3 | Final results of known disease detection for the test data.

Name	AP		
	Faster R-CNN	**+ Softmax classifier**	**+ Comb. w. *K*-NN classifier**
Angular leafspot	0.853	0.923	0.922
Anthracnose (fruit rot)	0.977	0.992	0.991
Anthracnose (runner)	0.865	0.885	0.883
Blossom blight	0.985	0.983	0.986
Gray mold (fruit)	0.881	0.905	0.904
Leaf spot	0.932	0.940	0.944
Powdery mildew (fruit)	0.924	0.958	0.956
Powdery mildew (leaf)	0.830	0.822	0.844
mAP	0.906	0.926	0.928

respectively, listed in columns 1 and 2 of **Table 2**. Note that we strictly split the set of images into training and testing sets with a ratio of 4:1 (5423:1807). **Table 2** only counts the number of bounding boxes. There may be more than one bounding box in an image. During the training, the online augmentation technique is applied to avoid overfitting by taking geometric transforms of horizontal/vertical flips and resizing, color jittering, blurring, and mosaicking. The total number of disease categories in this disease detection step was eight, and the results of classification were given one of the disease classes with proper bounding boxes. The training started from the weight parameters pretrained on the PlantNet in LifeCLEF 2017 dataset (Heredia, 2017), with the learning rate set to 0.002 and training for 180,000 iterations. To avoid local optimization, the learning rate was reduced by 10% at 30,000/50,000/130,000 iterations. The momentum was set to 0.9, and the stochastic gradient descent optimizer was used to minimize the difference from the ground truth. For better understanding, **Figure 5** shows several example samples used to train disease object detection.

Training Deep Metric Learning (DML) With Softmax and K-Nearest Neighbor (*K*-NN) Classifier

For the DML with the softmax classifier, we used the same training/test dataset that we used for the first object detection stage. To increase the training data, the same augmentation techniques were taken as in the first stage. The increased number of images of the extended bounding box can be seen in column 3 of **Table 2**, which include additional normal (flower, fruit, leaf, and runner) objects so that the embedded features can be learned differently from disease symptoms. In addition, the training of the CNN backbone started from the weight pretrained by the ImageNet dataset. We trained the network in 300 epochs with a batch size of 128. The learning rate was set to $1e^{-5}$ and $1e^{-4}$ for the backbone network and the classifier head, respectively. We used the Adam optimizer and the semi-hard margin sampling threshold set to 0.01.

After training the DML, we took the 256-dimensional features for reference images, which include eight known strawberry diseases with normal leaf, fruit, runner, and flower, and unknown diseases, and selected samples are shown in **Figure 6**. The unknowns are not included in the training by the DML with the softmax classifier for the second stage, but the

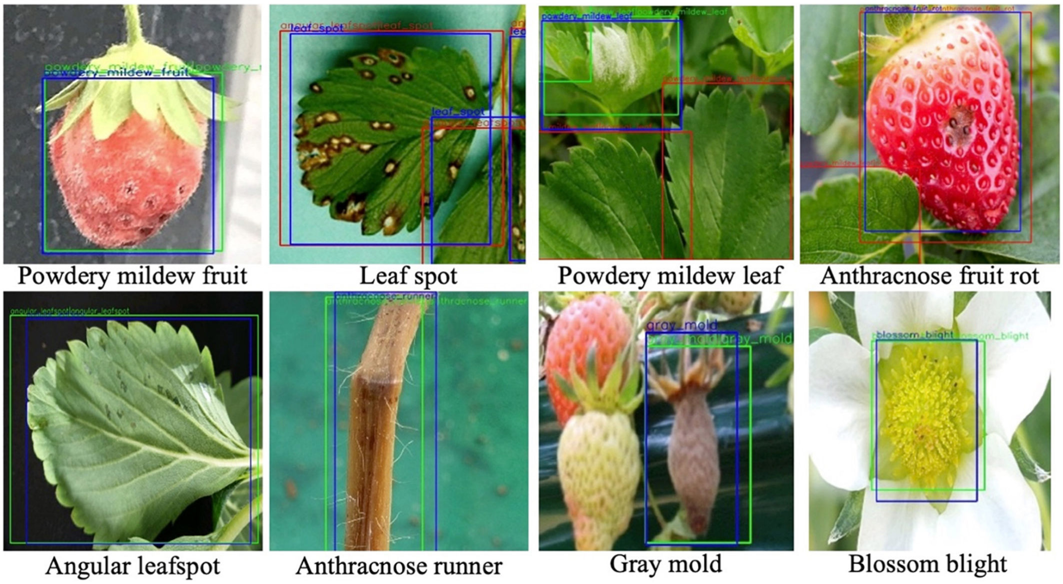


FIGURE 7 | Disease detection results from object detection and post-filter. Objects are annotated by different box colors and prediction labels. Blue bounding boxes are the ground truth annotation. Detected bounding boxes are labeled by "A| B" with two categories; "A" is the prediction result in the first stage, after which the detected area is cropped into patches and sent to the DML and given prediction label B. Green boxes mean prediction labels A and B are the same, otherwise they are red.

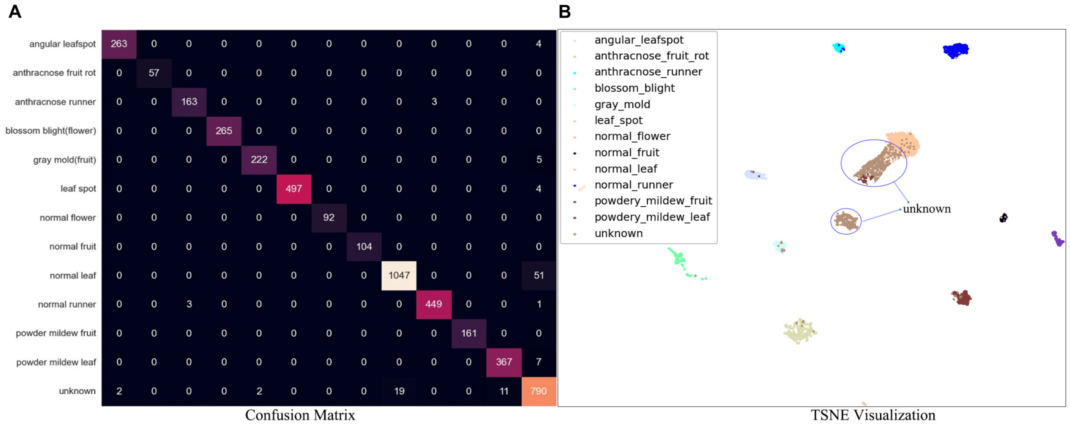


FIGURE 8 | (A) Confusion matrix of DML-based *K*-NN classifier. **(B)** TSNE visualization result for test data.

TABLE 4 | Reduced confusion matrix.

Category	Diseases	Normal	Unknowns	Recall (%)
Diseases	1,999	3	4	99.7
Normal	3	1,692	19	98.7
Unknowns	20	52	790	91.7
Precision (%)	98.9	96.9	97.2	97.8(Accuracy)

embedded features for unknowns are taken to build the *K*-NN classifier after training.

Results of Disease Detection

Table 3 presents the final results that explain the effect of post-filter. The results of the first stage of FPN-based Faster R-CNN and the second stage of classifiers are measured by average precision (AP) for each disease, and overall performance is obtained in mAP. The detection performance is found to be better for anthracnose (fruit rot) and blossom blight but comparatively worse for angular leafspot, anthracnose (runner), and powdery mildew (leaf). This is why the appearance of symptoms can be confused with other diseases (e.g., leafspot) or illumination reflecting on the leaves. In addition, the disease on the thin and long runner does not have sufficient resolution for it to be discriminated well, as is the case in the example of the anthracnose (runner).

When the DML with the softmax classifier was added to the object detection stage, the mAP increased approximately 2%, as can be seen in the third column of **Table 3**, but two diseases showed a slight degree of performance decrease: blossom blight and powdery mildew (leaf). In our conjecture, this is caused by the dislocation of bounding boxes enclosing the disease symptom in the first object detection stage, even though the enlarged bounding box is fed into the post-filter. In this case, there could be an erroneous decision in the second stage because the input image has never been experienced in the training phase.

However, when the two decisions from the softmax and *K*-NN classifiers are combined by Eq. (5), the AP performance for each disease was increased. As listed in the last column of **Table 3**, the effect of the combined decision was not significant, but there was a consistent performance increase for all diseases. **Figure 7** shows the disease detection results from the Fast R-CNN object detection followed by post-filter. A red box means a different prediction result in object detection and DML post-filter, and a green box means the two decisions are the same. The object detector finds potential objects well if the detected object is distinct from the background. However, the detector may give a false prediction label if the background is complex. For example, for the "powdery mildew leaf" in **Figure 7**, the network misdetected a normal leaf as a powdery mildew leaf, and the difference between these two categories is that the disease-infected leaves are covered in snow-white fungus, but the reflection of light on leaves shares similar features. The DML post-filter focused on the local context and successfully corrected the false detected object.

TABLE 5 | Strawberry images for field testing.

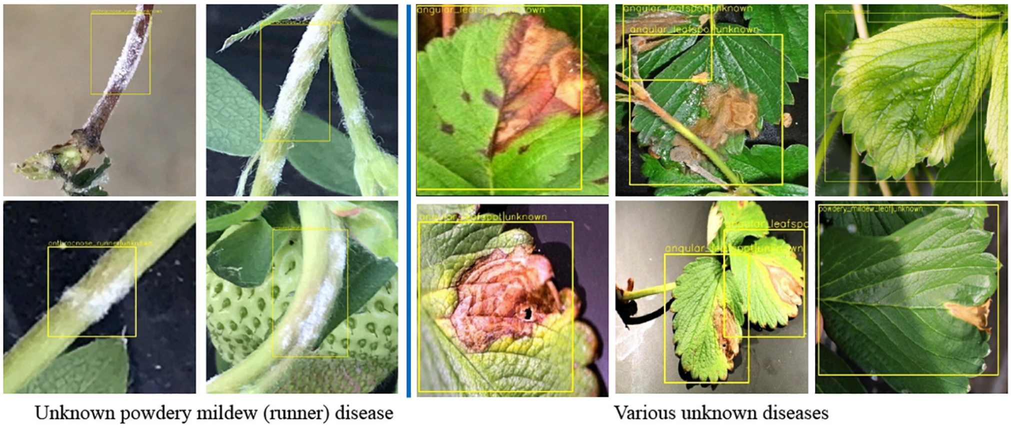
Location	Disease	# of images
Chugbuk chongju	Blossom blight	24
Chungnam non-san	Angular leafspot	36
Jeonbuk wanju	Blossom blight	167
	Gray mold (flower)	54
	Anthracnose (runner)	47
	Powdery mildew (fruit)	63
	Powdery mildew (leaf)	42
	Powdery mildew (runner)	24*
Total		457

**Trained system has never experienced disease.*

For separated DML followed by the *K*-NN classifier, the performance has been visualized by a confusion matrix, which is shown in **Figure 8A**. Note that the separate stage can be used for the classifier of ROI of the symptoms. For example, a picture of disease-like symptoms can be taken and a manual ROI can be denoted without using an automatic disease detection model such as Faster R-CNN, after which its class can be obtained from this separate *K*-NN-based classifier. The overall accuracy

TABLE 6 | Field test results of known disease detection.

Disease	BBox	Performance (AP)		
		Faster R-CNN	Faster R-CNN + softmax classifier	Faster R-CNN + *K*-NN combined decision
Angular leafspot	75	0.934	0.943	0.939
Blossom blight (f)(flower)	195	0.994	0.993	0.996
Anthracnose runner	161	0.853	0.866	0.913
Gray mold (fruit)	63	0.949	0.958	0.951
Powdery mildew fruit	48	0.881	0.915	0.931
Powdery mildew leaf	78	0.848	0.902	0.893
Total	620	0.909	0.930	0.937



Unknown powdery mildew (runner) disease

Various unknown diseases

FIGURE 9 | Detected unknown diseases.

of the separate *K*-NN classifier was 97.7% for the test data in the last column of **Table 4**, the summarized confusion matrix. In **Table 4**, the average recall and average precision were 96.7 and 97.7%, respectively. Again, a few instances of angular leafspot, gray mold (flower), and powdery mildew (leaf) were misclassified as unknowns. In addition, several normal (runners) were misclassified as anthracnose disease. Some unknown symptoms were confused with disease classes including angular leafspot, leafspot, gray mold (fruit), powdery mildew (leaf), and normal parts. Note that it is difficult to discern leafspot and angular leafspot from disorders on a leaf. For the same reason as in the object detection, there were several instances of confusion of disease classes of gray mold (flower), powdery mildew (leaf), and anthracnose (runner).

Figure 8B shows the t-SNE of the embedded features after DML. It is evident that almost all the classes of known diseases and normal parts are well separated, but the classes that confuse (**Figure 8B** and **Table 4**) are slightly overlapping, as shown in **Figure 8B**.

Final Field Test With Unseen Data

To validate the proposed scheme, strawberry images were captured from three greenhouses at different locations, and we used these images to construct the dataset as in **Table 5**. Note that only six known diseases are included, because at that time, leafspot and anthracnose (fruit rot) were hard to find. In the table, powdery mildew (runner) can be treated as unknown, because it was not considered in the training of any building block of our scheme. **Table 6** presents the mAP results of known diseases. It can be seen that the overall performances are increasing from the first object detection to the final combined decision of the softmax and *K*-NN classifiers. For unknown powdery mildew (runner), 19 images were detected with the proper bounding box out of 24 images. As shown in the left part of **Figure 9** (left), all the diseases were detected as anthracnose (runner) in the first object detection stage but corrected to unknowns in the *K*-NN classifier. Moreover, as shown in the right part of **Figure 9**, the disorders on the leaf are corrected to unknowns in the *K*-NN classifier after having been wrongly detected in the first stage as one of the leaf diseases.

CONCLUSION

This study has proposed a simple but effective strawberry disease detection scheme with unknown diseases that can produce reasonable performance. In the proposed scheme, the known strawberry diseases are better detected with DML-based classifiers, as are the unknown diseases that have certain symptoms. We have assumed that, in the training process, the unknowns are partly known. The pipeline of our proposed scheme consists of two stages: the first is an object detection stage with known disease classes, while the second is the DML-based post-filtering stage. The second stage has two different types of classifiers: softmax classifiers for only known diseases and the *K*-NN classifier for known and unknown diseases. In training the first stage and DML-based softmax classifier, we have only used the known samples of strawberry diseases. Then, we included the known unknown training samples to construct the *K*-nearest neighbor classifier. The final decision for known diseases has been made based on the combined results of the two classifiers, while unknowns have been detected from the *K*-NN classifier.

The experimental results showed that the DML-based post-filter was effective at improving the performance of known disease detection in terms of mAP. Furthermore, the separate DML-based *K*-NN classifier provided high recall and precision with respective average values of 96.7 and 97.7%, showing it could be exploited as an ROI classifier. For the real field data, the proposed scheme achieved a high mAP of 93.7% to detect seven classes (six known diseases and one unknown) of strawberry disease, and it also achieved reasonable detection results for unknowns. These results imply that the proposed scheme can be applied to find disease-like symptoms due to real known and unknown diseases or disorders for any kind of plant, including strawberry.

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

JL supervised the whole project and wrote the original draft of the manuscript. KJ responded to collect the data resource and organized the database. JY performed the experiment and statistical analysis. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by the Brain Korea 21 PLUS Project.

REFERENCES

Bastan, M., Wu, H.-Y., Cao, T., Kota, B., and Tek, M. (2019). Large scale open-set deep logo detection. *arXiv* [Preprint]. doi: 10.48550/arXiv.1911.07440

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv* [preprint]. doi: 10.48550/arXiv.2004.10934

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers", in: *European Conference on Computer Vision*. (New York, NY: Springer), 213-229.

Chopra, S., Hadsell, R., and Lecun, Y. (2005). "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Diego, CA: IEEE), 539–546.

Dananjayan, S., Tang, Y., Zhuang, J., Hou, C., and Luo, S. (2022). Assessment of state-of-the-art deep learning based citrus disease detection techniques using annotated optical leaf images. *Comput. Electron. Agric.* 193:106658.

Dhamija, A., Gunther, M., Ventura, J., and Boult, T. (2020). "The overlooked elephant of object detection: Open set," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (Snowmass, CO: IEEE), 1021–1030.

Fehérvári, I., and Appalaraju, S. (2019). "Scalable logo recognition using proxies," in *2019 IEEE Winter Conference on Applications of Computer Vision*, (Waikoloa, HI: IEEE), 715–725.

Fuentes, A., Yoon, S., and Park, D. S. (2020). "Deep Learning-Based Techniques for Plant Diseases Recognition in Real-Field Scenarios," in *Advanced Concepts for Intelligent Vision Systems. ACIVS 2020. Lecture Notes in Computer Science*, eds J. Blanc-Talon, P. Delmas, W. Philips, D. Popescu, and P. Scheunders (Cham: Springer), doi: 10.1007/978-3-030-40605-9_1

Geng, C., Huang, S.-J., and Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3614–3631. doi: 10.1109/TPAMI.2020.2981604

Heredia, I. (2017). "Large-scale plant classification with deep neural networks," in *Proceedings of the Computing Frontiers Conference*, (New York, NY: Association for Computing Machinery), 259–262.

Janarthan, S., Thuseethan, S., Rajasegarar, S., Lyu, Q., Zheng, Y., and Yearwood, J. (2020). Deep metric learning based citrus disease classification with sparse data. *IEEE Access* 8, 162588–162600.

Ji, Z., Xia, F., Xie, X., Wang, Z., Jin, S., and Yan, K. (2021). A Novel Computational Framework for Precision Diagnosis and Subtype Discovery of Plant With Lesion. *Front. Plant Sci.* 12:789630. doi: 10.3389/fpls.2021.789630

Joseph, K., Khan, S., Khan, F. S., and Balasubramanian, V. N. (2021). Towards open world object detection. *arXiv* [preprint]. doi: 10.48550/arXiv.2103.02603

Kaya, M., and Bilge, H. Ş (2019). Deep metric learning: A survey. *Symmetry* 11:1066,

Kim, B., Han, Y.-K., Park, J.-H., and Lee, J. (2021). Improved vision-based detection of strawberry diseases using a deep neural network. *Front. Plant Sci.* 11:559172. doi: 10.3389/fpls.2020.559172

Li, D., and Tian, Y. (2018). Survey and experimental study on metric learning methods. *Neural Netw.* 105, 447–462. doi: 10.1016/j.neunet.2018.06.003

Lin, T., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017a). Focal loss for dense object detection. *arXiv* [Preprint]. doi: 10.1109/TPAMI.2018.2858826

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017b). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Piscataway: IEEE), 2117–2125.

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

Mahdavi, A., and Carvalho, M. (2021). A survey on open set recognition. *arXiv* [Preprint]. doi: 10.1109/AIKE52691.2021.00013

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Netw.* 113, 54–71.

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*, (Las Vegas, NV: IEEE), 779–788.

Redmon, J., and Farhadi, A. (2017). "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu: IEEE), 7263–7271.

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

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

Schlachter, P., Liao, Y., and Yang, B. (2019a). "Open-set recognition using intra-class splitting," in *2019 27th European signal processing conference*, (A Coruna, Spain: IEEE).

Schlachter, P., Liao, Y., and Yang, B. (2019b). "Deep one-class classification using intra-class splitting," in *2019 IEEE Data Science Workshop*, (New York: IEEE), 100–104.

Schlachter, P., Liao, Y., and Yang, B. (2020). Deep Open Set Recognition Using Dynamic Intra-class Splitting. *SN Comput. Sci.* 1:77.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (New York City: IEEE), 815–823.

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*, (Piscataway: IEEE), 10781–10790.

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. (2021). Open-set recognition: A good closed-set classifier is all you need. *arXiv* [Preprint]. doi: 10.48550/arXiv.2110.06207

Xiao, J.-R., Chung, P.-C., Wu, H.-Y., Phan, Q.-H., Yeh, J.-L. A., and Hou, M. T.-K. (2021). Detection of strawberry diseases using a convolutional neural network. *Plants* 10:31. doi: 10.3390/plants10010031

Zhang, S., Wen, L., Bian, X., Lei, Z., and Li, S. Z. (2018). "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway: IEEE), 4203–4212.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst*. 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865

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.

ORIGINAL RESEARCH
published: 05 July 2022
doi: 10.3389/fpls.2022.899754

The Detection Method of Potato Foliage Diseases in Complex Background Based on Instance Segmentation and Semantic Segmentation

*Xudong Li[1,2], Yuhong Zhou[1,2], Jingyan Liu[1,2], Linbai Wang[2], Jun Zhang[2] and Xiaofei Fan[1,2]**

[1] State Key Laboratory of North China Crop Improvement and Regulation, Baoding, China, [2] College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding, China

Edited by:
Lei Shu,
Nanjing Agricultural University, China

Reviewed by:
Alireza Sanaeifar,
Zhejiang University, China
Ahmad Al-Mallahi,
Dalhousie University, Canada

***Correspondence:**
Xiaofei Fan
hbaufxf@163.com

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: 19 March 2022
Accepted: 06 June 2022
Published: 05 July 2022

Citation:
Li X, Zhou Y, Liu J, Wang L, Zhang J and Fan X (2022) The Detection Method of Potato Foliage Diseases in Complex Background Based on Instance Segmentation and Semantic Segmentation. Front. Plant Sci. 13:899754. doi: 10.3389/fpls.2022.899754

Potato early blight and late blight are devastating diseases that affect potato planting and production. Thus, precise diagnosis of the diseases is critical in treatment application and management of potato farm. However, traditional computer vision technology and pattern recognition methods have certain limitations in the detection of crop diseases. In recent years, the development of deep learning technology and convolutional neural networks has provided new solutions for the rapid and accurate detection of crop diseases. In this study, an integrated framework that combines instance segmentation model, classification model, and semantic segmentation model was devised to realize the segmentation and detection of potato foliage diseases in complex backgrounds. In the first stage, Mask R-CNN was adopted to segment potato leaves in complex backgrounds. In the second stage, VGG16, ResNet50, and InceptionV3 classification models were employed to classify potato leaves. In the third stage, UNet, PSPNet, and DeepLabV3+ semantic segmentation models were applied to divide potato leaves. Finally, the three-stage models were combined to segment and detect the potato leaf diseases. According to the experimental results, the average precision (AP) obtained by the Mask R-CNN network in the first stage was 81.87%, and the precision was 97.13%. At the same time, the accuracy of the classification model in the second stage was 95.33%. The mean intersection over union (MIoU) of the semantic segmentation model in the third stage was 89.91%, and the mean pixel accuracy (MPA) was 94.24%. In short, it not only provides a new model framework for the identification and detection of potato foliage diseases in natural environment, but also lays a theoretical basis for potato disease assessment and classification.

Keywords: potato foliage disease, convolutional neural network, image recognition, instance segmentation, semantic segmentation

INTRODUCTION

Potato is one of the world's four important food crops, one of the 10 most popular nutritious and healthy foods, as well as a high-yield crop with developmental prospects. Due to its high yield and stability, wide adaptability, full nutritional content, and long industrial chain, it has been highly valued in the world (Qu et al., 2005). The early blight and late blight, as the most destructive foliage diseases of potato crops (Tsedaley, 2014; Yellareddygari et al., 2018), could cause major losses in most potato-growing areas in the world. On potato leaves, late blight appears as light green or olive green areas that rapidly turn brownish-black, water-soaked, and oily. Likewise, early blight is round or irregular, which shows dark brown or black spots. Overall, early blight and late blight can occur in all stages of potato growth (Da Silva Silveira Duarte et al., 2019). To control and prevent diseases effectively and timely, it is of great significance to identify and detect the diseases of potato leaves.

In general, the traditional diagnosis of crop diseases is performed by experienced experts, but manual diagnosis is inefficient, subjective, and unsuitable for large regional scenarios. Besides, traditional diagnostic techniques of crop diseases tend to include polymerase chain reaction (PCR), fluorescence *in situ* hybridization (FISH), enzyme-linked immunosorbent assay (ELISA), thermal imaging, and hyperspectral imaging (Fang and Ramasamy, 2015; Xie et al., 2015; Madufor et al., 2018). In the real-life production, farmers need simple, rapid, and accurate ways to identify potato diseases. Therefore, it is crucial to develop a fast, low-cost, time-saving, and labor-saving automatic identification system for potato diseases.

With the advancement in computer vision, artificial intelligence, and machine learning technology, it has promoted the development and implementation of automatic disease recognition technology. For example, Adhikari et al. (2018) used Fast R-CNN (Ren et al., 2017) and R-FCN (Fuentes et al., 2017) to detect diseases of fruit trees, vegetable crops, and other crops, and confirmed good results. In addition, Zhang et al. (2018) used the PlantVillage dataset combined with transfer learning to identify nine tomato diseases. Among them, the models with ResNet as the backbone network have the best recognition effect, with an accuracy of 97.28%. Furthermore, Cheng et al. (2017) used ResNet and AlexNet to identify crop pests, and proved that ResNet101 could achieve the best results, with an accuracy of 98.67%. Khan et al. (2020) proposed a classification method of cucumber foliage disease, which was based on an improved saliency method and deep feature selection. Compared with the existing single-feature selection methods, the deep feature selection method has better performance. To identify cucumber leaf lesions, Wang et al. (2021) put forward a network model fused with UNet and DeepLabV3+, and verified that semantic segmentation has achieved good results for leaf lesions. Apart from that, Fan and Li (2019) proposed a detection method based on key feature points, which could quickly detect the disease in regions of interest by combining with color and texture features. Although this method recognizes 10 types of potato diseases with high speed and high accuracy, it does not have good performance for the recognition in complex environment. Brahimi et al. (2017) trained a convolutional neural network (CNN) composed of nine tomato diseases, with the accuracy of the final model reaching 99.1%. Then, Yang et al. (2020) proposed a potato disease leaf recognition method based on the combination of deep CNN and composite feature dictionary, adopted Faster R-CNN to detect the disease areas, and constructed a composite feature dictionary through extraction of image features. The disease recognition model was trained by support vector machine, and its average recognition accuracy could reach 84.16%. Nevertheless, the image background was relatively simple. To solve the difficult problem of locating and identifying typical potato disease regions under natural conditions, Xiao and Liu (2017) put forward an adaptive feature fusion and rapid recognition method for typical potato diseases. As proved by the recognition experiment of three typical potato diseases, the average recognition rate of the modified adaptive feature fusion method is at least 1.8 percentage points higher than that of the traditional adaptive method. Meanwhile, the average recognition rate of the recognition method is 95.2%, but it is slower than that of deep learning. Additionally, Krishnaswamy Rangarajan and Purushothaman (2020) achieved good results in classifying eggplant diseases, used multiclassification support vector machine (MSVM), and adopted VGG16 as a feature extractor in the eighth convolutional layer. Combining visual object recognition with language generation models, the detailed information about plant anomaly symptoms and scene interactions could be generated (Fuentes et al., 2019). In the task of identifying tomato pests and diseases, the accuracy of the method achieved 92.5%.

Previous studies have applied deep learning technology to the detection, segmentation, or classification of different crop diseases. Beyond that, some studies have proposed to classify different diseases that are found in leaves, and the accuracy rate is generally >90%. At present, there are the following problems in the crop disease recognition and disease spot detection: (1) The image collection in previous studies was often a single leaf, and there were few studies on the segmentation of images containing multiple leaves. (2) Traditional recognition methods have poor recognition rate for plant foliage disease. (3) The effect of plant leaf disease identification on small targets is poor.

Based on the existing research, this study proposed a method of detecting potato diseases in a complex background, which combines instance segmentation, classification model, and semantic segmentation. The main contents of this study are as follows:

(1) A three-stage potato leaf disease detection model based on deep learning was proposed. While segmenting the potato leaves and diseases accurately, this model could provide a basis for establishing a potato leaf disease detection system.

(2) By adopting the three-stage model of instance segmentation, classification model, and semantic segmentation, the advantages of each model were explored. Compared with

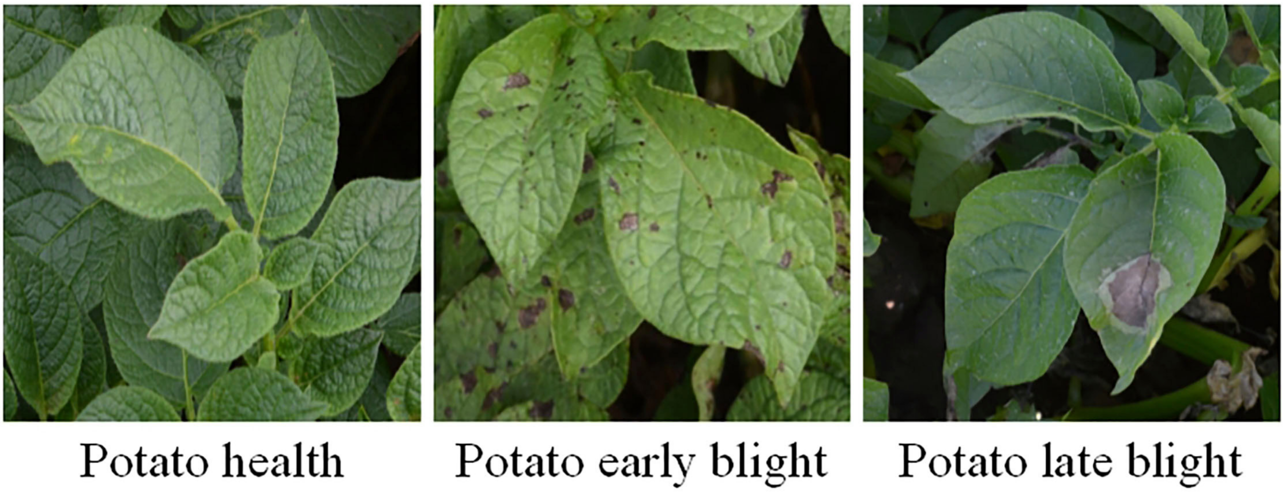


FIGURE 1 | Images of potato leaves.

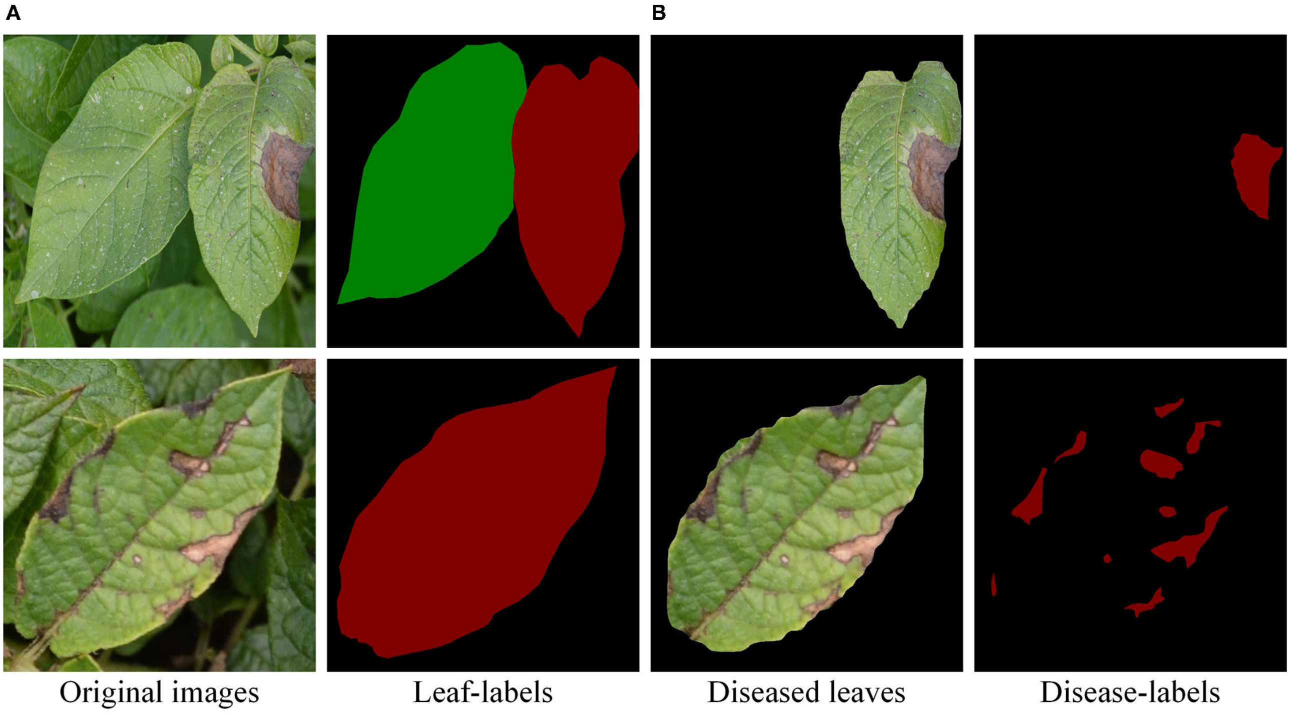


FIGURE 2 | Leaf-labels and disease-labels. **(A)** The individual leaf separated from the complex background. **(B)** The leaf scab was marked.

single model detection, the three-stage model in this study has good performance.

(3) The detection of potato leaf diseases in complex backgrounds was achieved, and the percentage of disease area to leaf area was calculated from the segmented disease area. Overall, this experiment could provide a technical basis for the classification and accurate control of plant diseases in the future.

MATERIALS AND METHODS

Data Collection

In this study, potato leaves were collected at the potato experimental site of Hebei Agricultural University, which was a representative planting site in northern China (Weichang and Fengning, Chengde City, Hebei Province). Besides, Nikon D7100 camera with a resolution of 6,000 × 4,000 pixels was used to

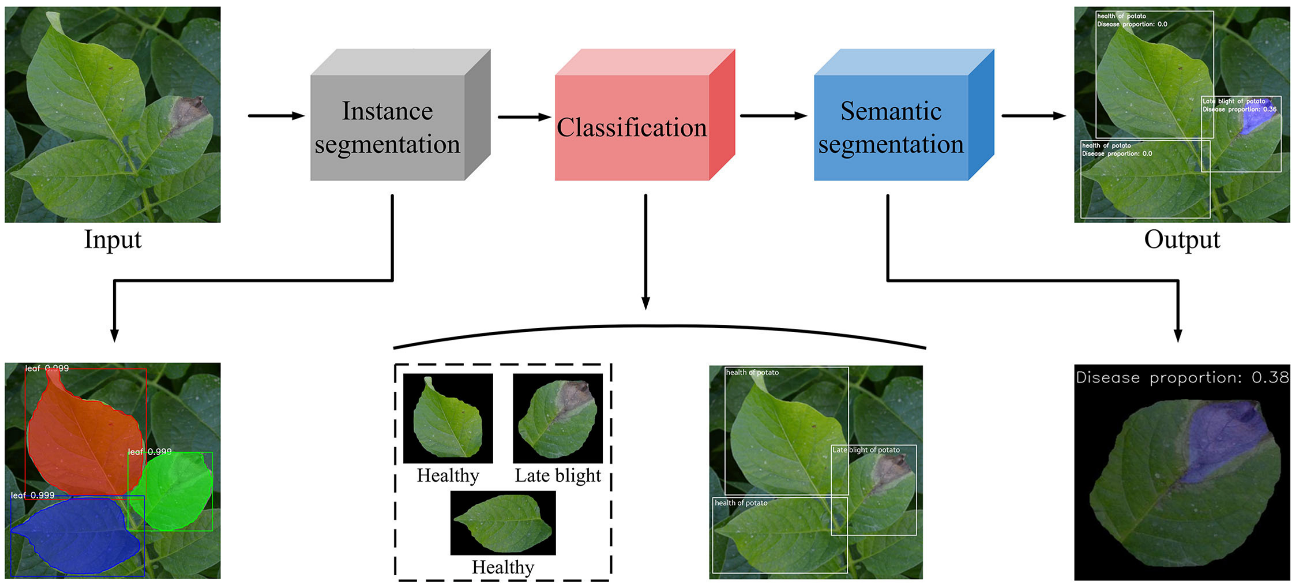


FIGURE 3 | The identification and classification of potato leaf process. This figure shows the whole experimental progress, from the input to the output.

photograph potato leaves, and it was set to close-up mode with automatic adjustment of focus, aperture, and white. The distance between the camera and the potato plant was about 50 cm, and the images were collected in a vertical manner. The three types of potato leaves are displayed in **Figure 1**.

Data Processing

A total of 500 original images had been collected, including healthy leaves, early blight leaves, and late blight leaves. The size of the original images was adjusted to 800 × 800 pixels. Then, the leaves and diseases were marked by the labelme software. As shown in **Figure 2A**, the mask images were generated. Apart from that, the accuracy of the model was evaluated by the mask image marked manually. Specifically, the experimental method in this study was divided into three stages. In the first stage, the 400 images were divided into the training set and validation set, respectively, according to the ratio of 4:1 and test set with 100 images after training. The second stage uses image enhancement to obtain 1,800 images, which are divided into training set and validation set according to the ratio of 4:1. The test set consists of 150 original images, including 50 pieces of each of the three types of leaves. In the third stage [as shown in **Figure 2B**], a total of 632 labeled early blight leaves and late blight leaves images were divided into training set and validation set of the semantic segmentation model in a ratio of 4:1. The test set consists of 50 original images.

Data Enhancement

Convolutional neural networks require enough data, and the training accuracy of the model could be increased by the amount of data. Therefore, in the second stage of this experiment, the samples were enhanced by image rotation. In addition, the original images were rotated according to the probability of 0.8, with the maximum left-hand angle of 10 and the maximum right-hand angle of 10. In addition, the left and right images were swapped according to the probability of 0.5. The images were zoomed in and out in accordance with the probability of 0.8. In brief, these image enhancement methods simulate the changes in the actual image acquisition angle, direction and distance, increase the diversity of training samples, and improve the robustness and generalization of the model.

Computer Configuration Parameters

Windows 10 operating system was applied in this study. Specifically, the computer memory is 16 GB, the CPU model is Intel Core (TM) i5-10400f, and the frequency is 2.90 GHz. Meanwhile, the graphics processor model is NVIDIA GeForce GTX 1660s, and the video memory is 6 GB. Software environment used in the experiment is Tensorflow and Keras (Python 3.6).

Model Evaluation Indicators

To test the performance of the model used in this study (e.g., segmentation, classification model, and semantic segmentation), *Precision* (%), Mean Intersection over Union (*MIoU*, %), *Accuracy* (%), and average pixel accuracy (*MPA*, %) were selected as the indicators. To explain the evaluation index formula conveniently, it was assumed that the data set had a total of $k + 1$ categories. Moreover, $\underline{P_{ij}}$ represents the number of pixels that category i is predicted into category j, P_{ii} represents the number of pixels that are predicted correctly, while P_{ij} and P_{ji} represent the number of false-negative and false-positive pixels, respectively.

Precision and Accuracy

In the formula mentioned below, *TP* denotes true positive, *FP* denotes false positive, and *FN* denotes false negative. *Precision* represents the proportion of the correct prediction that is positive to all predictions that are positive. *Accuracy* represents the proportion of all data that are correctly predicted.

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

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

MIoU and MPA

Pixel-based accuracy (PA, %) calculation is the basic index of semantic segmentation performance evaluation, and *MPA* is the average pixel accuracy. The average intersection ratio is a commonly used measurement index for semantic segmentation and target detection, which is often adopted to evaluate the overlap ratio of the predicted object and the target object. Compared with the pixel accuracy, the average intersection ratio will provide more information, such as the completeness of the predicted target and the coincidence with the actual target.

$$MPA=\frac{1}{k+1}\sum_{i=0}^{k}\frac{P_{ii}}{\sum_{j=0}^{k}P_{ij}}$$

$$MIoU=\frac{1}{k+1}\sum_{i=0}^{k}\frac{P_{ii}}{\sum_{j=0}^{k}P_{ij}+\sum_{j=0}^{k}P_{ji}-P_{ii}}$$

Test Model

Mask R-CNN Model

A series of region-based CNN algorithms (He et al., 2017; Ren et al., 2017) are the most representative methods in the target detection. Mask R-CNN, as a relatively novel achievement, can classify, identify, and segment the targets in images. In this study, the backbone network that combines ResNet (He et al., 2016) and FPN (Long et al., 2015) was used to extract features of potato leaves. Among them, the ResNet could sequentially extract low-level features (e.g., edges and corners) and high-level features (e.g., leaves and ground), which could form five layers of feature maps in different sizes and dimensions. If the last layer of features in the ResNet network is used as the output of the network, it is difficult to detect the relatively small leaf features due to its low resolution. Therefore, the FPN network was used to fuse the feature maps from the bottom to the high level, and the features extracted from each layer of the ResNet network were fully used. Apart from that, the feature map extracted from the backbone architecture was input to the regional candidate network. The regional candidate network is a typical binary network, the function of which is to divide the image into two categories, namely, the target leaf and the background. Besides, the plant leaves are boxed out separately in boxes that fit the size of the leaves as closely as possible. At this time, only the approximate region containing the target leaves and the background could be distinguished, and it is impossible to conduct detailed species classification and leaf segmentation of the target leaves. Through the region candidate network, one or more regions containing target blades could be obtained, which are input into ROIAlign to pool into a feature map with a fixed size, and then input into two branches, respectively. One of the branch networks performs target leaf identification by means of a region of interest classifier and a border regressor, both of which include one fully connected layer. One fully connected layer acts as the ROI classifier to classify the ROI into specific plant categories, while the other fully connected layer is used as the border regressor to adjust the center point position and aspect ratio of the ROI, to detect the target leaves more accurately. Another branch network is a segmentation mask generation network consisting of a fully convolutional network, which generates a mask of the same size and shape as the target leaf to segment the target leaf image. Finally, the recognition and results are combined to obtain an image that contains the target leaf class and a segmentation mask that is consistent with the size and shape of the target leaf.

Classification Model

The essence of the VGG16 model is an enhanced version of the AlexNet structure, which focuses on the depth of the CNN design. In addition, each convolution layer is followed by a pooling layer. VGG16 has five convolution layers, each with two or three convolution layers. To better extract feature information, this experiment uses three convolutional layers per segment. Beyond that, a maximum pooling layer is connected at the end of each segment to reduce the picture size. The number of convolution kernels in each segment is the same, and the closer they are to the fully connected layer, the more are the convolution kernels. At the same time, the number of convolution kernels in each segment is the same. In general, the closer they are to the fully connected layer, the more are the convolution kernels, and the smaller is the corresponding picture size. As for the VGG network, it uses a smaller convolution kernel, which reduces the number of parameters and saves computing resources. Due to the large number of layers, the convolution kernel is relatively small, so that the entire network has a better feature extraction effect.

The InceptionV3 network is a deep convolutional network developed by Google. Compared with the traditional Inception structure, the V3 version used in this study decomposes the large convolution kernel into small convolution kernels. For example, two 3 × 3 convolution kernels are used to replace the original 5 × 5 convolution kernel, which reduces the number of operations of the model. The BN convolutional layer (Batch Normalization) is added to the classification assistant to improve the accuracy of the model, and the Batch Normalization method is used to make the model perform data normalization preprocessing before each iteration training, which avoids each iteration of the network. All will adapt to different data distributions, which greatly shortens the training time of the model.

The ResNet50 model solves the problem that the actual effect becomes worse due to the increase in network depth and width.

It is noteworthy that the deep neural network model sacrifices a large amount of computing resources, while the error rate has also increased. This phenomenon is mainly attributed to the fact that as the number of layers of the neural network increases, the disappearance of the gradient becomes increasingly obvious. The ResNet50 model adds the residual structure (i.e., an identity mapping is added), which converts the original transformation function $H(x)$ into $F(x) + x$, makes the network no longer a simple stack structure, and solves the problem of gradient disappearance. This simple stack does not add extra parameters and calculations to the network but improves the effect and efficiency of network training.

Semantic Segmentation Model

UNet (Ronneberger et al., 2015) is a semantic segmentation network based on FCN (Long et al., 2015), and its network structure is similar to FCN (fully convolutional networks). The first half of the UNet network is feature extraction, and the second half is upsampling. This structure is generally referred to as an encoder-decoder structure. In addition, the input values of this network are 512 × 512 single-channel or three-channel images. The network, as a whole, can be constructed as a codec architecture or as a systolic path and extended path. On the one hand, each step of the contraction path consists of two 3 × 3 convolutions for feature extraction. On the other hand, each step of the expansion path includes an upsampling process of the feature map, which matches and fuses with the feature map starting from the contracted path. The shallower high-resolution layer in the UNet network is used to solve the pixel localization problem, while the deeper layer is adopted to solve the problem of pixel classification.

The main feature of the PSPNet (Zhao et al., 2016) model is the use of the PSP module. The pyramid pooling module proposed in this model can aggregate the contextual information of different regions, so as to improve the ability to obtain global information. As shown by the results of experiments, such *a priori* representation (referring to the structure of PSP) is effective, and has presented excellent results on multiple data sets. The function of the PSP structure is to divide the acquired feature layers into grids of different sizes, and each grid is pooled on average. It achieves the aggregation of contextual information from different regions, thus improving the capacity to obtain global information.

The main body of the Encoder of DeepLabV3+ (Cheng et al., 2017) is DCNN with hole convolution, which can adopt the commonly used classification networks, such as ResNet, followed by Atrous Spatial Pyramid Pooling (ASPP) module with null convolution (Chen et al., 2014). Compared with the conventional convolution, the hole convolution increases the receptive field without changing the feature map, and retains more spatial detail information. The hole convolution injects "holes" into the standard convolution kernel to increase the convolution kernel. Receptive field, hole convolution uses the hole structure to expand the size of the convolution kernel, which can increase the receptive field without downsampling, while retaining the internal structure of the input data. It is mainly for the introduction of multiscale information. Compared with DeepLabV3, V3+ introduces the Decoder module, which further merges the low-level features with the high-level features to improve the accuracy of the segmentation boundary.

Three-Stage Model Structure

In this study, the potato disease identification consists of four steps (see **Figure 3**).

(1) In the first stage, potato leaves were segmented by Mask R-CNN from complex background, and the individual leaves were extracted;
(2) The segmented individual leaves were used as the input in the classification model, which could classify healthy, early blight, and late blight leaves;
(3) The single leaf extracted from the second stage was used as the input of the third stage, and the training was carried out through semantic segmentation model;
(4) The disease identified in the semantic segmentation stage was adopted as the index of disease recognition in the classification stage. In addition, the healthy leaves, early blight leaves, and late blight leaves were marked by the instance segmentation model and classification model. The proportion of the disease to the whole leaf was also marked.

RESULTS

Mask R-CNN Models

Two different backbone networks, ResNet50 and ResNet101, were used in instance segmentation. Apart from that, 100 pictures were selected to test the models. **Table 1** summarizes the results of both networks. It can be observed that the ResNet101 backbone network has a good performance, indicating that a deeper backbone network for features used in Mask R-CNN could obtain the good performance. To better evaluate the accuracy of the whole model, the AP was selected when IoU = 0.5 and IoU = 0.7. Meanwhile, the AP obtained by ResNet50 and ResNet101 was 78.21 and 81.87%, respectively. Furthermore, the Precision obtained by ResNet101 was 97.13%, which was slightly better than that obtained by ResNet50. As ResNet101 has a deeper backbone network, its accuracy in the instance segmentation is higher. For testing 100 images, the two backbone networks need to take 29 and 32 s, respectively. This is because the ResNet101 structure has a deeper network.

The results of Mask R-CNN are shown in **Figure 4**. First, masks of different colors were generated on the leaves. Second, a prediction frame was generated. Finally, the identified leaves were divided into single leaves under the black background, which

TABLE 1 | The results of Mask R-CNN model instance segmentation in potato leaves.

Backbone	AP (%)	$AP_{IoU=0.5}$ (%)	$AP_{IoU=0.7}$ (%)	Precision (%)	Time/img
ResNet50	78.21	82.63	84.25	96.73	0.29 s/img
ResNet101	81.87	86.31	85.48	97.13	0.32 s/img

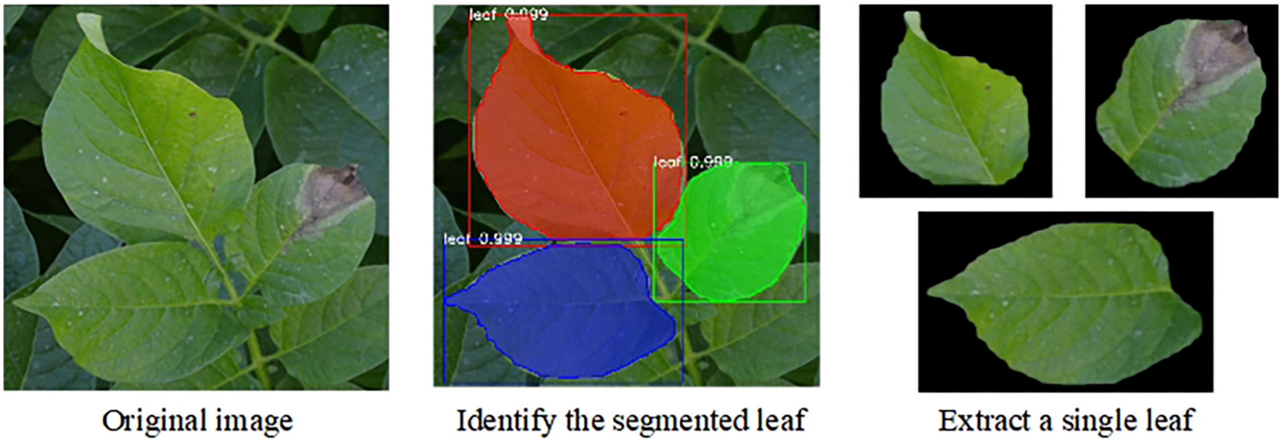


FIGURE 4 | The potato leaves segmented by Mask R-CNN and the single leaf under the black background extracted in the original image.

TABLE 2 | Accuracy of the classification model validation in the second stage.

Model	VGG16	ResNet50	InceptionV3
Accuracy/%	97.30	95.20	95.70

TABLE 4 | Comparison of the results in the semantic segmentation models.

Model	MIoU (%)	MPA (%)
UNet	89.91	94.24
PSPNet	86.08	93.19
DeepLabV3+	85.29	88.08

were used as the input of the second-stage classification model. As displayed in **Table 1**, the higher precision obtained by the models confirmed that the leaf features could be successfully detected by the models. The two backbone network structures could accurately segment the leaves.

Classification Models

The single leaf image segmented in the first stage was used as the input in this stage. Beyond that, the leaves were divided into healthy, early blight, and late blight. Additionally, the classification model of this stage utilized the cross-entropy loss function and the Adam optimizer. The batch size was 32, and the learning rate was 0.0001. If the performance of the model did not improve after three epochs, the learning rate would be reduced to continue training, and the iterations would be 150. **Table 2** presents the training accuracy of the validation set of the three models.

After the completion of the model training, 50 images were selected as the test set to verify the trained models (see the results in **Table 3**). Obviously, the Accuracy of the VGG16 network model was up to 95.33%, and the Accuracy ResNet50 and InceptionV3 were slightly lower than those of VGG16.

Identification and Detection Models of Early Blight and Late Blight

In the third stage, the single leaf image classified in the second stage was input into the three semantic segmentation models, such as UNet, PSPNet, and DeepLabV3+. **Table 4** lists the evaluation indices for the three models, which are obtained after training 150 generations. Obviously, the MIoU and MPA of UNet were higher than those of PSPNet and DeepLabV3+. This is mainly because the early blight is characterized by small area and disease dispersion, which affects the feature extraction of the models. After the completion of model training, 50 pictures of potato leaves with early blight and late blight were selected for testing. **Table 4** summarizes the test results of the three network models. It is obvious

TABLE 3 | Test results of the classification model.

Model	Number of targets (health/early blight/late blight)	Number of correct targets (health)	Number of correct targets (early blight)	Number of correct targets (late blight)	Accuracy/%
VGG16	50/50/50	48	48	47	95.33
ResNet50	50/50/50	48	46	48	94.67
InceptionV3	50/50/50	47	47	46	93.33

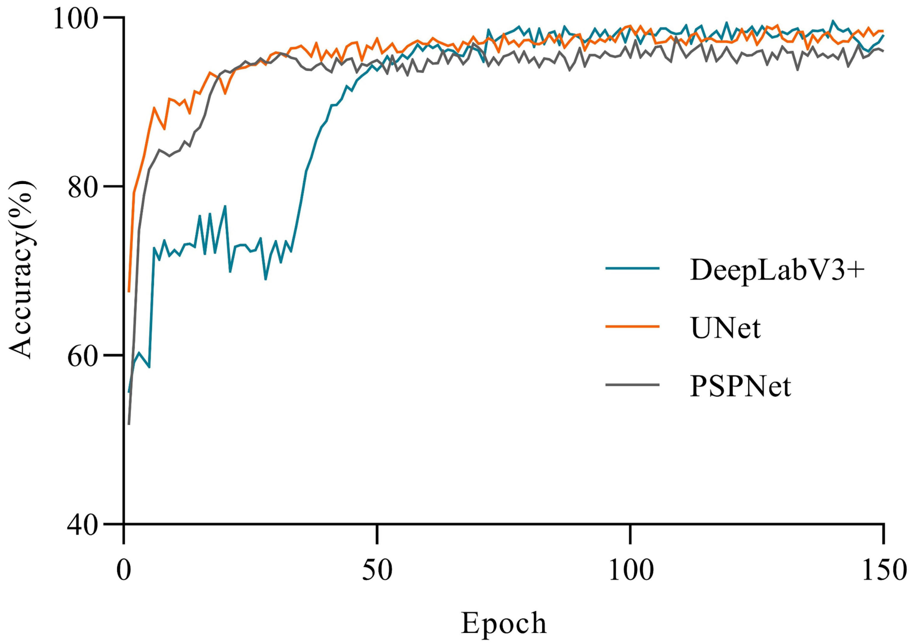


FIGURE 5 | Comparison of the variations of accuracy.

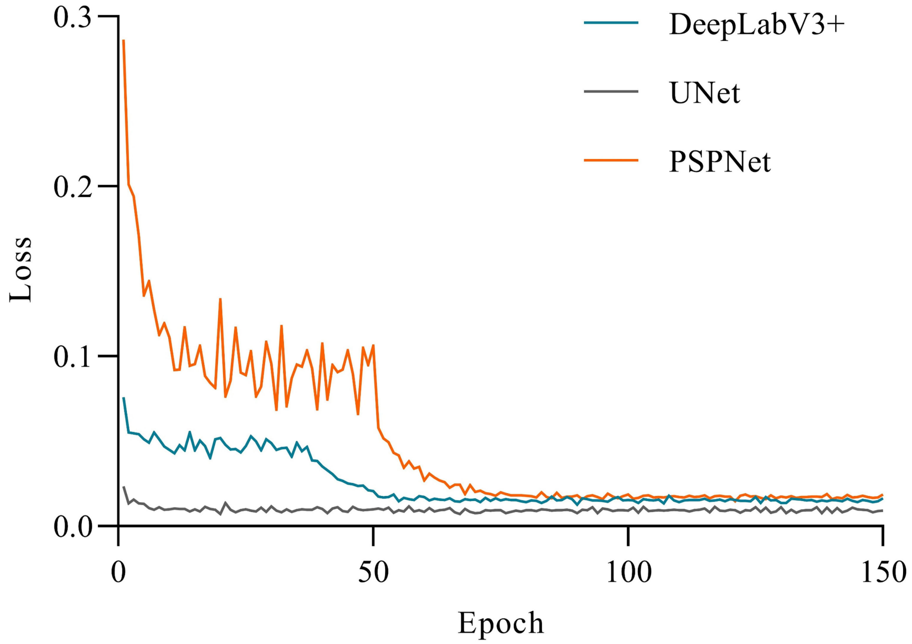


FIGURE 6 | Comparison of the variations of loss.

that the MIoU and MPA of UNet were 89.91 and 94.24%, respectively, which were better than PSPNet and DeepLabV3+. Among them, the MIoU and MPA obtained by DeepLabV3+ were relatively low, which may be due to the addition of hole convolution to the DeepLabV3+ network. Although the receptive field of the convolution layer was increased, some feature information were missed, and the area of some lesions is small, which affects the performance of DeepLabV3+. Compared with PSPNet and DeepLabV3+, UNet uses a more concise network structure and achieves better results. Therefore, UNet

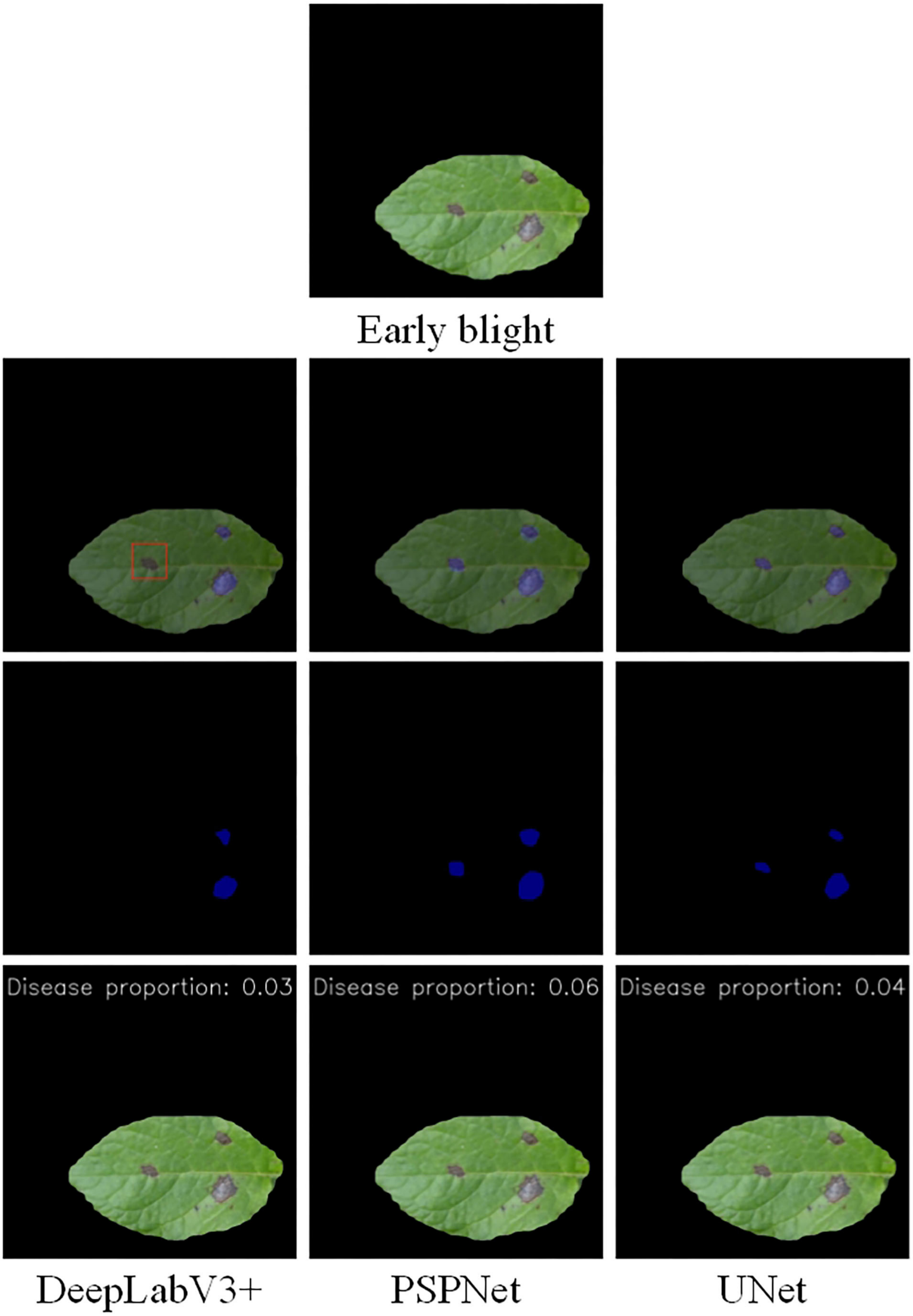


FIGURE 7 | Semantic segmentation results of early blight under the three models.

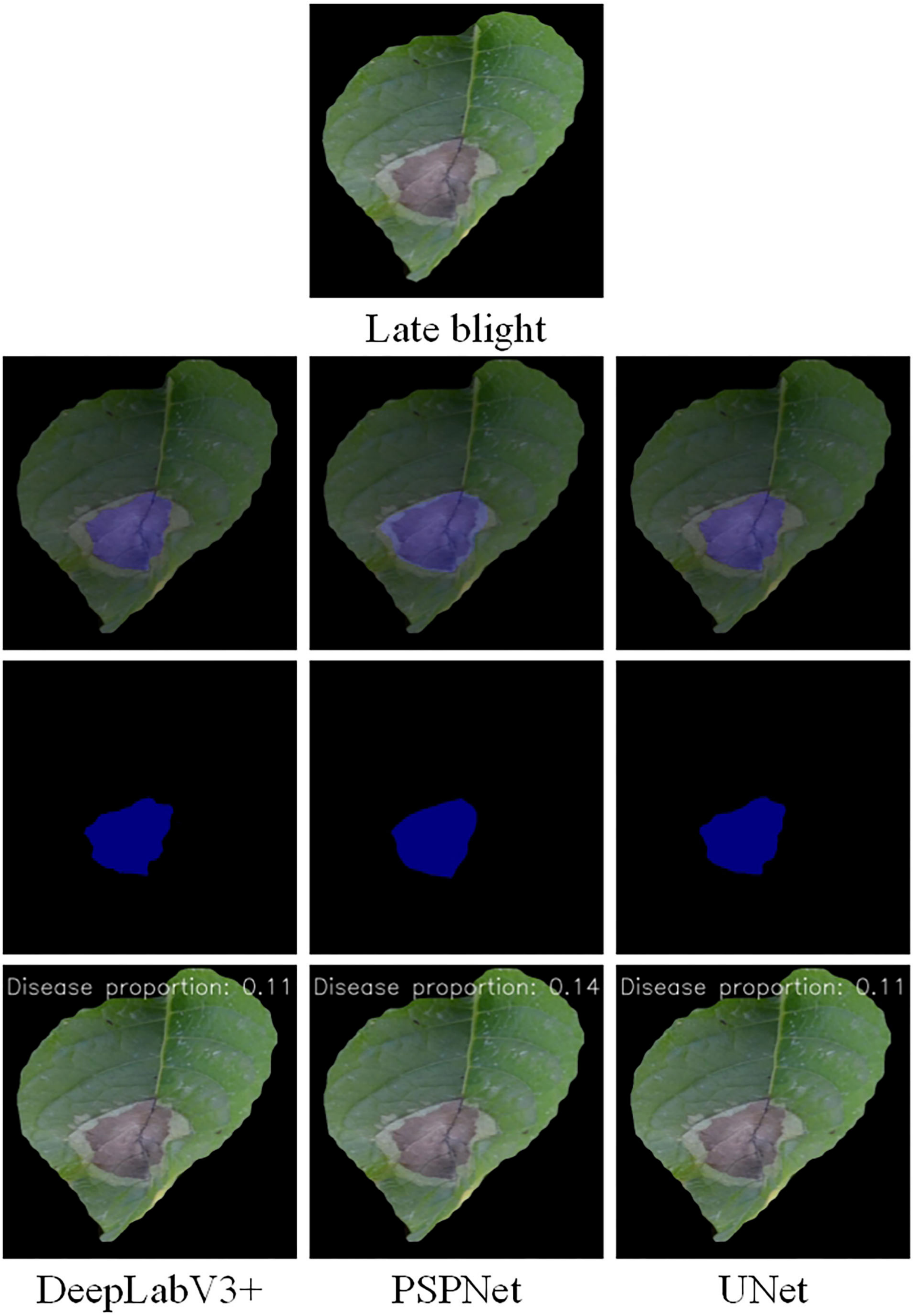


FIGURE 8 | Semantic segmentation results of late blight under the three models.

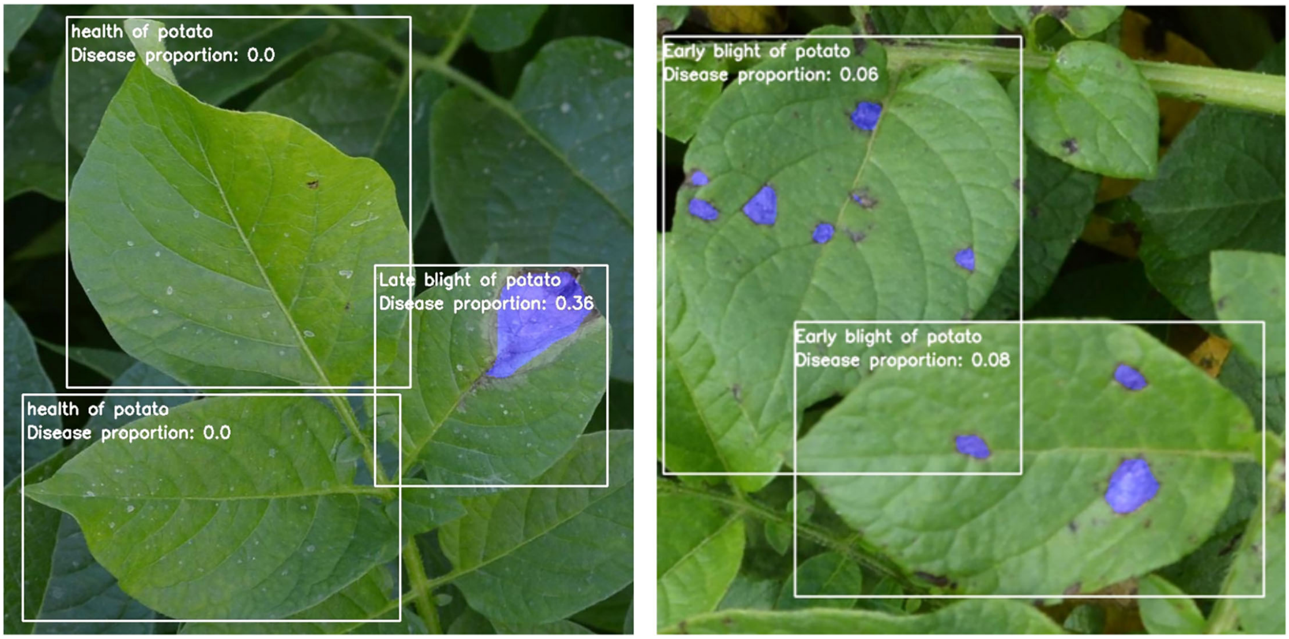


FIGURE 9 | The results of detection and recognition of potato leaves under the three-stage model.

provides the feasibility for deployment on resource-constrained mobile devices.

The accuracy of the three models had a large gap in the initial stage (see **Figure 5**). UNet achieved higher accuracy at the beginning of the training, and gradually stabilized after 10 epochs. Apart from that, DeepLabV3+ and PSPNet had a low accuracy at the beginning of the training, but DeepLabV3+ reached a relatively high accuracy after 10 epochs, and tended to be stable. Moreover, the first 40 epochs of the PSPNet model were set as the frozen epoch, so that its accuracy began to rise sharply in the 50th epoch. At the same time, PSPNet began to rise after the 40th epoch and gradually stabilized in the 80th epoch, which was closer to UNet at last. As shown in **Figure 6**, the loss of all models gradually decreased and tended to be stable with the increase of training epochs. Among them, the UNet network model converged faster than other networks and showed lower loss. Besides, the UNet network tended to be stable after 10 epochs. The DeepLabV3+ model gradually stabilized after the 50th epoch, while the PSPNet model had a sharp decline. Apart from that, the loss of PSPNet was stabilized at the 65th epoch, which was very close to DeepLabV3 + after 80 epochs.

The disease segmentation results are displayed in **Figures 7**, **8**. In the segmentation of late blight, the three models were relatively accurate and there was not much difference between them. Notably, the proportion of disease areas identified by PSPNet model was the largest. Among them, the edges of the disease area predicted by PSPNet were smoother. These rounded edges can be a factor for the slightly worse performance of PSPNet when compared with the UNet, as some pixels can end up being wrong. The edges predicted by UNet and DeepLabV3+ were more consistent with the actual disease. In the segmentation of the early blight, the disease areas segmented by UNet and PSPNet models were closest to the real situation. Meanwhile, the disease areas predicted by DeepLabV3+ were incomplete. As shown in **Figure 8**, the disease in the red box was not marked, so that the predicted disease proportion was far from the other two models.

Model Test Results

Figure 9 shows the final performance of the three-stage model on potato disease recognition. Initially, an instance segmentation stage processes the input image *via* Mask R-CNN. The instance segmentation stage splits the cropped leaves as the input of the second stage classification model. The classification model classifies leaves into healthy, early blight, and late blight, and takes two diseased leaves as input for the third-stage semantic segmentation. The potato images with complex backgrounds were input into the combined model for detection. In the prediction box, the categories of leaf diseases and the proportion of disease spots were displayed in the upper left corner. In addition, the disease areas were marked on the leaf by calling the model in the semantic segmentation stage.

DISCUSSION

In summary, the work of this study mainly consists of three parts, namely, leaf segmentation, disease area segmentation, and classification of disease category. Among them, leaf segmentation and disease area segmentation were completed by instance segmentation and semantic segmentation models, respectively. In the first stage, images with complex backgrounds were input into the Mask R-CNN networks, and the leaves without background could be obtained. In the second stage, the leaves without backgrounds were input into the classification networks to distinguish healthy or diseased leaves. In addition, to verify the applicability of the model in real-world scene detection, we further trained the model using the public Plant Village dataset. Finally, the results of this dataset are similar to the data collected in this study, which proves that the classification model used in this study can effectively identify the types of leaves under different disease stages and different degrees of infection. In the third stage, diseased areas based on the labels corresponded to the categories classified in the second stage. In the previous literature, a single model was often used to detect diseases. The experiments in this study have completed the segmentation, classification, and disease spots segmentation of leaves under natural conditions. And this study fuses the three-stage models to realize the detection of the three models on one image. In the final image detection, this study fuses the three models into an input end and an output end, reducing the complex process required for previous detection.

The combination of multi-stage CNN models has been widely applied in various research fields. For instance, Wang et al. (2021) segmented cucumber foliage diseases using a two-stage semantic segmentation model, and the results were better than the single model segmentation. Beyond that, Tassis et al. (2021) identified coffee foliage diseases using a three-stage model, and the AP and MIoU reached 71.90 and 94.25%, respectively. As indicated by the results, compared with the single model, the multi-stage model had a greater improvement in the accuracy of leaf disease detection. Although the three-stage model framework proposed in this study has achieved good results in potato disease detection, there are still some aspects that need to be improved. (1) First, potato early blight disease spots are characterized by small and dense disease area. In this model framework, some disease areas with small area and unclear color differentiation could be identified inaccurately. In the future research, the segmentation accuracy of the little lesions should be improved. (2) Second, in practical potato production, the speed of detection should be increased, and the network structure needs to be improved, so as to shorten the time of model segmentation and better serve the production. (3) In the actual working environment, due to factors, such as large planting area, the efficiency of disease spot detection is high. In this study, the use of mobile phones or cameras to take pictures to collect data will affect the efficiency of actual detection. In the future, we will try to adopt a light-weight CNN structure to reduce the model calculation time, and carry the camera and model program on the drone to achieve rapid detection of the planting area.

CONCLUSION

In the first stage, the Mask R-CNN model used two backbone networks, ResNet50 and ResNet101, respectively. The final APs obtained were 78.21 and 81.87%, respectively, and the Precisions were 96.73 and 97.13%, respectively, which achieved accurate segmentation of potato leaves in complex backgrounds.

In the second stage, the classification models were used. Apart from that, the three main networks of VGG16, ResNet50, and InceptionV3 were adopted for experiments. The potato leaves were divided into healthy leaves, early blight leaves, and late blight leaves. Besides, the accuracy of the three networks was 95.33, 94.67, and 93.33%, respectively.

In the third stage, semantic segmentation models PSPNet, UNet, and DeepLabV3+ were used for training of disease region identification. Furthermore, the identification and detection of the early blight and late blight areas were accomplished. The MIoUs were 86.08, 89.91, and 85.29%, respectively, whereas the MPAs were 93.19, 94.24, and 88.08%, respectively, indicating that the segmentation and recognition of potato disease areas were achieved.

In short, this model framework could effectively reduce the impact on potato leaf segmentation in the wild environment, improve the accuracy of disease spot segmentation, and provide technical support for potato leaf disease detection and prevention. The framework presented consisting of three models of CNN can be applied to other crops with some adjustments. In the future, the camera and the program of this study can be mounted on the UAV to realize the application in real scenes.

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/s.

AUTHOR CONTRIBUTIONS

XL: writing of the original draft. YZ: guiding and supervision. LW: data collection. JZ: proofreading and polishing of the manuscript. JL and XF: editing, supervision, and proofreading. All authors contributed to the article and approved the submitted version.

FUNDING

This study was supported by the National Natural Science Foundation of China (32072572), Hebei Talent Support Foundation (E2019100006), the Key Research and Development Program of Hebei Province (20327403D), and the Talent Recruiting Program of Hebei Agricultural University (YJ201847). Detection of crop diseases and insect pests is based on artificial intelligence and multispectral imaging (2021 Shijiazhuang City, the introduction of foreign technology projects).

REFERENCES

Adhikari, S., Shrestha, B., Baiju, B. and Saban, K. (2018). "Tomato plant diseases detection system using image processing," in *1st KEC Conference on Engineering and Technology*.

Brahimi, M., Boukhalfa, K., and Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Appl. Artif. Intell.* 31, 299–315. doi: 10.1080/08839514.2017.1315516

Chen, L. C., Papandreou, G., and Kokkinos, I. (2014). Semantic image segmentation with deep convolutional nets and fully connected CRFs. *Comput. Sci*. 4, 357–361. doi: 10.1080/17476938708814211

Cheng, X., Zhang, Y., Chen, Y., Wu, Y., and Yue, Y. (2017). Pest identification via deep residual learning in complex background. *Comput. Electron. Agric*. 141, 351–356. doi: 10.1016/j.compag.2017.08.005

Da Silva Silveira Duarte, H., Zambolim, L., Machado, F. J., Pereira Porto, H. R., and Rodrigues, F. A. (2019). Comparative epidemiology of late blight and early blight of potato under different environmental conditions and fungicide application programs. *Semin. Agrar*. 40, 1805–1818. doi: 10.5433/1679-0359.2019v40n5p1805

Fan, Z., and Li, X. (2019). Recognition of potato diseases based on fast detection and fusion features of ROI. *Southwest China J. Agric. Sci.* 544–550. doi: 10.16213/j.cnki.scjas.2019.3.015

Fang, Y., and Ramasamy, R. P. (2015). Current and prospective methods for plant disease detection. *Biosensors* 5, 537–561. doi: 10.3390/bios5030537

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

Fuentes, A., Yoon, S., and Park, D. S. (2019). Deep learning-based phenotyping system with glocal description of plant anomalies and symptoms. *Front. Plant Sci*. 10:1321. doi: 10.3389/fpls.2019.01321

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis*. 2980–2988. doi: 10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*. 770–778. doi: 10.1109/CVPR.2016.90

Khan, M. A., Akram, T., Sharif, M., Javed, K., Raza, M., and Saba, T. (2020). An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection. *Multimed. Tools Appl*. 79, 18627–18656. doi: 10.1007/s11042-020-08726-8

Krishnaswamy Rangarajan, A., and Purushothaman, R. (2020). Disease classification in eggplant using pre-trained VGG16 and MSVM. *Sci. Rep*. 10, 1–11. doi: 10.1038/s41598-020-59108-x

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. doi: 10.1109/CVPR.2015.7298965

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 640–651.

Madufor, N. J. K., Perold, W. J., and Opara, U. L. (2018). detection of plant diseases using biosensors: a review. *Acta Hortic*. 1201, 83–90. doi: 10.17660/ActaHortic.2018.1201.12

Qu, D., Xie, K., Jin, L., Pang, W., Bian, C., and Duan, S. (2005). Development of China's potato industry and food safety. *Sci. Agric. Sin.* 358–362. doi: 10.3321/j.issn:0578-1752.2005.02.022

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

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. *arXiv e-prints, arXiv:*1505, 04597. doi: 10.1007/978-3-319-24574-4_28

Tassis, L. M., Tozzi de Souza, J. E., and Krohling, R. A. (2021). A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images. *Comput. Electron. Agric*. 186, 106191. doi: 10.1016/j.compag.2021.106191

Tsedaley, B. (2014). Late blight of potato (*Phytophthora infestans*) biology, economic importance and its management approaches. *Journal of Biology Agriculture & Healthcare* 4, 215–226.

Wang, C., Du, P., Wu, H., Li, J., Zhao, C., and Zhu, H. (2021). A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-net. *Comput. Electron. Agric*. 189, 106373. doi: 10.1016/j.compag.2021.106373

Xiao, Z., and Liu, H. (2017). Adaptive features fusion and fast recognition of potato typical disease images. *Trans. Chin. Soc. Agric. Machin.* 26–32. doi: 10.6041/j.issn.1000-1298.2017.12.003

Xie, C., Shao, Y., Li, X., and He, Y. (2015). Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging. *Sci. Rep*. 5, 1–11. doi: 10.1038/srep16564

Yang, S., Feng, Q., Zhang, J., Sun, W., and Wang, G. (2020). Identification method for potato disease based on deep learning and composite dictionary. *Trans. Chin. Soc. Agric. Machin.* 22–29. doi: 10.6041/j.issn.1000-1298.2020.07.003

Yellareddygari, S. K. R., Taylor, R. J., Pasche, J. S., Zhang, A., and Gudmestad, N. C. (2018). Prediction of potato tuber yield loss due to early blight severity in the midwestern United States. *Eur. J. Plant Pathol*. 152, 71–79. doi: 10.1007/s10658-018-1449-0

Zhang, K., Wu, Q., Liu, A., and Meng, X. (2018). Can deep learning identify tomato leaf disease? *Adv. Multimed*. 2018, 6710865. doi: 10.1155/2018/6710865

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2016). Pyramid scene parsing network. *IEEE Comput. Soc.* doi: 10.1109/CVPR.2017.660

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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 06 July 2022
doi: 10.3389/fpls.2022.864045

ASP-Det: Toward Appearance-Similar Light-Trap Agricultural Pest Detection and Recognition

Fenmei Wang [1,2,3], Liu Liu [4], Shifeng Dong [1,2], Suqin Wu [3], Ziliang Huang [1,2], Haiying Hu [1] and Jianming Du [1]**

[1] Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Science, Hefei, China, [2] University of Science and Technology of China, Hefei, China, [3] Computer Teaching and Research Office of the Department of Information Engineering PLA Army Academy of Artillery and Air Defense, Hefei, China, [4] Shanghai JiaoTong University, Shanghai, China

Automatic pest detection and recognition using computer vision techniques are a hot topic in modern intelligent agriculture but suffer from a serious challenge: difficulty distinguishing the targets of similar pests in 2D images. The appearance-similarity problem could be summarized into two aspects: texture similarity and scale similarity. In this paper, we re-consider the pest similarity problem and state a new task for the specific agricultural pest detection, namely **A**ppearance **S**imilarity **P**est **D**etection (ASPD) task. Specifically, we propose two novel metrics to define the texture-similarity and scale-similarity problems quantitatively, namely Multi-Texton Histogram (MTH) and Object Relative Size (ORS). Following the new definition of ASPD, we build a task-specific dataset named PestNet-AS that is collected and re-annotated from PestNet dataset and also present a corresponding method ASP-Det. In detail, our ASP-Det is designed to solve the texture-similarity by proposing a Pairwise Self-Attention (PSA) mechanism and Non-Local Modules to construct a domain adaptive balanced feature module that could provide high-quality feature descriptors for accurate pest classification. We also present a Skip-Calibrated Convolution (SCC) module that can balance the scale variation among the pest objects and re-calibrate the feature maps into the sizing equivalent of pests. Finally, ASP-Det integrates the PSA-Non Local and SCC modules into a one-stage anchor-free detection framework with a center-ness localization mechanism. Experiments on PestNet-AS show that our ASP-Det could serve as a strong baseline for the ASPD task.

Keywords: appearance-similarity pest detection, pairwise self-attention, skip-calibrated convolution, object relative size, anchor-free

OPEN ACCESS

Edited by:
Lei Shu,
Nanjing Agricultural University, China

Reviewed by:
Marcin Wozniak,
Silesian University of Technology, Poland
Peng Chen,
Anhui University, China

****Correspondence:***
Liu Liu
liuliu1993@sjtu.edu.cn
Jianming Du
djming@iim.ac.cn

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *28 January 2022*
Accepted: *31 May 2022*
Published: *06 July 2022*

Citation:
Wang F, Liu L, Dong S, Wu S, Huang Z, Hu H and Du J (2022) ASP-Det: Toward Appearance-Similar Light-Trap Agricultural Pest Detection and Recognition. Front. Plant Sci. 13:864045. doi: 10.3389/fpls.2022.864045

1. INTRODUCTION

Diversity pest control and prevention are always a crucial agricultural issue worldwide (Sivakoff et al., 2012). To build a cost-effective and efficient pest controlling system, most of the current methods deal with pest monitoring as a pest detection task (Shen et al., 2018). Specifically, the applications employing computer vision techniques attempt to exploit vision features extracted from pre-defined Convolutional Neural Network (CNN) and analyze the visual information to recognize or detect a targeted pest (Deng et al., 2018) and plant leaf disease (Dhaka et al., 2021). Generally, these applications are deployed into a mobile camera or other flexible vision sensors (Liu et al., 2017).

However, in the practical agricultural environment, the in-field pest detection systems require high-quality image resolution and strict image collection standards, e.g., the distance between the camera and pest targets cannot be larger than 1 m (Wang et al., 2021). Besides, these approaches might confront troubles in recognizing lots of pest categories at the same time (Ayan et al., 2020). These limit the functional performance when employing these computer vision algorithms in real-world pest monitoring (Wang et al., 2020). Under this case, several works attempted to install fixed stationary cameras in light traps to monitor pest occurrence by recognizing and detecting the trapped pests (Liu et al., 2019a). But there are two challenges when identifying these captured pests: (1) a large number of pest categories usually share similar textures in images that prevent fine-grained classification. (2) the size of one pest is very close to each other, making it difficult to distinguish them. These challenges are considered appearance-similarity problems in computer vision and pest detection tasks.

In this paper, we pay attention to dealing with the challenges of pest recognition and detection in light traps, which use frequency-vibrating insecticidal lamps to capture pests and use a fixed camera to take pictures of pests that fall into the trapping tray, and stating a new task for the specific agricultural pest detection problem, namely **A**ppearance **S**imilarity **P**est **D**etection (ASPD) task. This task clearly defines and summarizes the appearance-similarity problems from two aspects: texture-similarity and scale-similarity. To further describe these two problems, we define the corresponding metrics: (1) Multi-Texton Histogram(MTH), a statistical index representing the distribution of pests' textures. (2) Object Relative Size (ORS), measuring the pest sizes in captured RGB images. From MTH and ORS, we formulate the ASPD to be a novel pest detection task.

To validate the difficulty of the ASPD task, we build a task-specific dataset, namely PestNet-AS. This dataset is collected and re-annotated from the famous pest detection benchmark PestNet (Liu et al., 2019b). In PestNet-AS, we present a hierarchical category taxonomy. The sup-classes in PestNet-AS are Lepidoptera and Coleoptera, the former contains 17 sub-class categories and the latter contains 7. In total, the PestNet-AS dataset covers 87,672 images and 554,761 pest annotations. Our dataset is aligned with the ASPD task.

Accompanying with ASPD task and PestNet-AS dataset, we propose a deep learning framework ASP-Det to evaluate the performance of the ASPD task. Specifically, our ASP-Det is designed to solve the texture-similarity by submitting a Pairwise Self-Attention (PSA) mechanism and Non-Local Modules to construct a domain adaptive balanced feature module that could provide high-quality feature descriptors. On the other hand, we also present a Skip-Calibrated Convolution (SCC) module that can balance the scale variation among the pest objects and re-calibrate the feature maps into the sizing equivalent of pests. Finally, we constructed a one-stage feature detector for the ASPD task, using a deep convolutional layer of free-anchor. We also introduce a center-ness calibration center strategy for the construction to compensate for the potential localization inaccuracy caused by the absence of the RPN. Finally, this model considers meeting the practical application requirements in agricultural fields.

Our contributions could be summarized as follows:

- We re-consider the light-trap pest recognition and detection problem and state a new pest detection task ASPD. In this task, we quantitatively define the texture-similarity and scale-similarity problems in pest detection using MTH and ORZ metrics.
- We build a new large-scale dataset PestNet-AS specific to ASPD tasks. The dataset contains 87,672 images and 556,521 pest annotations.
- We propose a novel ASP-Det network to address the challenges of the ASPD task. We present PSA mechanism and Non-Local Modules module for dealing with the texture-similarity problem and the SCC module for Scale-Similarity. We believe our ASP-Det could serve as a strong baseline for ASPD tasks and further promote agricultural pest monitoring applications.

2. RELATED WORK

2.1. Anchor-Free Object Detection

Convolutional neural network-based Object detectors can be divided into two types, namely anchor-based and anchor-free, based on whether anchors are preset. The former can be divided into one-stage and two-stage detection models, and the latter can be divided into key-point-based and center-based detection models. Anchor-free based on keypoint detection algorithms include CornerNet (Law and Deng, 2020), Grid R-CNN (Lu et al., 2020), ExtremeNet (Zhou et al., 2019), and CenterNet (Duan et al., 2019). Anchor-free based on the center point algorithm is a type of detection method that defines the target center point or central area as a positive sample and then regresses the distance from the four sides of the bounding box. YOLO series (Redmon et al., 2016; Bochkovskiy et al., 2020), DenseBox, RetinaNet (Lin et al., 2017b), FCOS (Tian et al., 2019), and FoveaBox (Kong et al., 2020) all belong to this category. Generally, these methods occupy less computing resources and are faster than anchor-based methods. They are suitable for high-speed real-time object detection tasks in applications.

2.2. Pest Detection

At present, scholars have studied more general object detection methods. However, these methods cannot be directly utilized in the pest detection tasks, which we confront are relatively particular. Different from pest recognition methods, pest detection methods based on the deep learning methods used deep convolutional networks (Dai et al., 2016) to automatically identify the category and location of the target according to the model algorithm. Liu et al. (2019b) put forward an approach for large-scale multi-class pest detection, which can detect 16 classes of agricultural pests using an End-to-End deep convolutional neural network. Jiao et al. (2020) proposed a two-stage anchor-free convolutional neural network to realize small-scale pests detection for the multi-categories agricultural pest. Yao and Xu (2020) proposed an automatic detection model for pest damage symptoms on rice canopy based on improved RetinaNet. The average accuracy of the detection of the two pests in the pest-like area reached 93.76%. Dan et al. (2021) showed a method of automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification. Tetila, EC. used five deep learning architectures with a fine-tuning for the category of soybean pest images, which reached an accuracy of up to 93.8% (Tetila et al., 2020). Wang. et al. integrated context-aware information representation in-field. A multi-projection pest detection model (MDM) was proposed and trained by crop-related pest images in Wang et al. (2020). Automatic in-trap pest detection by end-to-end on a GPU workstation with data augmentation and then deployed on embedded devices with minimal prepossessing in Sun et al. (2018).

2.3. Similar Object Detection

Similar object detection considers detection methods with more detailed features. The general approaches adopt fine-grained strategies to address the challenges. The current research on fine-grained detection mainly includes the following content: Feng (2013) proposed a set of training images, which can identify a sparse number of image patches in the training set which cover most parts of the target object in the test image. Li et al. (2016) used fine-grained detection for face-screen distance on smartphones. However, there are only a few applications of fine-grained detection related to agriculture and almost few for similar pest detection. Thus, this paper conducts a detailed study on the feature extraction of similar pests, builds a model, and provides an algorithm framework with better accuracy and real-time performance.

3. PROBLEM STATEMENT

We present the Appearance-Similarity Pest Detection(ASPD) task in our work. Specifically, we define ASPD task from two aspects: texture-similarity that describes the gray-level and color-level appearance of these pest targets (Section 3.1), and scale-similarity that describes size-level appearance of pests (Section 3.2). For each problem, we propose the corresponding metrics to define these settings.

3.1. Texture-Similarity

To quantitatively define texture-similarity, we consider it from the following: (1) gray-level similarity that defines whether the objects are similar in gray images. (2) color-level similarity that defines whether the colorized pests are similar.

For gray-level similarity, a Hash algorithm is a common method to describe image similarity. In detail, the perceptual Hash (pHash) algorithm usually achieves better performance than deference Hash (dHash) as well as average Hash (aHash). Thus, we propose to use the pHash to analyze and define the gray-level similarity problem. In this metric, we randomly select 100 images from one category of pest, calculate 32 × 32 Discrete Cosine Transform (DCT), and select 8 × 8 matrix in the upper left corner. Next, we apply pHash algorithm to extract the pest target representation value, as the object gray-level representation. Finally, we define the object similarity such that the representation value is larger than 0.6.

On the other hand, we consider color-level pest similarity. In this problem, we first use MTH to describe the repetition law and repetition mode of the image pixel-level information, expressed in texture information in different color spaces. In terms of texture information, the multi-element histogram method uses the Sobel operator to detect the edge of the image and detect the texture direction and then describes the texture and shape information of the image. The Sobel operator calculates the three color channels separately in the RGB color space. The two vectors corresponding to the horizontal and vertical directions are returned in each channel. $a(R_x, G_x, B_x)$ and $b(R_y, G_y, B_y)$ represent the gradient information in the corresponding direction of the corresponding channel. Further, we can obtain the texture by calculating formulas 1–4.

$$\mid a \mid= \sqrt{(R_x)^2 + (G_x)^2 + (B_x)^2} \tag{1}$$

$$\mid b \mid= \sqrt{(R_y)^2 + (G_y)^2 + (B_y)^2} \tag{2}$$

$$a \cdot b = R_x \cdot R_y + G_x \cdot G_y + B_x \cdot B_y \tag{3}$$

$$\theta = \arccos \left[\frac{a \cdot b}{\mid a \mid \cdot \mid b \mid} \right] \tag{4}$$

In terms of color information, the results obtained from the three channels of R, G, and B are quantified into 64 color images with four different primitives in $C(x, y)$. Perform texture detection in the process to obtain the texture primitive image $T(x, y)$. Finally, according to $T(x, y)$, a multi-element histogram describes texture features. The definition of the MTH is shown in formulas 5 and 6:

$$H(T(P_1)) = N \left\{ \theta(P_1) = v_1 \bigwedge \theta(P_2) = v_2 \|P_1 - P_2\| = D \right\} \tag{5}$$

$$H(T(P_1)) = \overline{N} \left\{ \theta(P_1) = w_1 \bigwedge \theta(P_2) = w_2 \|P_1 - P_2\| = D \right\} \tag{6}$$

where $P1 = (x_1, y_1)$, $P2 = (x_2, y_2)$ represent two adjacent pixels with a distance of D in the original image. Their corresponding pixels in the primitive image $T(x, y)$ are $T(P1) = w_1$ and $T(P2) = w_2$, respectively. In the texture direction matrix $\theta(x, y)$, the directions of the points P1 and P2 are $\theta(P1) = v_1$, $\theta(P_2) = v_2$. N represents the number of times v_1 and v_2 appear together, and $\overline{N}$ represents the number of times w_1 and w_2 appear together. $H[T(P1)]$ represents the number of times that the same edge direction appears at the same time under a certain color background; it represents the number of times the same color appears under a certain edge direction. Therefore, the texture feature vector f_v of the image is expressed as shown in formula 7:

$$\mathrm{f}(v) = H(T(P_1)) \circ H(\theta(P_1)) \tag{7}$$

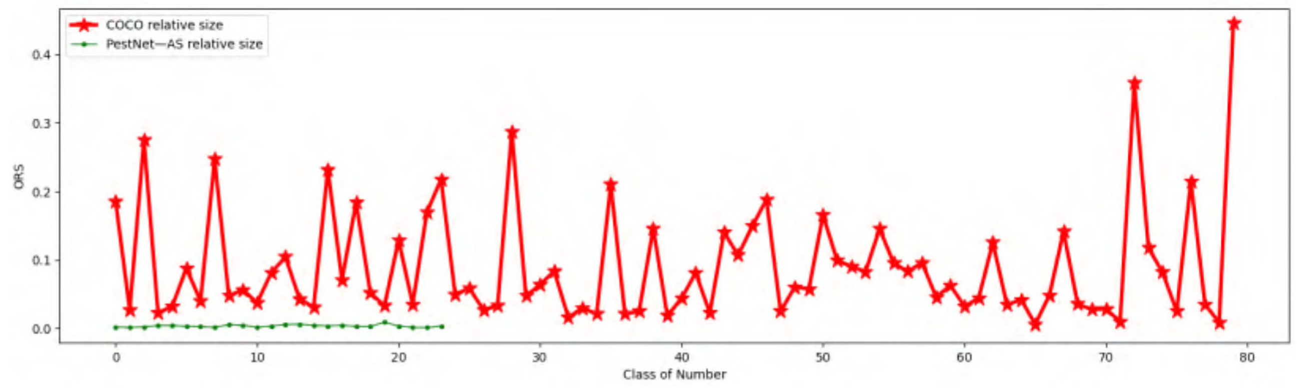


FIGURE 1 | Comparison of Object Relative Size (ORS) with common object datasets MS COCO and PestNet-AS.

where ∘ means connection.

The similarity of images I_1 and I_2 is defined as shown in Equation (8):

$$S_I(I_1, I_2) = \|f_v(I_1) - f_v(I_2)\|^{-1} \tag{8}$$

where$\|f_v\|$ denotes Euclidean distance.

3.2. Scale-Similarity

We adopt ORS to measure the problem for scale-similarity. Specifically, given an RGB image with a shape of $H \times W$ and the i-th pest bounding box $H_i \times W_i$, the ORS$_i$ of this pest object is defined as follows:

$$\mathrm{ORS}_i = \frac{H_i \cdot W_i}{H \cdot W} \tag{9}$$

In this way, we can count the ORS for the c-th category in the entire dataset by

$$\mathrm{ORS}^{(c)} = \frac{\sum_{i=1}^{M} ORS_i \cdot \mathrm{sgn}(c_i, c)}{\sum_{i=1}^{M} \mathrm{sgn}(c_i, c)} \tag{10}$$

where M is the number of pest objects and function sgn(·) indicates whether the category of i-th pest is c-th class, that belongs to defined as

$$\mathrm{sgn}(c_i, c) = \begin{cases} 1 & c_i = c \\ 0 & c_i \neq c \end{cases} \tag{11}$$

Finally, we can obtain the ORS distribution map of all the categories of pest species. **Figure 1** illustrates the Relative Size distribution of our targeted 24 pest categories. All the ORS of all pest objects are not larger than 1%, which indicates that all the pests in our work are small in size. Furthermore, most of the categories hold nearly 0.5% ORS, which is in line with the difficulty of scale-similarity in the ASPD task.

4. DATASET

To solve the ASPD task, we present a large-scale dataset named PestNet-AS, which is built from a popular dataset PestNet (Section 4.1). To meet the ASPD problem setting, we analyze our PestNet-AS dataset from texture-similarity and scale-similarity (Section 4.2).

4.1. Data Collection

To the best of our knowledge, there is no dataset suitable for the similarity pest detection task, so we extract a sub-dataset with a similar appearance from PestNet, filter, and re-annotate it. We select part of the categories of PestNet to validate our PestNet-AS task and method. Specifically, we build a simple category taxonomy, as shown in **Figure 2**. The taxonomy contains 2 sup-classes and 24 sub-classes(categories).

This paper resizes these pest images to 1,333 × 800 from 2,560 × 1,920 and 2,592× 1,944. We chose 87,672 pictures and divided into two sup-classes and 24 sub-classes. **Table 1** shows two categories of pests' scientific names, their average relative size to the whole pest images. The two significant pest portraits are shown in **Figure 1**.

Data annotation was done by professionals using Labeling software under the guidance of entomologists[1]. The pest location coordinates and classes are saved as an XML file, then converted to JSON format, which has the same format as COCO. The number of annotations corresponds to the number of bounding boxes labeled in each image. Every image could contain more than one annotation depending on the number and classes of pests. To evaluate the effectiveness and practicability of the model, we randomly selected images from the dataset according to the proportion of 80% (70,138 images) of the training set and 20% (17,534 images) of the test set.

[1] The PestNet is a set of light trap datasets jointly annotated by professionals and agricultural experts from Jiaduo Company, which provides data support for intelligence agriculture. Artificial Intelligence Agriculture Valley has developed a special labeling software for agricultural pests and diseases. This dataset is also selected and organized in this dataset driven by similar pest detection problems.

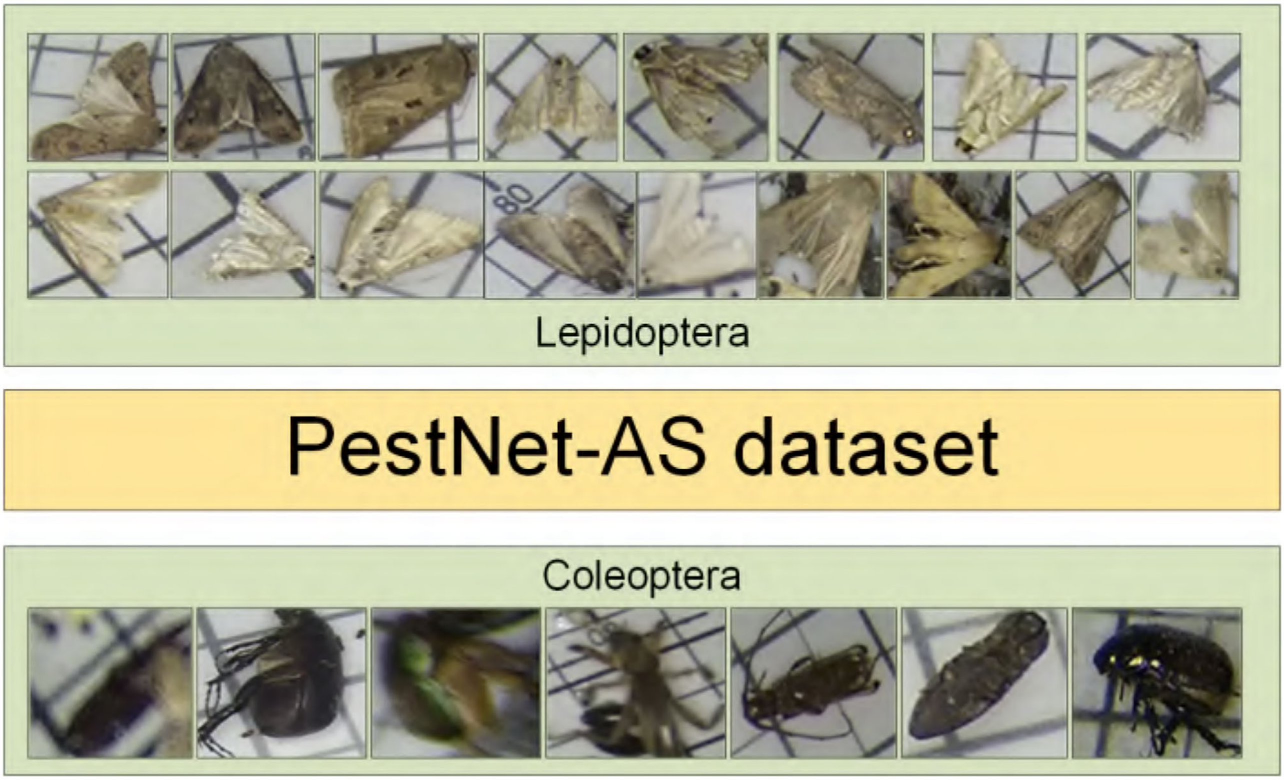


FIGURE 2 | Visualization of two sup-class of pests: the figure shows the visualization of similar pests in the 17 sub-classes of the Lepidoptera and 7 sub-classes of Coleoptera.

4.2. Dataset Analysis

The PestNet-AS dataset is established to solve the ASPD task, thus it is built to meet the definitions of texture-similarity and scale-similarity problems. We use the designed metric to validate the dataset characteristics on texture-similarity. Concerning gray-level similarity, we apply the pHash algorithm described above to evaluate the 24 sub-classes in the two sup-classes. The results are shown in **Tables 2**, **3**. Almost all pest similarities are more extensive than 0.6, which aligns with the gray-level pest similarity problem definition, which indicates that the pest objects in our PestNet-AS are highly similar in texture.

In terms of color-level similarity, we adopt the MTH algorithm to evaluate PestNet-AS dataset. Specifically, we crop all the pest targets in our dataset and calculate their MTH features. **Figure 3** shows the t-SNE map on these features. These pests from various categories lie in very close feature spaces and have identical characteristics. Therefore, our PestNet-AS meets the requirement of texture similarity.

For the scale-similarity problem, we calculate ORS for each pest object, and the results are shown in **Figure 1**. Due to the specific attribute of each object class, the ORS of labeled instances are unevenly distributed among these categories for MS COCO (Lin et al., 2014). Compared with MS COCO, the ORS for our dataset PestNet-AS holds a similar scale for almost all the types, which indicates that our PestNet-AS also meets the scale-similarity problem. Therefore, we can conclude that PestNet-AS could be used as a benchmark for ASPD tasks.

5. ASP-DET, A DEEP LEARNING FRAMEWORK FOR ASPD

5.1. Motivation

In this paper, we aim to solve the problem of pests with similar-appearance and size equivalent, which is one of the major challenges in the fine-grained detection task. Specifically, the Pest classification problem is worse than detection. We pay more attention to developing practical pest monitoring systems for appearance-similar pest datasets in light-trap (PestNet-AS). As shown in **Figure 4**, PestNet-AS contains many challenging issues for pest detection approaches, such as pest targets with dense occlusion, high similarity, including texture similarity and scale similarity. In addition, the relative size of our similar dataset is also smaller than that of the COCO dataset, as shown in **Figure 1**. Given these thorny problems, we must consider both the detection accuracy and real-time characteristics. Therefore, we propose to use a one-stage pyramid feature extraction model to detect ASPD tasks. The SCC module and the non-local module are added to the model to solve the problem of scale similarity and texture similarity.

TABLE 1 | Description of pests of the two sup-classes.

Pest ID	Sup-class	Sub-class	No. of images	No. of instances	ORS (%)
1	Lepidoptera	Spodoptera frugiperda	226	241	0.189
2		Rice leaf roller	7,430	12,994	0.124
3		Chilo suppressalis	3,323	8,462	0.206
4		Xestia c-nigrum	1,691	2,224	0.397
5		Mythimna separata	12,502	25,526	0.403
6		Helicoverpa armigera	25,364	74,769	0.293
7		Ostrinia furnacalis	19,536	43,316	0.238
8		Proxenus lepigone	24,041	122,509	0.144
9		Agrotis exclamationis	1,082	1,782	0.530
10		Spodoptera litura	8,083	10,936	0.448
11		Spodoptera exigua	14,615	28,133	0.151
12		Stem borer	5,719	8,475	0.306
13		Agrotis ipsilon	9,944	15,397	0.567
14		Land cutworms	1,131	1,805	0.601
15		Cabbage moth	7,108	10,410	0.434
16		Scotogramma trifolii Rottemberg	13,114	23,301	0.346
17		Yellow cutworms	3,825	4,933	0.434
18	Coleoptera	Holotrichia parallela	24,041	122,509	0.286
19		Anomala corpulenta	1,082	1,782	0.240
20		Gryllotalpa orientalis	8,083	10,936	0.904
21		Pleonomus canaliculatus	14,615	28,133	0.323
22		Agriotes fuscicollis miwa	5,719	8,475	0.130
23		Melanotus caudex	9944	15,397	0.101
24		Holotrichia oblita	1,131	1,805	0.320

TABLE 2 | Description of the 17 sub-classes of phash 32 × 32 similarity of pests.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	70.01	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2	69.74	68.16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3	68.06	68.46	68.53	–	–	–	–	–	–	–	–	–	–	–	–	–	–
4	68.47	65.77	69.26	72.18	–	–	–	–	–	–	–	–	–	–	–	–	–
5	70.84	69.61	71.57	72.36	73.61	–	–	–	–	–	–	–	–	–	–	–	–
6	70.57	69.66	71.47	70.41	71.85	75.34	–	–	–	–	–	–	–	–	–	–	–
7	72.76	71.35	73.17	70.01	72.97	72.97	75.34–	–	–	–	–	–	–	–	–	–	–
8	70.62	68.69	70.45	72.26	71.87	72.94	75.34	77.74	–	–	–	–	–	–	–	–	–
9	69.19	66.68	67.45	70.54	67.98	71.60	72.33	73.96	66.61	–	–	–	–	–	–	–	–
10	70.44	67.85	69.91	69.90	69.69	71.58	74.17	75.16	65.17	70.30	–	–	–	–	–	–	–
11	70.43	67.57	69.61	71.20	71.19	72.26	74.82	77.40	66.53	70.75	65.88	–	–	–	–	–	–
12	71.33	69.33	71.57	69.66	72.34	71.46	76.66	76.11	68.26	69.79	67.92	72.16	–	–	–	–	–
13	69.02	66.89	68.93	71.34	70.07	72.97	72.50	75.81	66.40	70.56	65.83	72.11	72.56	–	–	–	–
14	70.82	68.73	69.35	73.30	70.36	74.34	74.43	76.80	68.26	71.61	67.10	72.16	72.81	72.07 –	–	–	–
15	69.54	67.71	69.50	72.28	70.42	73.32	73.22	76.80	67.35	70.36	67.07	72.68	73.28	71.19	74.12–	–	–
16	71.61	68.92	71.07	72.53	71.96	74.54	76.19	78.33	67.09	71.84	67.80	74.33	73.56	73.19	76.04	74.41	–
17	70.03	68.66	71.03	70.97	71.97	71.25	76.75	76.13	66.21	70.72	67.52	71.68	69.94	70.37	73.86	71.77	68.95

5.1.1. Pest Recognition on Texture-Similarity Problem

In the process of pests in the ASPD task, it is not easy to accurately classify because the appearance and texture are too similar. The main reason is that the feature expression is not strong enough. The current method only considers the low-level feature maps in the feature pyramid as their local features.

TABLE 3 | Description of the 7 sub-classes of phash 32 × 32 similarity of pests.

	18	19	20	21	22	23	24
18	62.23	–	–	–	–	–	–
19	62.45	60.85	–	–	–	–	–
20	63.04	62.14	66.07	–	–	–	–
21	63.10	63.17	67.25	68.18		–	–
22	62.45	63.89	66.96	70.05	68.68	–	–
23	62.42	63.64	68.64	68.42	66.92	71.41	–
24	61.75	62.23	66.93	67.21	68.02	72.45	70.63

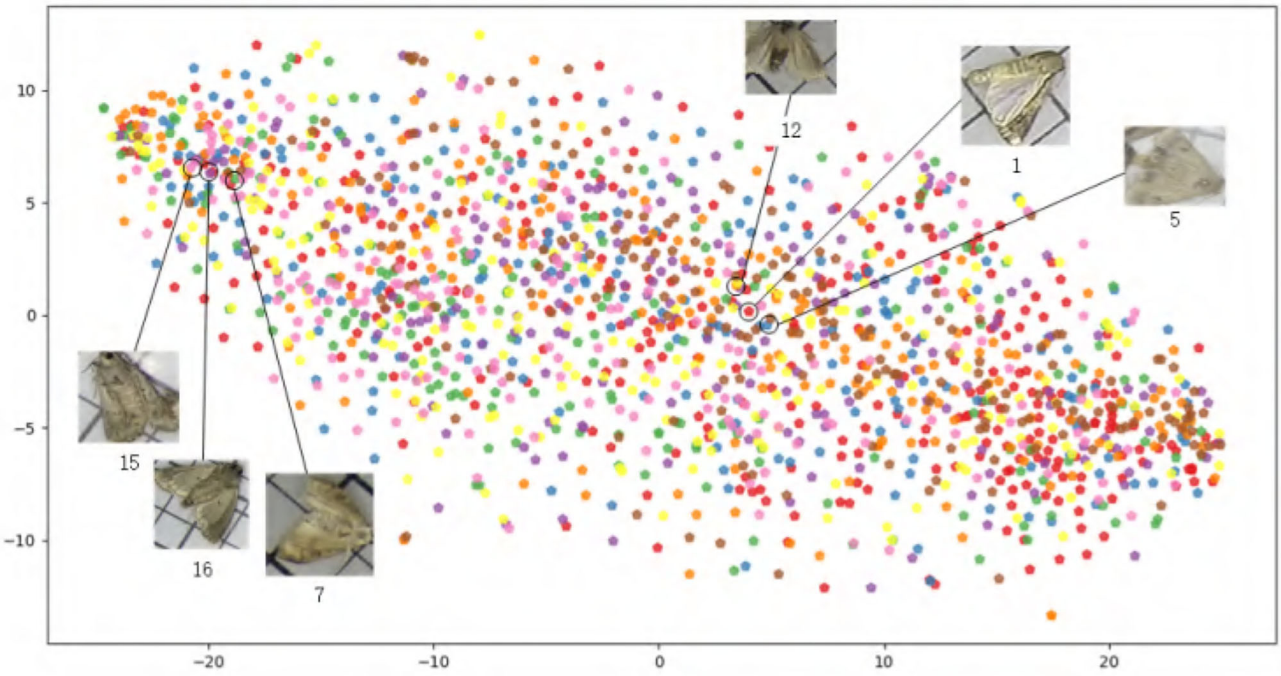


FIGURE 3 | PestNet-AS similarity description in Multi-Texton.

It ignores the high-level semantic information so that the pest targets have sound positioning effects, but classification accuracy is not good. On the other hand, simultaneously considering the simple superposition of low-level and high-level feature map information will cause confusion on local characteristics of pests. Lack of pertinence for pests with high similarity will affect the recognition effect and cause the detection method to be inaccurate. The classification results are shown in **Table 4**.

5.1.2. Pest Detection on Scale-Similarity Problem

The pest scales are too close, and a large number of redundant anchors are not used, which seriously affects the positioning of the frame, so the detection is not very accurate. First, we investigate the network performance in the standard feature pyramid network algorithm. The primary purpose is to express various dimensional characteristics for objects of different sizes effectively. However, the relative scale of our dataset changes little, and the appearance features are incredibly similar. So, the recall rate is not satisfactory at all stages of the IOU. Especially when the IOU becomes more prominent, the recall rate decays more severely. The results are shown in the following **Table 5**. Considering the characteristics of the PestNet-AS dataset, we expect to use the feature extraction of the feature pyramid network in the model training. To avoid the poor effect caused by small size changes, we need to reconstruct the feature pyramid.

5.2. ASP-Det Overview

This section describes the proposed scale-calibrated free anchor CNN detection method for appearance-similar agricultural pests. The proposed pest detection model ASP-Det consists of pest features extraction network multi-classes pest detection network. We construct a non-local feature pyramid network (NFP). We construct ASP-Det with PSA module,which can fuse the features with different levels.Then joint skip-calibrated convolution module (SCC) in the features pyramid network for detecting similar pest object. Overview of ASP-Det framework shown in

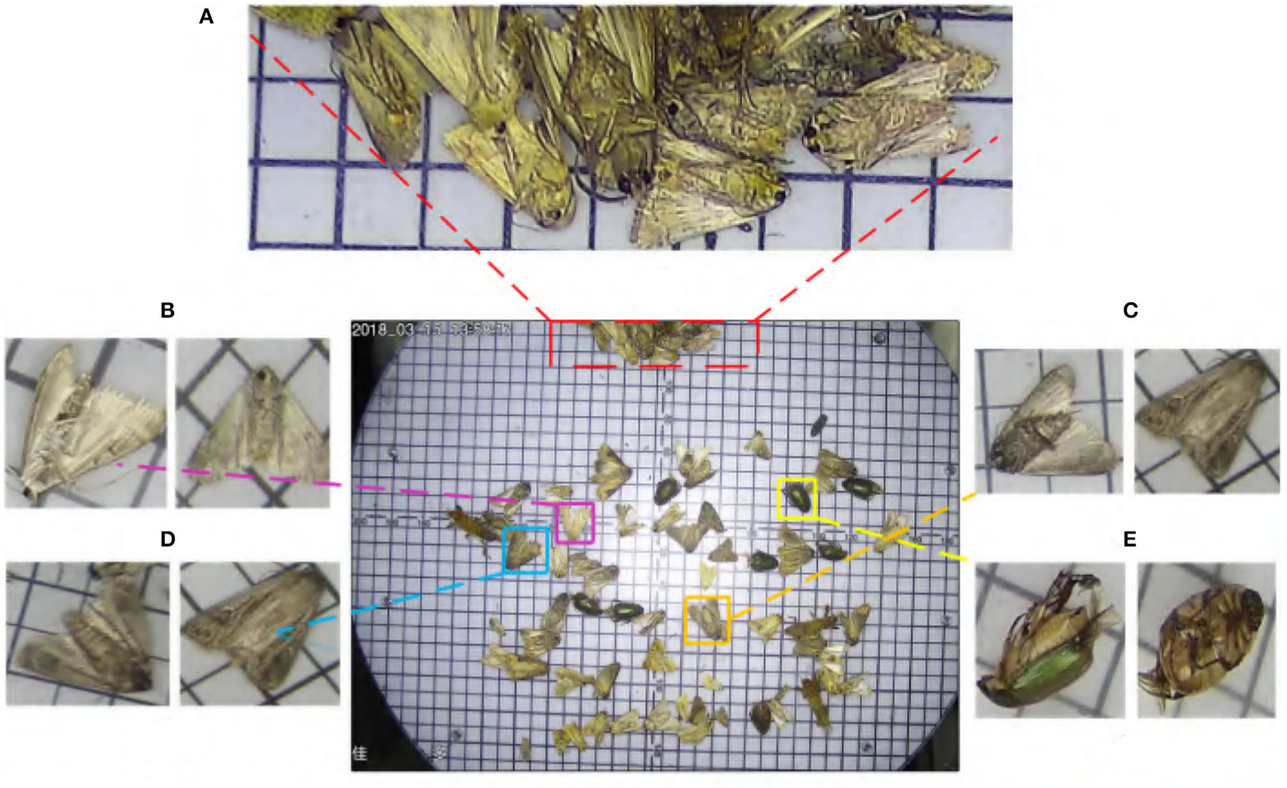


FIGURE 4 | Some typical challenges in appearance-similar pest detection **(A)** appearance-similar pest density distributed; **(B,C)** pests with high similarity on the ventral and dorsal sides; **(D,E)** different postures of appearance-similar pests of the Lepidoptera and Coleoptera.

TABLE 4 | Classification results of appearance-similar pests using different methods.

Methods	Top-1 (%)	Top-5 (%)
ResNet-50	50.2	71.1
SENet	58.6	75.6
VGG-16	48.6	73.7
Inception	42.3	62.8

Figure 5. Specifically, we first fed a picture entering the CNN feature extraction network, and we added the PSA channel module during the feature extraction process. Second, a non-local operation is performed on the obtained feature map and then input into the feature pyramid network. Finally, we design an SCC strategy that takes an interval in the feature pyramid to form a feature sampling layer, ensuring the integration of sample features across levels. Third, we introduce center-ness to suppress the low-quality detected bounding boxes produced by the locations far from the center of an object. Finally, non-maximum suppression (NMS) algorithm is employed to remove redundant boxes for the same object (Symeonidis et al., 2019).

5.3. PSA Module

Because the dataset has large similarity in appearance and morphology and the number of samples of various classes is not balanced. This paper designs a new feature pyramid that joins the non-local and SCC Modules to resolve the above problems. Different from former approaches (Lin et al., 2017a; Yu et al., 2021) that integrate multi-level features using lateral connections, our key idea is to strengthen the multi-level features using the same deeply integrated balanced semantic features. Each layer simultaneously realizes two functions in CNN, feature aggregation and feature transformation. The former incorporates the characteristics of all positions extracted by the kernels, and the latter performs conversion through linear mapping and nonlinear scalar functions. Thus, the integration function is suitable for phase detection networks, and the transformation function is ideal for feature pyramid networks. Suppose the feature transformation is set as an element-level operation composed of linear mapping and nonlinear scalar functions. In this paper, we introduce the Pairwise module (Zhao et al., 2020) to establish feature aggregation. Consistent with global activated PSA modules, the final result is expressed as a weighted sum of adaptive weights and features:

$$y_i = \sum_{j \in R(i)} \alpha(x_i, x_j) \odot \beta(x_j) \tag{12}$$

Where x_i and x_j are feature maps with indexes i and j, $\odot$ is the Hadamard product called aggregation with the local footprint R(i), several parameters in the PSA module will not be affected

TABLE 5 | Recall performance: FCOS on PestNet-AS with ResNet-50-FPN as a backbone.

IoU	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
Recall_1	0.356	0.337	0.317	0.278	0.254	0.231	0.171	0.112	0.051	0.001
Recall_{10}	0.472	0.431	0.415	0.356	0.314	0.251	0.192	0.163	0.082	0.003
Recall_{100}	0.614	0.585	0.462	0.382	0.366	0.341	0.275	0.195	0.123	0.007
MRecall	0.588	0.513	0.426	0.365	0.344	0.313	0.254	0.182	0.091	0.005

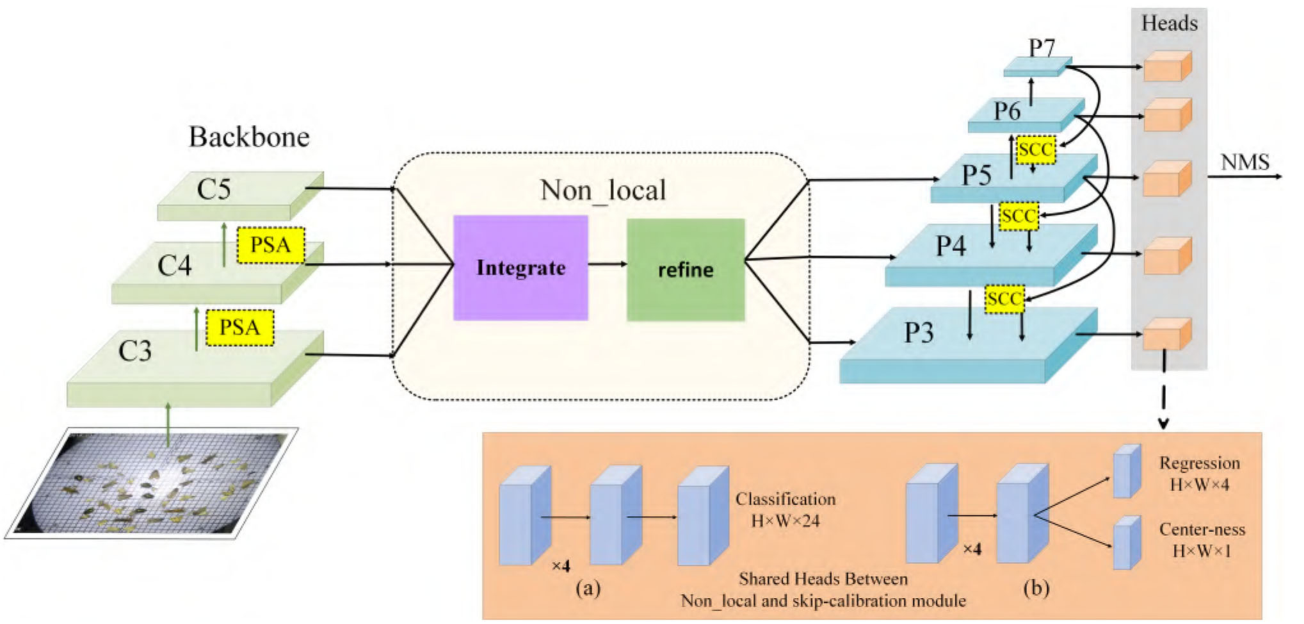


FIGURE 5 | Overview of ASP-Det framework. (a) Classification branch, (b) regression and center branch. PSA, pairwise self-attention module; SCC, skip-calibrated convolution module.

by the size of the footprint. After this aggregation, the result y_i can be obtained.

The vector$\beta(x_j)$ generated by the function $\beta(\cdot)$ will be aggregated with the adaptive vector $\alpha(x_i, x_j)$ introduced later. Compared with ordinary weights, adaptive vector $\alpha(x_i, x_j)$ has strong content adaptability. It can be decomposed as follows:

$$\alpha(x_i, x_j) = \gamma(\delta(x_i, x_j)) \tag{13}$$

where $\delta(\cdot)$ and $\gamma(\cdot)$, respectively, represent a relation function and a hybrid map composed of linear and nonlinear functions. Based on the relation $\delta(\cdot)$, the function $\gamma(\cdot)$ is used to obtain a vector result, which can be combined with $\beta(x_j)$ in Equation (10). In general, matching the output dimension of $\gamma(\cdot)$ with the dimension of $\beta(x_j)$ is unnecessary because attention weights can be shared among a group of channels. We choose the subtraction as the relation function, which can be formulated:

$$\delta(x_i, x_j) = \varphi(x_i) - \phi(x_j) \tag{14}$$

where $\varphi(\cdot)$ and $\phi(\cdot)$ are convolution operations matching output dimensions. $\delta(\cdot)$ calculates spatial attention for each channel instead of sharing between channels. We adopt a non-local refine the feature as a pyramid network after aggregation.

Non-local mean (Wang et al., 2017) is a classical filtering algorithm that computes a weighted mean of all pixels in an image. It allows distant pixels to contribute to the filtered response at a location based on patch appearance similarity. The non-local behavior in Equation (15) is because all positions [$\forall(j)$] are considered in operation. A convolutional process sums up the weighted input in a local neighborhood as a comparison. A non-local process is a flexible building block that can be used with convolutional layers. It can be added into the earlier part of deep neural networks, unlike *fc* layers that are often used in the end, which allows us to build a hierarchical model that combines non-local and local information.

$$y_i = \frac{1}{C(x)} \sum_{\forall(j)} f(X_i, X_j) g(X_j) \tag{15}$$

The above PSA module uses novel vector attention, which can generate content adaptation ability while maintaining the channel adaptation ability. PSA module makes our appearance-similar target detection model have strong adaptability, which can effectively enhance the salient differences between different features. The pipeline is shown in **Figure 6**. It consists of two

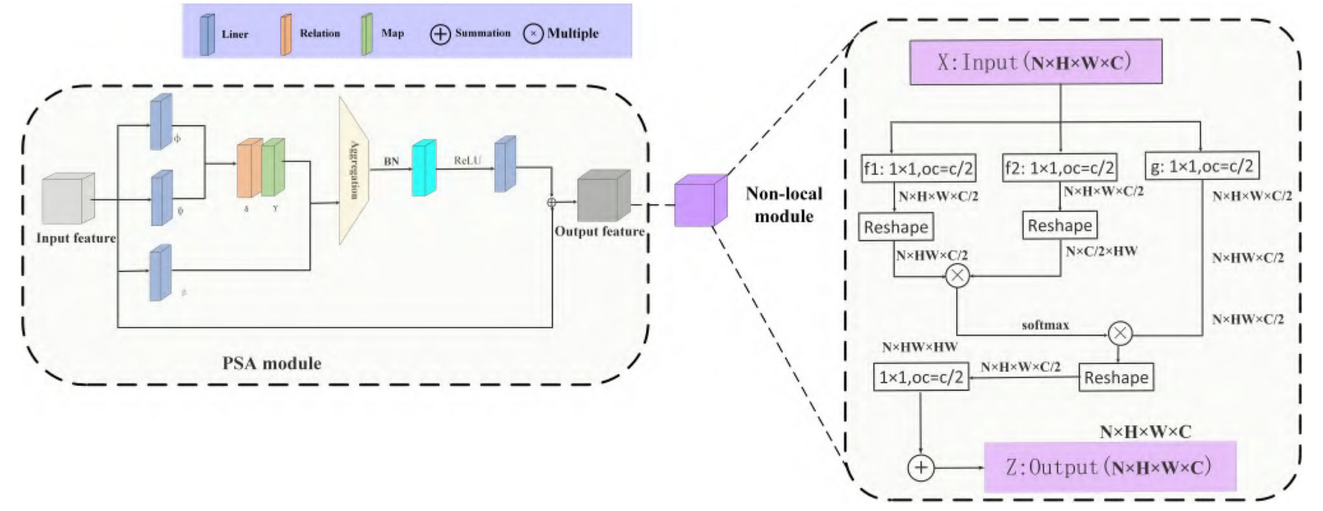


FIGURE 6 | PSA module and non-local module.

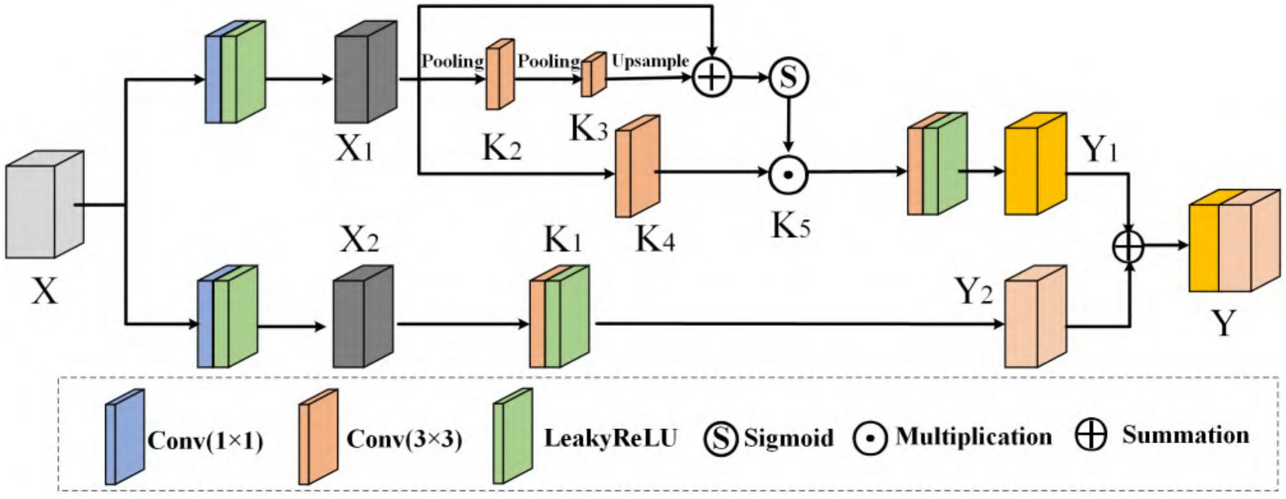


FIGURE 7 | SCC module.

branches and four steps: re-scaling, integrating, refining, and strengthening.

Also, we observe that the similar pests in the images are primarily small and size equivalent. Using state-of-the-art object detection approaches to these images will make similar pest features prone to lose after high-level convolution. It is challenging to extract similar pest features in the network. Hence, the novel Skip-Calibrated Convolution model can combine the delicate features in a high-level convolutional layer. The integral structure of pest come from a low-level convolutional layer. Then, we could fuse the contextual information around pests from the low-level convolutional layer and address the issue of features misjudged for the similar object in the deep convolution layer. In the next section, we will present the alternative optimization for similar pest detection from the internal structure of a CNN and give details of the ASP-Det.

5.4. SCC Module

The structure of deep CNNs is becoming more and more complicated, which can enhance the network's learning ability. The novel module called SCC considers improving the feature transformation process in convolution since pests with high similarity may be difficult to judge in adjacent layers. We do not only use the features of the upper layer to perform up-sampling directly but also introduce the information of the following high-level into the sampling so that features have better recognition, adding a specific architecture in **Figure 7**.

A given group of filter sets K with the shape (C, C, kxh, kxw) is divided into two branches, which are responsible for conducting

[K1, K2, K3, K4, K5] different functions, respectively. In SCC, we perform feature transform at two scales: the original scale and the smaller scale after down-sampling. For a given X, we adopt max pooling to reduce the scale:

$$M_1 = MaxPool_r(X1) \tag{16}$$

$$T_1 = MaxPool_r(M_1) \tag{17}$$

where r is the down-sampling rate and stride of the pooling process. The receptive field at each spatial location can be effectively expanded by benefiting from the down-sampling operation. Next, T_1 can be used as an input to the filter K2 and K3 following the up-sample procedure, which restores the feature to the original scale, resulting in

$$X_1' = Up(F_2(T_1)) = Up(T_1 \times K_2) \tag{18}$$

$$X_1'' = Up(F_3(X_1')) = Up(X_1' \times K_3) \tag{19}$$

where $F_2(T_1) = T_1 \times K_2, F_3(X_1' = X_1' \times K_3)$ is a simplified form of convolution. Then, the calibrated operation can be formulated as

$$Y_1' = F_4(X_1) \odot Sigmod(X_1'') \tag{20}$$

Where $F_4(X_1) = X_1 \times K_4$, $Sigmoid(\cdot)$ is an activation function. The final result of the skip-calibrated part is calculated:

$$Y_1 = F_5(Y_1') \tag{21}$$

Where $F_5(Y_1') = Y_1' \times K_2$. The other part can be obtained from another branch that does not require scale transformation. The formula is as follows:

$$Y_2 = F(X_2) \times K_1 \tag{22}$$

Finally, we sum Y_1 and Y_2 to get the final result Y. Reviewing the entire SCC enables each spatial position to adaptively encode the context from a long-range region, which is also a vast difference between it and the traditional FPN network.

5.5. Optimization

ASP-Det is a fully convolutional one-stage object detector. Unlike anchor-based sensors, which consider the location on the input image as the center of anchor boxes and regress the target bounding box for these anchor boxes, we directly revert the target bounding box for each location. Let $F_i \in R^{H \times W \times C}$ be the feature maps at layer i of a backbone CNN. For each location(x, y) on the feature map F_i, we can map it back onto the input image as$(\frac{S}{2} + xs, \frac{S}{2} + ys)$, which is near the center of the receptive field of the location(x, y). Besides the label for classification, we also have a 4D ground truth vector $q = (l, r, t, b)$ being the regression target for each sample. Here l, r, t, and b are the distances from the location to the four sides of the bounding box. If a location falls into multiple bounding boxes, it is considered an ambiguous sample.

In addition, we observed that it is due to many low-quality predicted bounding boxes produced by locations far away from the center of an object. We propose a simple yet effective strategy to suppress these low-quality detected bounding boxes without introducing any hyper-parameters. Specifically, we add a single layer branch in parallel with the regression branch to predict

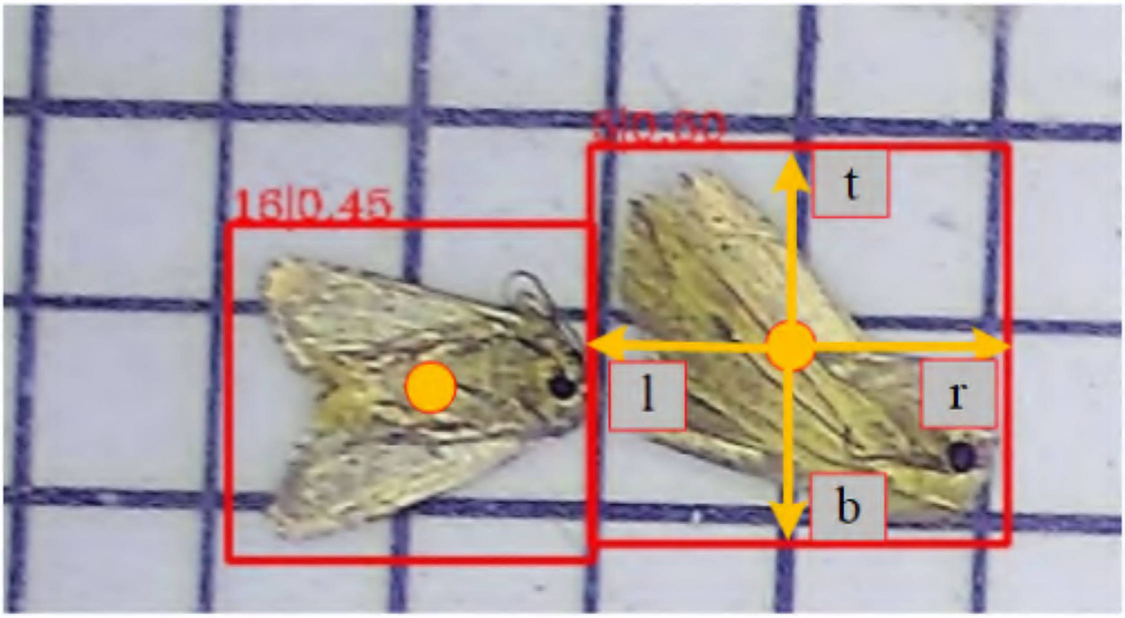


FIGURE 8 | ASP-Det works by predicting a 4D vector (l,t,r,b) encoding the location of a bounding box at each foreground pixel.

the center-ness of a location, as shown in **Figure 8**. Given the regression targets l, t, r, and b for a site, the center-ness target is defined as,

$$center - ness = \sqrt{\frac{min(l,r)}{max(l,r)} \times \frac{min(t,b)}{max(t,b)}} \tag{23}$$

We define our training loss function as follows:

$$\begin{aligned} L(p_{x,y}, q_{x,y}, O_{x,y}) \; = \; & \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) \\ & + \lambda_1 \frac{1}{N_{pos}} \sum_{x,y} \mathbf{sign}(c^*_{x,y} > 0) L_{reg}(q_{x,y}, q^*_{x,y}) \\ & + \lambda_2 L_{centerness}(O_{x,y}, O^*_{x,y}) \end{aligned} \tag{24}$$

where L_{cls} is the focal loss as in Lin et al. (2017c), L_{reg} is the IOU loss as in UnitBox (Yu et al., 2016), and $L_{centerness}$ is the center-ness loss ranges from 0 to 1 and is thus trained with binary cross entropy (BCE) loss. N_{pos} denotes the number of positive samples and the summation is calculated over all locations on the feature maps F_i. The indicator function being 1 if $c^*_{x,y} > 0$ otherwise is 0. The balanced parameter λ_1 and λ_2 are set to 1. We employ sqrt here to slow down the decay of the center-ness. When testing, the final score $S_{x,y}$ (used for ranking the detections in NMS) is the square root of the product of the predicted center-ness $O_{x,y}$ and the corresponding classification score $P_{x,y}$. After the above center-ness suppression, we can obtain better pest detection performance.

$$S_{x,y} = \sqrt{P_{x,y} \times \; O_{x,y}} \tag{25}$$

6. EXPERIMENTS

6.1. Experiment Settings

6.1.1. Evaluation Metrics

In this paper, we apply five metrics to evaluate the performance of our similar pest detection method: AP50 (Precision in 0.5), AP75 (Precision in 0.75), mAP (mean Average Precision), Recall and MR (mean Recall), and BPR (Best Possible Recall).

TABLE 6 | The MR and BPR for Ablation study for different strategies of assigning objects to FPN levels.

Methods	PSA	NFP	CL	SCC	MR	BPR
Faster R-CNN					57.0	87.2
YOLOv3					50.2	88.9
FCOS					58.8	88.7
ATSS					61.4	93.6
Swin-t					61.8	93.7
ASP-Det (ours)	√				62.2	91.9
ASP-Det (ours)	√	√			62.3	93.5
ASP-Det (ours)	√	√	√		62.4	94.3
ASP-Det (ours)	√	√	√	√	62.3	94.5

TABLE 7 | Overall performance comparison.

Method	PSA	NFP	CL	SCC	*AP*	AP_{50}	AP_{75}
General object detection							
Faster R-CNN (Ren et al., 2015)					41.9	70.7	46.2
YOLOv3 (Redmon and Farhadi, 2018)					30.8	63.2	25.1
FCOS (Tian et al., 2019)					44.0	73.0	49.0
ATSS (Zhang et al., 2020)					44.2	73.0	49.0
Swin-t (Liu Z. et al., 2021)					43.6	74.1	47.2
Pest sdetection							
AF-RCNN (Jiao et al., 2020)					31.6	50.3	32.6
PestNet (Zhang et al., 2020)					42.1	70.9	36.3
Ours							
ASP-Det	√				44.1	73.2	49.2
ASP-Det	√	√			44.3	73.6	49.4
ASP-Det	√	√	√		44.6	74.3	49.9
ASP-Det	√	√	√	√	**45.0**	**74.9**	**50.2**

Boldface represents emphasis.

6.1.2. Training Details

ResNet-50 is used as our backbone network, and the same hyper-parameters with FCOS are used. Specifically, our network is trained with stochastic gradient descent (SGD) for 90 k iterations with the initial learning rate is 0.0125 and a mini-batch of four images. We trained the network for 12 epochs, ran SGD for the first eight epochs, reduced the learning rate to one-tenth in the 11th epoch, and reduced the learning rate to one-tenth in the 11th epoch. We initialize our backbone networks with the weights pre-trained on ImageNet (Jia et al., 2009). For the newly added layers, we initialize them as in Lin et al. (2017c).

6.1.3. Inference Details

We first forward the input image through the network and obtain the predicted bounding boxes with the predicted class scores. The next post-processing of ASP-Det strictly follows that of FCOS. The post-processing hyper-parameters are also the same, except we use NMS threshold of 0.5 instead of 0.6 in FCOS. Moreover, we use the exact sizes of input images as in training.

6.2. Pest Detection Performance of ASP-Det

The section shows that the concern is not particularly important by comparing the MR of ASP-Det and that of its anchor-based counterpart on the dataset. The following analyses are based on the ASP-Det implementation in mmdetection2.

6.2.1. Mean Recall (MR) Performance

Formally, MR is defined as the ratio of the number of ground-truth boxes that a detector can recall at the average to the number of all ground-truth boxes. A ground-truth box is recognized if the box is assigned to at least one training sample (i.e., a location in ASP-Det or other detectors), and a training sampling can be associated with at least one ground-truth box. As shown in **Table 6**, both with a NFP, a SCC, and Center-ness Loss (CL) on reg obtain similar $MR(58.8 vs. 62.3\%)$, 12.1 points higher than YOLOv3, 5.3 points higher than Faster R-CNN, and 3.5% higher than FCOS. Moreover, because the best recall of current detectors is much lower than 90%, the small Best Possible Recall gap (<1%) between ASP-Det(NFP), ASP-Det(NFP+SCC), and ASP-Det will not affect the performance of a detector. Therefore, the concern about the low Best Possible Recall may not be necessary for our method.

6.2.2. Average Precision (AP) Performance

To test the effectiveness of our ASP-Det, we compare the quality pest bounding box by ASP-Det and other state-of-the-art detectors. We choose faster R-CNN, FCOS, and YOLOv3 to compare our proposed ASP-Det on a similar pest dataset. The pest detection results are shown in **Tables 7**, **8**. We can observe that our method outperforms faster R-CNN and YOLOv3. The mAP of our method can achieve 45%, 14.2 higher than YOLOv3, and 3.1 higher than Faster R-CNN. For extreme special pests

TABLE 8 | AP50 and all classes of pests for different detection methods on the similar pest dataset.

Pest ID	YOLOv3	Faster R-CNN	FCOS	ATSS	Swin	ASP-Det (ours)
1	55.6	64.7	71.2	73.2	73.6	**73.9**
2	56.0	65.2	68.5	70.9	70.8	**70.9**
3	67.9	72.0	75.3	75.6	76.4	**76.6**
4	64.1	72.3	69.0	72.5	72.6	**73.3**
5	73.0	79.1	81.4	81.4	81.5	**81.6**
6	85.8	88.3	90.1	**90.2**	89.9	90.0
7	75.7	78.7	81.0	81.4	81.5	**81.6**
8	72.6	76.2	78.7	78.4	**78.8**	**78.8**
9	59.0	77.6	77.7	**82.1**	81.5	81.6
10	65.4	72.6	75.2	76.8	76.9	**77.0**
11	52.6	57.4	60.0	61.6	61.2	**62.3**
12	74.3	79.5	82.1	82.9	**83.4**	82.6
13	75.6	85.6	86.6	**87.5**	87.6	87.2
14	38.1	62.7	67.8	66.5	69.7	**69.8**
15	55.5	66.5	67.9	**69.8**	**69.8**	69.6
16	65.9	74.2	75.7	**76.3**	75.7	75.8
17	54.3	59.4	63.4	**65.4**	64.0	64.1
18	84.2	87.8	89.4	89.3	89.5	**89.6**
19	88.3	90.1	90.3	91.1	91.1	**91.2**
20	94.2	95.5	95.7	95.1	95.1	**95.9**
21	17.5	34.4	46.1	39.7	48.9	**49.0**
22	79.2	82.2	83.4	**85.4**	84.8	85.0
23	27.9	29.4	34.4	31.7	35.4	**36.0**
24	35.6	46.7	50.0	54.1	54.3	**54.4**
mean	63.2	70.7	73.0	74.5	**74.1**	**74.9**

Boldface represents emphasis.

(classes “21” and “23”), the detection accuracy is lower than other classes of pests. However, our method still performs better than YOLOv3 and Faster R-CNN, benefiting from our feature fusion module.

In order to be able to directly observe the advantages of our proposed pest detection method compared with other methods. We show some visualized pest detection results of our practices, YOLOv3 and Faster R-CNN, as shown in **Figure 9**. It shows that our method can achieve more accurate results and fewer missing pests than the other methods. The model also uses the detection results to graph the classification value and recall rate of IOU in the interval of 0.5 and 0.95 from the **Figure 10**; our model has good convergence and a high recall rate and accuracy rate.

6.3. Ablation Experiments

6.3.1. The Effectiveness of PSA

A PSA mechanism introduces, which prevents background noises, and refines similar pest features. The self-attention module uses novel vector attention, generating content adaptation ability while maintaining the channel adaptation ability. The self-attention module makes our similar target detection model have strong adaptability, effectively removing and enhancing the salient differences between different features. The PSA mechanism is beneficial for feature extraction of objects with appearance-similar. We introduce the PSA mechanism to obtain the weights for each channel and multiply them with the raw feature map.

6.3.2. The Effectiveness of SCC

Because some pests are highly similar in appearance and almost the same size, in the training process, we deal with the ambiguity of the same FPN level by selecting the bounding box with the smallest area. In the test, if two objects A and B with the same category overlap, no matter which objects the position in the overlap prediction is, the forecast is correct. The missing object can be predicted by the work only belonging to it. If A and B do not belong to the same category, the overlapping position may indicate the category of A but will return to the bounding box

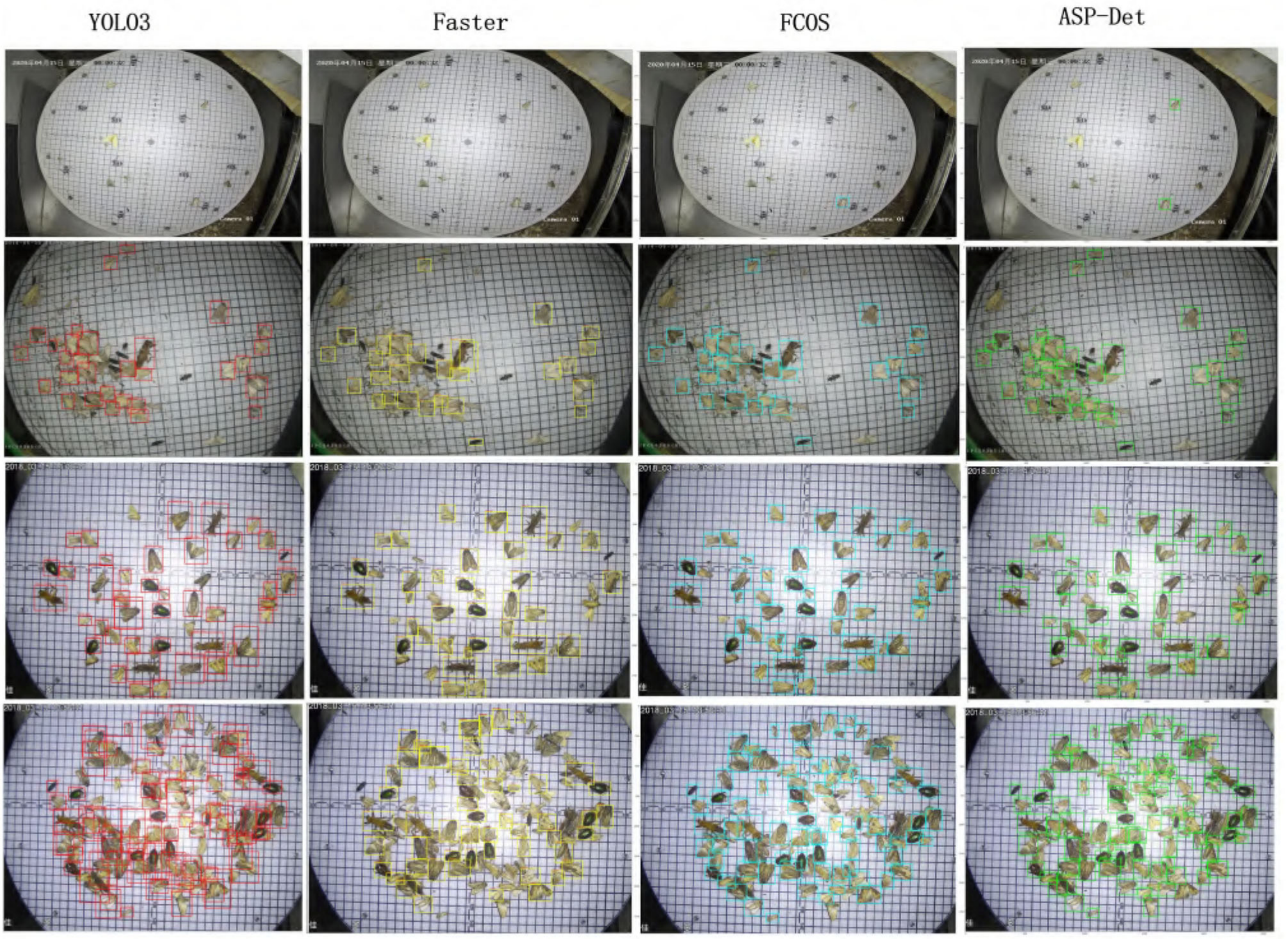


FIGURE 9 | Detection results of YOLOv3 (column 1), Faster RCNN (column 2), FCOS(column 3), and our ASP-Det (column 4).

of B, which will cause errors. The SCC module is mainly used to adjust the size jump problem in FPN. Using the SCC module can make the pests have a larger field of vision in feature areas of similar sizes, which helps distinguish the illusion of classification confusion caused by similar texture problems.

6.3.3. The Effectiveness of Center-Ness

ASP-Det using multi-level FPN prediction can only solve the target occlusion between different sizes. In the same feature-level processing, intractable ambiguity will still appear. However, the size of most of the target data in our dataset is not much different. Many of these problems that need to be considered are the occlusion problems of targets of the same scale. As mentioned before, we introduce center-ness to suppress the low-quality detected bounding boxes produced by the locations far from the center of an object. As shown in **Table 7**, the center-ness branch is used in regression and classification. The AP improvement of the dataset is not very large; AP from 44.3 to 44.6% is not obvious.

6.3.4. The Effectiveness of Different Backbones

To prove that our module plays a vital role in different backbones, we use several backbone frameworks for experiments, as shown in the **Table 9**. Our proposed method has good performance for our proposed ASPD task, so applications that expect the same task can refer to and use this algorithm framework. Using different backbones for ASPD tasks, from the results, the resnet network structure is more mature and robust, and the accuracy is higher. Without a better and faster implementation method, it is relatively safe to use the resnet network architecture at the current practical stage.

6.4. Real-Time Performance

In the field of real-time image enhancement, image super-resolution (SR) is a crucial research hotspot (Liu X. et al., 2021). In real-time applications in agriculture, real-time performance is also critical. Real-time depth models are prominent in practical applications as an agricultural image detection method. Moreover, we also designed a real-time version named ASP-Det_RT. We reduce the scale of input images from 1,333 × 800 to 800 × 512, which decreases the inference time per image by 50%. The effect is shown in **Figure 11**.

We evaluate the computation efficiency of our multi-categories similar pest detector from the aspects of training and testing time and compare it with FCOS, YOLOv3, and Faster R-CNN. The testing time of our method and FCOS method takes 0.045 s per pest image in total, which is slightly faster than Faster R-CNN and 2.5 times slower than the YOLOv3 detector. However, compared with FCOS and YOLOv3 detectors, the

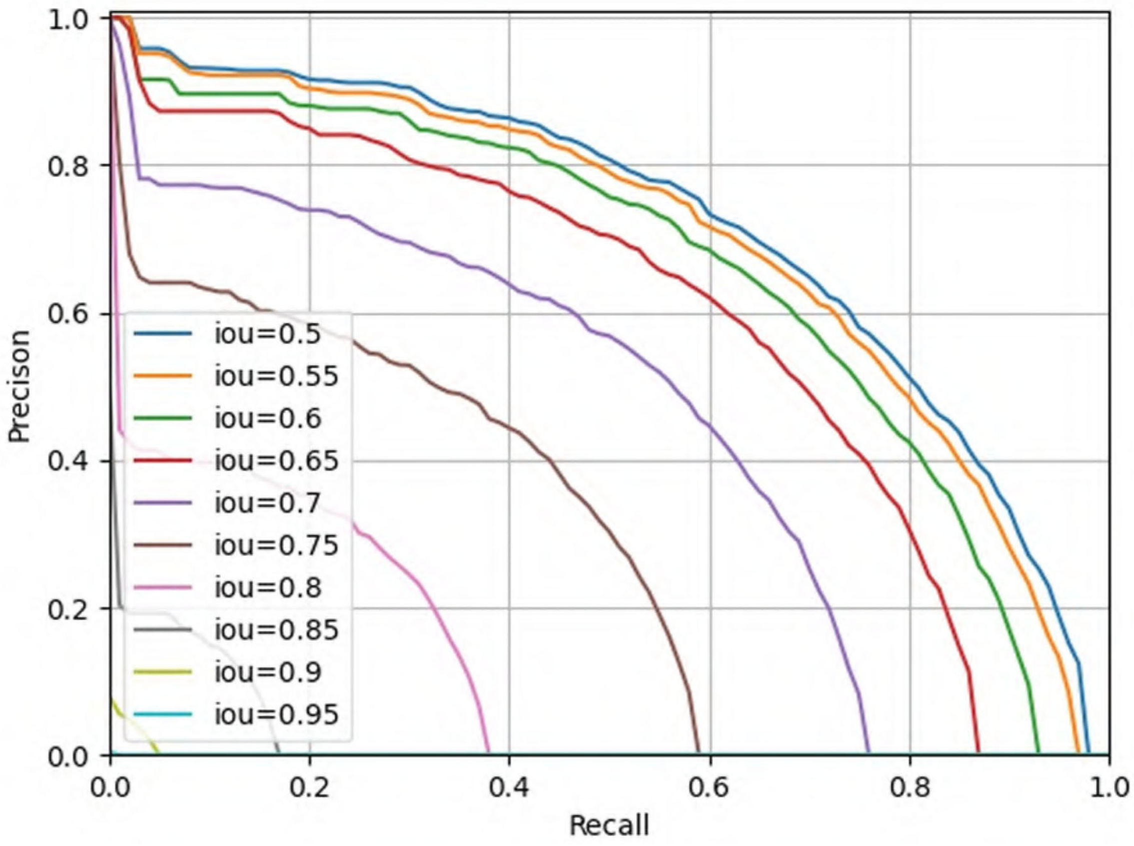


FIGURE 10 | Classification results of IOU (0.5–0.95).

training time of our pest detector is faster, and most importantly, the detection precision of our approach is primarily higher than YOLOv3. Otherwise, the hyper-parameter of our approach is less than Faster R-CNN and YOLOv3. Therefore, considering detection efficiency and accuracy, our method is the best choice and applicable to detect the 24-category similar pests.

6.5. Qualitative Results

For appearance-similar agricultural pests, even if we use the attention mechanism, non-local fusion, and skip module for processing, the target still has some misclassifications and undetectable situations. As shown in **Figure 12**, other pests located around the larger size pests inside the red box are difficult to identify and may be affected by the size and posture of the pests in the box. Another part is due to the problem of the time interval for catching pests, which causes some distortion of the color of some pests (the pink boxes) and misses inspection. The model may not recognize some pests because they are too similar to the background color or neighboring pests (like the sample in the purple box in the first image). Another part is that the size of the pests is relatively small compared to the original size in other pictures, and the posture is also more diverse, which causes the model to miss detection (such as the sample in the cyan box). Finally, there may be missed detection due to the model's limitations, which will be the main focus of follow-up research.

TABLE 9 | The ap value for Pest-as under different backbones.

Backbone	AP	AP_{50}	AP_{75}
ResNet-50	45.0	74.9	50.2
HRnet	44.6	74.4	49.9
ResNetXt	45.5	75.5	50.9
Res2Net	45.1	74.6	50.2
Swin-t transform	44.6	74.9	48.2

7. CONCLUSION

Our proposed ASP-Det does not employ IoU scores between anchor and ground-truth boxes to determine the training labels. Additionally, ASP-Det avoids all computation and hyper-parameters related to anchor boxes and solves similar pest detection in a per-pixel prediction fashion, similar to other dense prediction tasks, such as semantic segmentation. Fortunately, the accuracy of ASP-Det is also excellent for pest appearance-similarity. Given the superior performance and merits of the anchor-free detector (e.g., much more straightforward and fewer hyper-parameters), we encourage

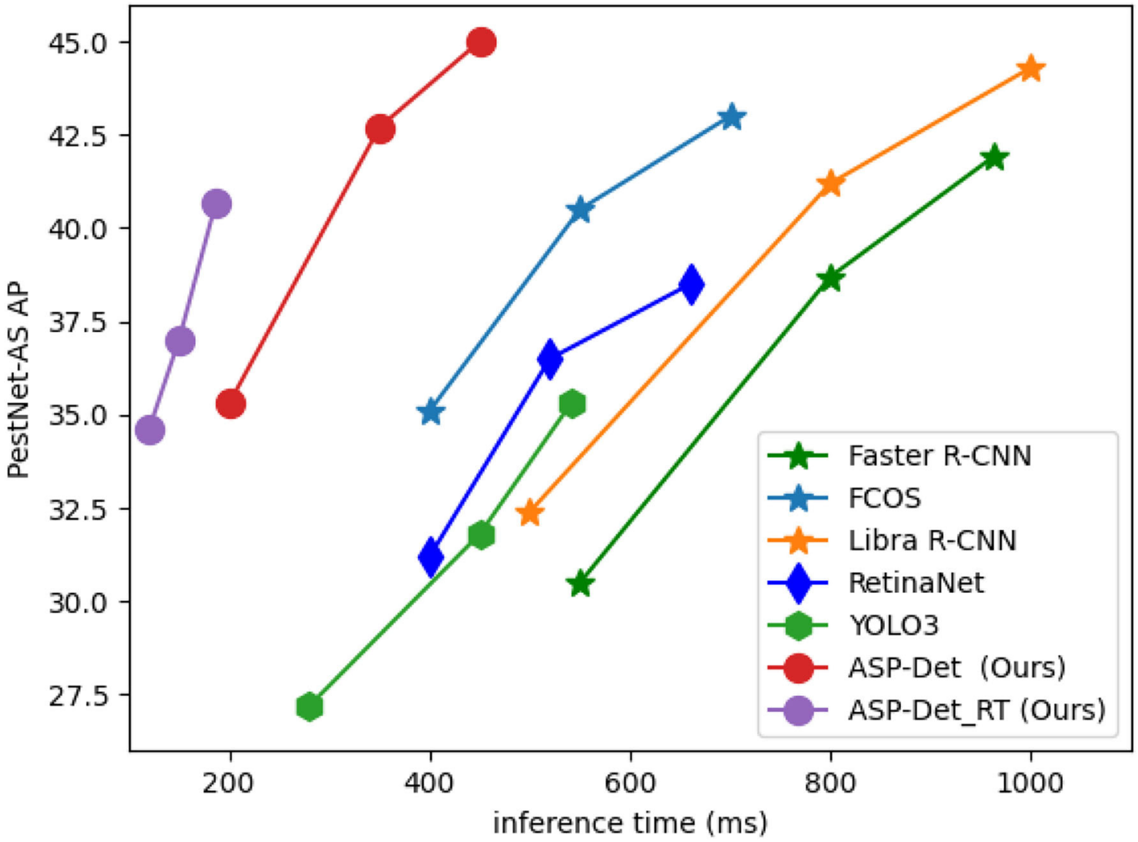


FIGURE 11 | Comparisons of efficient of different modules proposed in this paper with the-state-of-arts method on similar pest dataset on a single GPU.

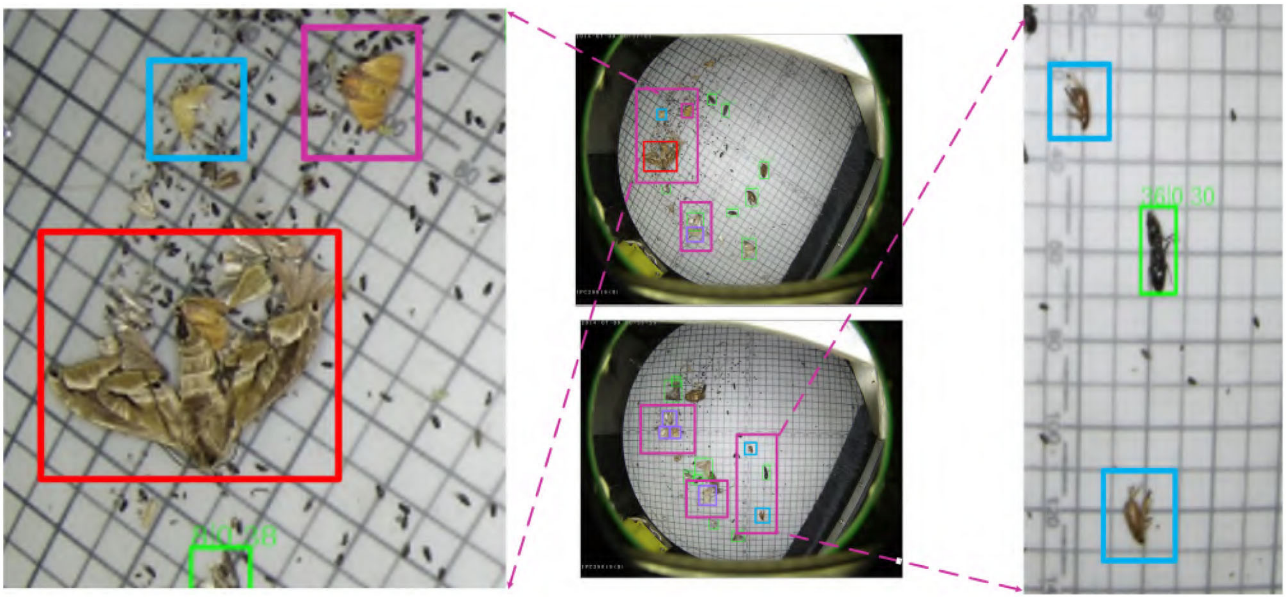


FIGURE 12 | Some problems in the ASPD-Det detection method, misclassification, or omission of detection.

plant protection to rethink the necessity of anchor boxes in object detection. Additionally, to apply our pest detection method in practice, we present some real-time models of our detector, which has excellent performance and inference speed. Given its effectiveness and efficiency, we hope that ASP-Det can serve as a solid and straightforward alternative for promoting agricultural production.

DATA AVAILABILITY STATEMENT

The data analyzed in this study is subject to the following licenses/restrictions: The data set was provided by Jiaduo Company, which is a cooperative unit of our research institution. The disclosure of the data requires the consent of the company before it can be released to the public. Requests to access these datasets should be directed to wangfenmei205@126.com.

AUTHOR CONTRIBUTIONS

FW is responsible for overall model building, paper writing, and dataset training and modeling. LL is responsible for guiding the writing of the thesis and the construction of the overall architecture. SD is responsible for model optimization and code debugging. SW is responsible for the derivation and verification of the model formula. ZH only needs to participate in the drawing work. HH is responsible for the curation and fabrication of the dataset. JD gave a lot of guidance in the revision process of the paper, and proofread the full text. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported in part by the project of the Dean's Fund of Hefei Institute of Physical Science, Chinese Academy of Sciences (YZJJ2022QN32) and the major special project of Anhui Province Science and Technology (2020b06050001).

ACKNOWLEDGMENTS

The authors would like to thank Jia Duo company for providing data support. Besides, thanks all the authors cited in this paper and a referee for their helpful comments and suggestions.

REFERENCES

Ayan, E., Erbay, H., and Varçin, 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

Bochkovskiy, 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

Dai, J., Li, Y., He, K., and Sun, J. (2016). "R-FCN: object detection *via* region-based fully convolutional networks," in *Advances in Neural Information Processing Systems* (Barcelona).

Dan, J., Chao, J., Chiu, L., Wu, Y., Chung, J., Hsu, J., et al. (2021). Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. *J. Appl. Entomol.* 145, 206–222. doi: 10.1111/jen.12834

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

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., and Woniak, M. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21:4749. doi: 10.3390/s21144749

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: keypoint triplets for object detection. doi: 10.1109/ICCV.2019.00667

Feng, Y. (2013). Fine-grained detection and localization of objects in images.

Jia, D., Wei, D., Socher, R., Li, L. J., Kai, L., and Li, F. F. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL), 248–255.

Jiao, L., Dong, S., Zhang, S., Xie, C., and Wang, H. (2020). AF-RCNN: an anchor-free convolutional neural network for multi-categories agricultural pest detection. *Comput. Electron. Agric*. 174:105522. doi: 10.1016/j.compag.2020.105522

Kong, T., Sun, F., Liu, H., Jiang, Y., and Shi, J. (2020). Foveabox: beyond anchor-based object detection. *IEEE Trans. Image Process*. 29, 7389–7398. doi: 10.1109/TIP.2020.3002345

Law, H., and Deng, J. (2020). Cornernet: detecting objects as paired keypoints. *Int. J. Comput. Vis*. 128, 642–656. doi: 10.1007/s11263-019-01204-1

Li, Z., Chen, W., Li, Z., and Bian, K. (2016). Look into my eyes: fine-grained detection of face-screen distance on smartphones. doi: 10.1109/MSN.2016.048

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 2117–2125. doi: 10.1109/CVPR.2017.106

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988. doi: 10.1109/ICCV.2017.324

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017c). Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell*. 99, 2999–3007. doi: 10.1109/TPAMI.2018.2858826

Lin, T. Y., Maire, M., Belongie, S., Hays, J., and Zitnick, C. L. (2014). "Microsoft coco: common objects in context," in *European Conference on Computer Vision* (Zurich). doi: 10.1007/978-3-319-10602-1_48

Liu, H., Lee, S. H., and Chahl, J. S. (2017). A multispectral 3-d vision system for invertebrate detection on crops. *IEEE Sensors J*. 2017.2757049. doi: 10.1109/JSEN.2017.2757049

Liu, L., Wang, R., Xie, C., Yang, P., and Li, R. (2019a). "Deep learning based automatic approach using hybrid global and local activated features towards large-scale multi-class pest monitoring," in *IEEE International Conference on Industrial Informatics 2019*. doi: 10.1109/INDIN41052.2019.8972026

Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., et al. (2019b). 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, X., Chen, S., Song, L., Woniak, M., and Liu, S. (2021). Self-attention negative feedback network for real-time image super-resolution. *J. King Saud Univ*. doi: 10.1016/j.jksuci.2021.07.014

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: hierarchical vision transformer using shifted windows. doi: 10.48550/arXiv.2103.14030

Lu, X., Li, B., Yue, Y., Li, Q., and Yan, J. (2020). "Grid R-CNN," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi: 10.1109/CVPR.2019.00754

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* (Seattle, WA), 779–788. doi: 10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2018). Yolov3: an incremental improvement. *arXiv [Preprint] 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. *Adv. Neural Inform. Process. Syst*. 28, 91–99. doi: 10.1109/TPAMI.2016.2577031

Shen, Y., Zhou, H., Li, J., Jian, F., and Jayas, D. S. (2018). Detection of stored-grain insects using deep learning. *Comput. Electron. Agric*. 145, 319–325. doi: 10.1016/j.compag.2017.11.039

Sivakoff, F. S., Rosenheim, J. A., and Hagler, J. R. (2012). Relative dispersal ability of a key agricultural pest and its predators in an annual agroecosystem. *Biol. Control* 63, 296–303. doi: 10.1016/j.biocontrol.2012.09.008

Sun, Y., Liu, X., Yuan, M., Ren, L., Wang, J., and Chen, Z. (2018). Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring. *Biosyst. Eng*. 176, 140–150. doi: 10.1016/j.biosystemseng.2018.10.012

Symeonidis, C., Mademlis, I., Nikolaidis, N., and Pitas, I. (2019). "Improving neural non-maximum suppression for object detection by exploiting interest-point detectors," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* (Pittsburgh, PA). doi: 10.1109/MLSP.2019.8918769

Tetila, E. C., Machado, B. B., Astolfi, G., de Souza Belete, N. A., Amorim, W. P., Roel, A. R., et al. (2020). Detection and classification of soybean pests using deep learning with UAV images. *Comput. Electron. Agric*. 179:105836. doi: 10.1016/j.compag.2020.105836

Tian, Z., Shen, C., Chen, H., and He, T. (2019). "FCOS: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 9627–9636. doi: 10.1109/ICCV.2019.00972

Wang, F., Wang, R., Xie, C., Yang, P., and Liu, L. (2020). Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition. *Comput. Electron. Agric*. 169:105222. doi: 10.1016/j.compag.2020.105222

Wang, R., Liu, L., Xie, C., Yang, P., and Zhou, M. (2021). Agripest: a large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors* 21:1601. doi: 10.3390/s21051601

Wang, X., Girshick, R., Gupta, A., and He, K. (2017). Non-local neural networks. doi: 10.48550/arXiv.1711.07971

Yao, Q., Jiale, G., Jun, L., Longjun, G., Jian, T., Baojun, Y., et al. (2020). Automatic detection model for pest damage symptoms on rice canopy based on improved retinanet. *Trans. Chinese Soc. Agric. Eng*. 36, 182–188. doi: 10.11975/j.issn.1002-6819.2020.15.023

Yu, J., Jiang, Y., Wang, Z., Cao, Z., and Huang, T. (2016). UnitBox: an advanced object detection network. doi: 10.1145/2964284.2967274

Yu, X., Wu, S., Lu, X., and Gao, G. (2021). Adaptive multiscale feature for object detection. *Neurocomputing* 449, 146–158. doi: 10.1016/j.neucom.2021.04.002

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA). doi: 10.1109/CVPR42600.2020.00978

Zhao, H., Jia, J., and Koltun, V. (2020). Exploring self-attention for image recognition. 10073-82. doi: 10.1109/CVPR42600.2020.01009

Zhou, X., Zhuo, J., and Krhenbühl, P. (2019). Bottom-up object detection by grouping extreme and center points. doi: 10.48550/arXiv.1901.08043

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.

ORIGINAL RESEARCH
published: 13 July 2022
doi: 10.3389/fpls.2022.814681

Tomato Pest Recognition Algorithm Based on Improved YOLOv4

Jun Liu[1]*, *Xuewei Wang*[1], *Wenqing Miao*[1] and *Guoxu Liu*[2]

[1] *Shandong Provincial University Laboratory for Protected Horticulture, Blockchain Laboratory of Agricultural Vegetables, Weifang University of Science and Technology, Weifang, China,* [2] *College of Information and Control Engineering, Weifang University, Weifang, China*

Tomato plants are infected by diseases and insect pests in the growth process, which will lead to a reduction in tomato production and economic benefits for growers. At present, tomato pests are detected mainly through manual collection and classification of field samples by professionals. This manual classification method is expensive and time-consuming. The existing automatic pest detection methods based on a computer require a simple background environment of the pests and cannot locate pests. To solve these problems, based on the idea of deep learning, a tomato pest identification algorithm based on an improved YOLOv4 fusing triplet attention mechanism (YOLOv4-TAM) was proposed, and the problem of imbalances in the number of positive and negative samples in the image was addressed by introducing a focal loss function. The K-means + + clustering algorithm is used to obtain a set of anchor boxes that correspond to the pest dataset. At the same time, a labeled dataset of tomato pests was established. The proposed algorithm was tested on the established dataset, and the average recognition accuracy reached 95.2%. The experimental results show that the proposed method can effectively improve the accuracy of tomato pests, which is superior to the previous methods. Algorithmic performance on practical images of healthy and unhealthy objects shows that the proposed method is feasible for the detection of tomato pests.

Keywords: image processing, pests identification, YOLO, object detection, tomato

OPEN ACCESS

Edited by:
Rujing Wang,
Hefei Institute of Technology Innovation, Hefei Institutes of Physical Science (CAS), China

Reviewed by:
Sijia Yu,
Rutgers, The State University of New Jersey, United States
Zongmei Gao,
Columbia University, United States

****Correspondence:***
Jun Liu
Liu_jun860116@wfust.edu.cn

Specialty section:
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

Received: *14 November 2021*
Accepted: *17 May 2022*
Published: *13 July 2022*

Citation:
Liu J, Wang X, Miao W and Liu G (2022) Tomato Pest Recognition Algorithm Based on Improved YOLOv4. Front. Plant Sci. 13:814681. doi: 10.3389/fpls.2022.814681

INTRODUCTION

Agricultural pests are known to be one of the main factors causing damage to the world's agricultural economy. As a kind of insect, they mainly depend on the survival of various plants and crops, causing different degrees of harm to agriculture, forestry, and animal husbandry. The economic impacts of agricultural pests spread worldwide. The economic losses of agriculture in Europe reached 28.2%, in North America reached 31.2%, and in Asia and Africa reached more than 50%. Since the 1960s, integrated pest control (IPM) (Parsa et al., 2014) has been the main pest control mode. IPM has formulated the best pesticide recommendations for economic development and ecological maintenance based on the results of pressure detection of different pests. Therefore, the accurate identification and location of pests are very important for IPM. At present, most detection methods are expensive and time-consuming because they require IPM professionals to collect and classify field samples manually, which prevents the developing countries that lack IPM technological support from using these technologies for pest control. Therefore, in the field of IPM, a fast and low-cost automatic detection method for agricultural pests is urgently needed.

In recent years, deep learning has developed rapidly and has attracted an increasing number of researchers' attention because of its superior performance in feature extraction, model generalization, and fitting. The convolutional neural network (CNN) in the deep learning method performs well in large-scale image recognition tasks. The biggest difference between CNN and traditional pattern recognition methods is that it automatically extracts features layer by layer from images, which can contain thousands of parameters.

At present, many pest recognition systems have been proposed by researchers. Yang et al. (2017) proposed an insect recognition model based on deep learning and image saliency analysis. On the test set of tea garden images, the average accuracy was 0.915, the running time was reduced to 0.7 ms, and the required memory was 6 MB. Shen et al. (2018) used deep neural network technology to establish the detection and recognition method of stored grain pests. Faster R-CNN was used to extract the possible insect areas in the image and classify the insects in these areas. The average accuracy was 88%. Mique and Palaoag (2018) used a CNN-based model to retrieve and compare the collected images with a pile of rice pest images. The model can achieve 90.9% of the final training accuracy. Zhong et al. (2018) designed and implemented a vision-based classification system for flying insect counting. First, yellow sticky traps were set up in the monitoring area to trap flying insects, and a camera was set up to capture images in real-time. Then, a method of object detection and rough counting based on YOLO was designed, and a support vector machine based on global features was designed. Finally, six kinds of flying insects, including bees, flies, mosquitoes, moths, scarabs, and fruit flies, were selected to evaluate the effectiveness of the system. Compared with the conventional method, the experimental results show that the method performs better, and the average classification accuracy is 90.18%. Barbedo and Castro (2019) studied the effect of image quality on the identification of psylla using CNN. A total of 1,276 images were used in the experiment. Half of them were collected using a flat panel scanner, and the other half by two different brands of smartphones. The accuracy was 70 and 90%, respectively, which shows that a more realistic environment can guarantee the robustness of the trained network. He et al. (2020) built a brown rice planthopper detection model based on deep learning and achieved good results through the improvement of faster RCNN and YOLOv3 models. The authors compared these two models under equivalent conditions and showed that the YOLOv3 model performs better and has a higher detection rate than the faster RCNN. Liu et al. (2020) fused semantic information (temperature, humidity, longitude and latitude, etc.) of pest images with CNN models and verified the advantages of the attention mechanism in solving the problem of imbalanced data.

In this study, an algorithm that can diagnose tomato pests quickly and effectively by improving the YOLO model is proposed. It can solve the problem of low diagnostic accuracy of pests encountered by tomato producers during cultivation, and has some implications for future research on tomato pest prevention, and advance the development of intelligent agriculture.

RELATED WORKS

Object Detection

Object detection refers to recognizing the corresponding object category, location, and size from a given image or video, to carry out the next analysis. Object detection algorithms based on regression do not need to generate branches from candidate regions. For a given input image, the candidate boxes and categories of objects are directly regressed at multiple positions of the image. Therefore, this research will adopt the object detection algorithm based on regression.

In 2016, the YOLO network was proposed by Redmon et al. (2016). Based on YOLO, YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), and YOLOv4 (Bochkovskiy et al., 2020) were proposed. The YOLO network, as a new and outstanding object detection technology, has been widely recommended by scholars. It needs only one neural network to detect objects. YOLO can read the whole image at a time and can recognize the local information of the image, which greatly reduces the false detection rate of the background. It has a slight decrease in accuracy compared with the most popular network, but it has a great improvement in speed. Fast YOLO has a speed of 155 frames per second, which can be well applied in the scenes with high real-time requirements. At present, YOLO has different versions, with YOLOv4 being much faster than the other versions in speed.

With the deepening of research on object detection, scholars apply the improved YOLO algorithm to the real-time detection of vehicles (Zhou et al., 2020), pedestrians (Xu et al., 2022), traffic signs (Zhou et al., 2020), ships (Tang et al., 2021), fruits (Wang and He, 2021), and so on. In addition, its application in the field of agricultural pest detection also began to appear. Zhong et al. (2018) designed a vision-based flying insect counting and classification system based on YOLO. The average counting accuracy of raspberry peel was 92.50%, and the average classification accuracy was 90.18%. He et al. (2020) proposed a rapid and accurate detection algorithm for brown rice planthopper, Yolov3. The average recall rate was 49.60%, and the average accuracy rate was 96.48%. Zha et al. (2021) proposed the YOLOv4_ MF model to detect forestry pests. The experimental results showed that compared with the YOLOv4 model, the mAP of the proposed model was 4.24% higher. Xin and Wang (2021) used YOLOv4 to test and verify images after quality level classification, and the recognition accuracy was 95%, which was much higher than the basic 84% of the DCNN model.

Compared with other CNN networks that use sliding classifiers, YOLO is a unified network that can simultaneously predict the location, size, and category of objects. It is a real-time object detection system based on a deep convolution neural network. As the YOLO network has the characteristics of end-to-end, the whole training and detection process from data input to result in output is completed in the network model, so it can guarantee accuracy and show a faster detection speed. So, this study combines the idea of YOLOv4 to detect pests.

Attention Mechanism

Attention mechanisms play an important role in human perception (Corbetta and Shulman, 2002). An important property of the human visual system is that the entire scene cannot be processed simultaneously. Instead, to better capture the visual structures, humans utilize a range of local saccades and selectively focus on the salient parts (Zheng et al., 2015).

The introduction of attention mechanisms into CNN networks has recently been proposed in the field of object detection to improve performance on large-scale classification tasks. Wang et al. (2017) proposed a residual attention network using an encoder attention module. By refining the feature maps, the network can't only perform well but also be robust to noise inputs. Hu et al. (2018) introduced a compact attention feature extraction library using global average pooling features to calculate the information weight of channel attention. Woo et al. (2018) used an efficient architecture that simultaneously utilizes spatial and channel attention modules to focus on more information, and excellent results have been achieved. Ju et al. (2021) introduced the attention mechanism into the YOLO algorithm, and the detection accuracy has been improved. Inspired by this, this study combines the YOLO algorithm with the attention module to do further research.

The Aim of This Study

With the advancement of agricultural intelligence, object detection has achieved certain development in the agricultural field. At present, many deep learning methods for object detection are widely used in crop identification, long-range potential as well as pests and diseases detection, weed identification, fruit and vegetable quality detection, and automatic picking.

The pests that often occur in tomatoes include whiteflies, aphids, and leafminers. Once they occur, they will cause a lot of loss. Therefore, it is of great significance to identify tomato pests in order to control them in time and eliminate them in germination. The actual environment of tomato pest identification is very complex. To achieve a more effective and widely applicable pest detection technology and meet the needs of using the least and most convenient operation to complete expert-level pest detection, this study combines deep learning with tomato pest detection. To achieve the goal of rapid and highly accurate detection of images of tomato pests, this study proposed a deep learning model that is fast and can perform multi-object detection based on YOLOv4 and improved it by fusing the triplet attention (Song et al., 2018) mechanism. Experiments showed that the proposed model greatly improved the comprehensive detection ability of the images of tomato pests.

METHODOLOGY

Principle of YOLO

The YOLO algorithm treats the detection problem of an object as a regression problem of position coordinate and confidence score directly. Therefore, the YOLO algorithm can predict the category and location of multiple objects in real-time at one time. Unlike traditional object detection algorithms, which select the sliding window method and the Faster R-CNN algorithm

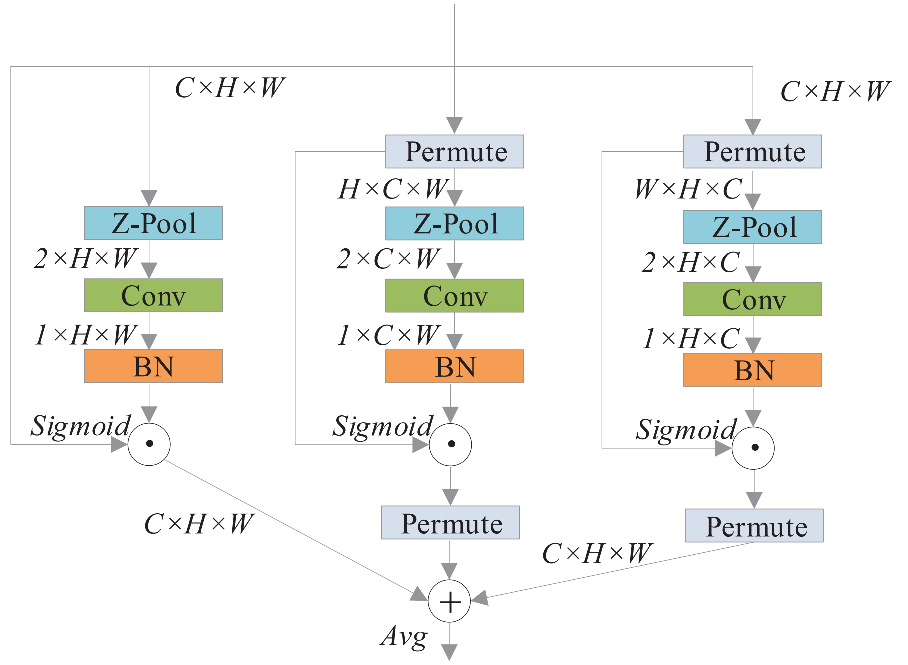


FIGURE 1 | Network structure diagram of triplet attention (Song et al., 2018).

to extract candidate regions, YOLO directly inputs the whole image into the network model for training and detection. This idea greatly improves the training and detection speed of the network model.

YOLOv4 is the fourth version of the YOLO series of algorithms. The first major improvement of the YOLOv4 model is to use CSPDarknet53 as its backbone network. CSPDarknet53 is mainly composed of the CBM module and CSP module. The CBM module is composed of the Conv, batch normalization (BN), and Mish activation functions. The CSP module contains two branches; one is the convolution of the main cadres. One is used to generate a large residual edge, which enhances the learning ability of CNN by splicing two branches across different levels and integrating channels. Another major improvement of YOLOv4 is that in the detection section, a spatial pyramid pooled layer SPP module is used, which enables any size of feature map to be converted to a fixed size feature vector, inherits the YOLOv3 approach in the prediction of the boundary box, generates *a priori* box of different scales using K-means clustering, and predicts on the feature map at different levels. The difference is that it uses the idea of PANet to fuse features at different levels.

In addition, YOLOv4 introduces mosaic augmentation. Its principle is to randomly select four images at a time and randomly scale, flip horizontally, flip vertically, and change the color gamut of the images. Then, according to a certain proportion, the four images are intercepted and stitched into a new training image. Because many objects in the real natural environment are not the detection target as the detection background, they will seriously affect the accuracy of the algorithm. So a mosaic is used to enrich the background of the detection object, which is conducive to the weight distribution of different characteristics of different pests in the training algorithm.

Triplet Attention Module

The YOLOv4 network treats the characteristics of each channel equally, which limits the detection performance of the algorithm to some extent. The tomato pest image background is complicated, and some pest targets are small in the area occupied by the image, which can easily cause misdetection. Therefore, the improvement of YOLOv4 is needed. To further improve the model accuracy, this study uses triplet attention to improve the CSPDarknet53 feature extraction network in YOLOv4. The triplet attention module (Song et al., 2018) is an inexpensive and effective attention mechanism with few parameters and does not involve dimensionality reduction. It is an additional neural network, as shown in **Figure 1**.

The triplet attention module consists of three parallel branches, two of which capture cross-dimensional interactions between channel C and space H or W. The last branch is used to build spatial attention. The output of the final three branches is aggregated on average.

This study uses the triplet attention module to improve the CSPDarknet53 network of YOLOv4, enabling the network to acquire cross-dimensional interactions through automatic learning, increasing effective feature channel weights, and thus making the network focus on important feature channels. The backbone network structure of the YOLOv4 model improved with the triplet attention module (YOLOv4-TAM) is shown in **Figure 2**.

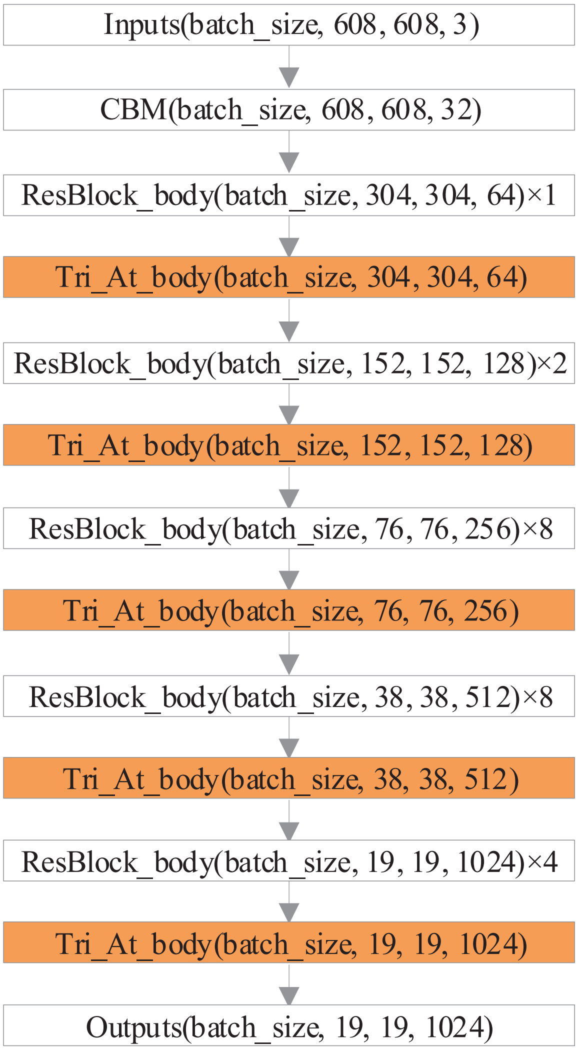

Inputs(batch_size, 608, 608, 3)
CBM(batch_size, 608, 608, 32)
ResBlock_body(batch_size, 304, 304, 64)×1
Tri_At_body(batch_size, 304, 304, 64)
ResBlock_body(batch_size, 152, 152, 128)×2
Tri_At_body(batch_size, 152, 152, 128)
ResBlock_body(batch_size, 76, 76, 256)×8
Tri_At_body(batch_size, 76, 76, 256)
ResBlock_body(batch_size, 38, 38, 512)×8
Tri_At_body(batch_size, 38, 38, 512)
ResBlock_body(batch_size, 19, 19, 1024)×4
Tri_At_body(batch_size, 19, 19, 1024)
Outputs(batch_size, 19, 19, 1024)

FIGURE 2 | Network structure diagram of the proposed model.

The New Loss Function

During the loss value calculation in YOLOv4, the detector divides the prediction box into positive and negative samples. The predicted box with the largest IOU value from the annotated box is divided into positive samples, and predicted boxes with all annotated boxes having IOU less than 0.5 are classified as negative samples. The small object occupies far fewer pixels in the image than the background does, resulting in a large difference in the number of positive and negative samples during training.

To this end, this study addresses the problem of imbalances in the number of positive and negative samples in the image by introducing a focal loss function, which is shown in the following formula:

$$
\begin{aligned}
Loss &= Loss_{coord} + Loss_{obj} + Loss_{class} \\
&= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} \left(2 - w_i \times h_i\right)\left(1 - CIOU\right) - \\
&\lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} \left|C_i - \hat{C}_i\right|^{\beta} \cdot \left[\alpha \hat{C}_i \log\left(C_i\right) + (1-\alpha)\left(1 - \hat{C}_i\right)\right. \\
&\left. \cdot \log\left(1 - C_i\right)\right] - \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{noobj} \left|C_i - \hat{C}_i\right|^{\beta} \cdot \left[\alpha \hat{C}_i \log\left(C_i\right)\right. \\
&\left. + (1-\alpha)\left(1 - \hat{C}_i\right) \cdot \log\left(1 - C_i\right)\right] - \lambda_{obj} \sum_{i=0}^{S^2} \sum_{c \in class} l_{ij}^{obj} \\
&\left[\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))\right] \qquad (1)
\end{aligned}
$$

In the abovementioned formula, λ_{coord} is the weight coefficient of the coordinate prediction. w_i and h_i are the width and height of the annotation box, respectively. Complete intersection over union (CIOU) is a new IOU that has added the penalty coefficient of the annotation box and the predicted box. λ_{obj} is the weight coefficient when there is an object. λ_{noobj} is the weight coefficient when there is no object. α is used to balance positive and negative sample numbers, and this study takes the value of 0.75. β is used to moderate the weight of difficult and simple samples, and this study takes the value of 2. S^2 is the number of grids. B is the number of predicted boxes in each grid. $\hat{C}_i$ and C_i are the confidence scores of the predicted box vs. true box, respectively. $\hat{p}_i(c)$ and $p_i(c)$ are the probability values for the category of the predicted box vs. true box, respectively. l_{ij}^{noobj} indicates that the object does not belong to the j bounding box of the i grid. l_{ij}^{obj} indicates that the object belongs to the j bounding box of the i grid.

The New Anchor Boxes

Since the original YOLOv4 network was experimented on the VOC dataset, the original anchor box mechanism was set for the VOC dataset. For pest detection, utilizing the original anchor box mechanism would affect the IOU value, resulting in the inability to screen out the optimal prediction box. Therefore, the anchor box mechanism in the original YOLOv4 network needs to be improved. The K-means + + clustering algorithm can randomly generate clustering centers, which ensures a discrete type of initial cluster center, elevating the effect of anchor box generation. So the K-means + + clustering method is used to randomly choose the center of the sample and locate the anchor box for pest images. The new anchor boxes are obtained, including (13, 15), (19, 22), (23, 28), (44, 49), (52, 56), (64, 67), (87, 93), (102, 116), and (126, 139).

EXPERIMENTS

The experimental step flow of the study is shown in **Figure 3**.

Dataset Collection

The main pests harming tomatoes in greenhouses are whiteflies, aphids, and leafminers. The pest image acquisition apparatus was installed in the Shouguang tomato greenhouse (36.8N, 118.7E) for this experiment (**Figure 4**). The yellow insect induction plate was utilized to attract the pests according to the principle of pest chemotaxis, and then the pests were glued by the high viscosity on the plate to achieve the trapping effect, by timed photographing the image of the insect induction plate and transmitting the image to the computer PC end for processing.

The image acquisition time of pests was from 22 October 2019 to 30 December 2020, and the species of pests captured by the induced insect plate were comprehensive and large in number. A total of 10 mutagen plates with a length of 35 cm and a width of 25 cm were suspended in the greenhouse and replaced every 5 days, and images of the mutagen plates were captured using an image acquisition device. The acquired image

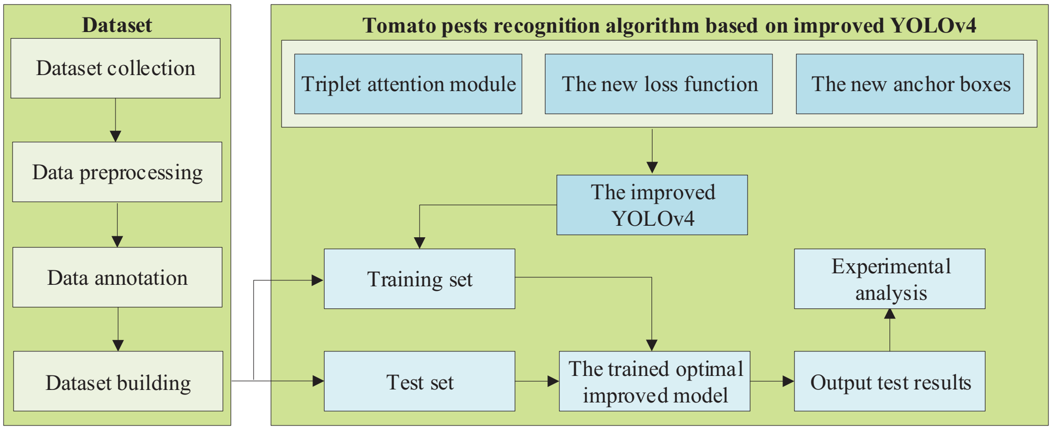


FIGURE 3 | The experimental step flow of the study.

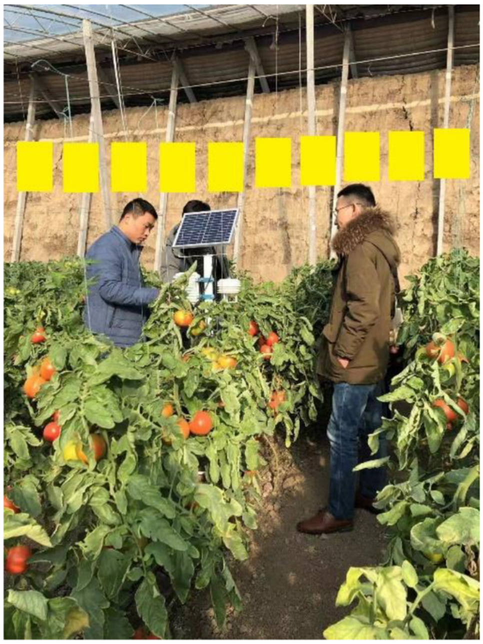


FIGURE 4 | The experimental image acquisition site.

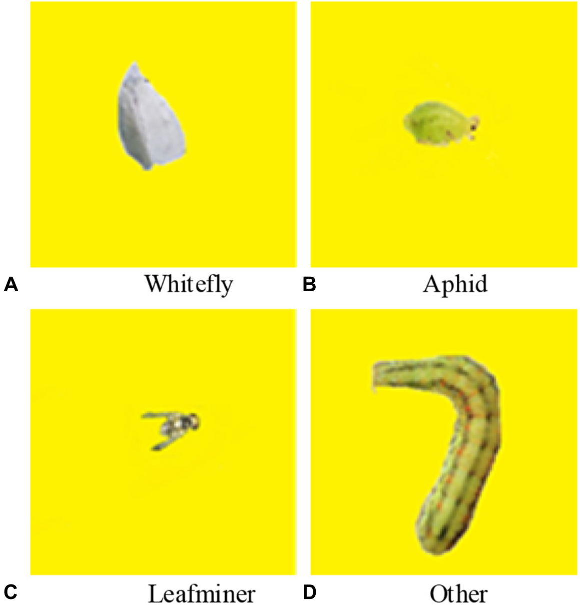


FIGURE 5 | Examples of input images used in this study.

TABLE 1 | Information on tomato pest dataset.

Class	Pests class	Labeling quantity
1	Whitefly	6327
2	Aphid	5687
3	Leafminer	6912
4	Other	6679

size was 1,960 × 1,080, and the image storage format was jpg. To make the experiment more closely resemble the real farm environment, all images were taken under natural conditions, and the adhered pests on the induced plate were cleaned up regularly by a dedicated person. A total of 2,893 images of induced plate pests were acquired for this experiment.

Data Pre-processing

To further enrich the sample data while making up for the size and distribution limitations of pest targets and allow the model to achieve a better training effect, this study preprocessed the sample data. Mosaic, image rotation, multiscale cropping and magnification, image translation, image mirroring, and image denoising were used for data enhancement. After data pre-processing, the position distribution situation of the pest targets was enriched, and the small-size targets were enlarged to some extent, thus improving the generalization ability and training efficiency of the model.

Data Annotation

This experimental label was mainly divided into 4 categories, which were whiteflies, aphids, leafminers, and other large pests. The main purpose of classifying other large pests into one category was to explore the potential pest outbreak because large pests have a strong migration ability and are prone to large pest invasions in real-life production, which can increase the stress resistance of the algorithm when applied in practice. The sample number of pests in the image of the induced insect plate is huge, the situation when the occurrence of pests adhesion leads to an unclear separation is much lower than the situation when the pests are at an independent stage, and the removal of the number of the attached pests in the actual production does not affect the overall induced insect plate pests warning, so this study will only label the independent pests. The images were annotated using labeling, and the number of samples of whiteflies, aphids, potential leaf flies, and other large pests was 6,327, 5,687, 6,912, and 6679, respectively, as shown in **Table 1** and **Figure 5**. Finally, 70% of images were randomly selected to construct the training set, 20% of images were used as the verification set, and the remaining images were used as the test set.

Experimental Operation Environment

To better evaluate the performance of the proposed algorithm, it was compared with other pest recognition algorithms based on existing popular object detection methods, including DPM, R-CNN, Fast R-CNN, Faster R-CNN, and SSD, and the simulation platform configuration is shown in **Table 2**.

Evaluating Indicator

In the field of object detection, according to the research emphasis, the evaluation indexes can be different. The commonly used evaluation indexes include detection accuracy, efficiency, speed, positioning accuracy, and so on. This experiment mainly evaluates the model according to detection accuracy and detection speed.

TABLE 2 | Configuration of an experimental platform.

Server	CPU Processor: INTEL I7-9800X
	GPU: GEFORCE GTX1080Ti
	Memory: The Kingston 32G DDR4
Software	Operating System: Ubuntu 18.04
	Language: Python
	GCC 7.3.0
	CUDA 10.0.130
	OpenCV 3.4.5

Among them, GPU acceleration was used for CUDA programming, and OpenCV was mainly used to display images during testing.

(1) Detection accuracy

① mAP (mean average precision).

Usually, mAP is used as the evaluation criterion for detection accuracy. First, the average accuracy of each category in the dataset needs to be calculated as follows:

$$P_{average} = \frac{1}{R}\sum_{j=1}^{n} I_j \cdot \frac{R_j}{j} \tag{2}$$

In the above formula, R represents the number of objects related to a category in the dataset (including detected and undetected), and n represents the number of objects in the dataset. If object j is relevant, then $I_j = 1$; if object j is irrelevant, then $I_j = 0.R_j$ represents the number of related objects in the first j objects. Then the average of the average precision of multiple categories is taken as mAP:

$$mAP = \frac{P_a verage}{N_{(class)}} \tag{3}$$

$N(class)$ represents the number of all the categories. The larger the mAP value, the higher the monitoring accuracy of the algorithm; conversely, the lower the accuracy of the algorithm.

② Average precision (AP).

First, we need to introduce the precision-recall (PR) curve: the horizontal axis recall of the PR curve represents the ability of the classifier to cover the positive samples; the vertical axis precision represents the accuracy of the classifier to predict positive samples. Then the PR curve represents the trade-off between the accuracy of recognition of positive cases and the coverage ability of positive cases. AP is the area of the image enclosed by the PR curve and the horizontal axis.

For continuous PR curves:

$$AP = \int_0^1 PRdr \tag{4}$$

For discrete PR curves:

$$AP = \sum_{k=1}^{n} Pk\Delta rk \tag{5}$$

(2) Detection speed

Frames per second (FPS) is used to evaluate the detection speed. The more the FPS, the faster the detection speed of the algorithm is, otherwise, the slower the detection speed of the algorithm is.

TABLE 3 | Comparison of training results of six models.

Object detection algorithms	mAP	FPS
Faster R-CNN	68.7	9
SSD	72.3	43
YOLOv3	73.6	71
YOLOv4	87.1	82
The proposed algorithm	93.4	83

TABLE 4 | Proportion of detection errors (%) for the six algorithms.

Algorithms	Number of false checks	Misdetection rate/%
Faster R-CNN	190	1.27%
SSD	65	0.43%
YOLOv3	71	0.47%
YOLOv4	63	0.42%
The proposed algorithm	54	0.36%

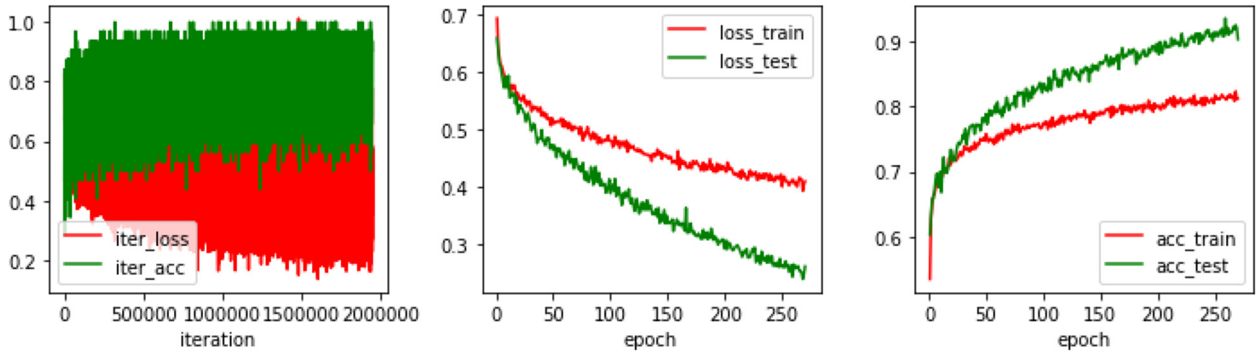


FIGURE 6 | Process of model training.

TABLE 5 | Algorithmic performance on practical images of healthy and unhealthy objects.

Pests class	AP (%)
Whitefly	84.7
Aphid	83.9
Leafminer	62.7
Other	89.6
mAP (%)	78.1

EXPERIMENTAL RESULTS AND ANALYSIS

Model Training

Before training on the model, some initial settings are required. The values of hyperparameters must first be determined. In this experiment, the value of the batch is set to 32, and the value of the subdivisions is set to 16. That is, 2 images are passed into the network each time, 32 images are processed, and the model is updated and trained again with parameters. So, one epoch is for every 32 images. The learning rate is set to 0.0001, the weight delay is set to 0.0005, and the momentum is set to 0.9. After the first training, the prediction result of the network is not ideal enough. Through training with multiple epochs, a satisfactory training effect is produced. **Figure 6** shows the training process. It can be seen that after training with 200 epochs, the loss of the network model decreases and stabilizes in a stepwise manner, i.e., a relatively satisfactory effect can be achieved after 200 epochs, and the training is continued in the experiment until the loss convergence is close to 0.

Performance Comparison of Different Object Detection Algorithms

The experiment was carried out on the Darknet53 network. Faster R-CNN, SSD, YOLOv3, YOLOv4, and the proposed algorithm are the comparison algorithm. The five network model parameters are initialized by using the pre-training network model.

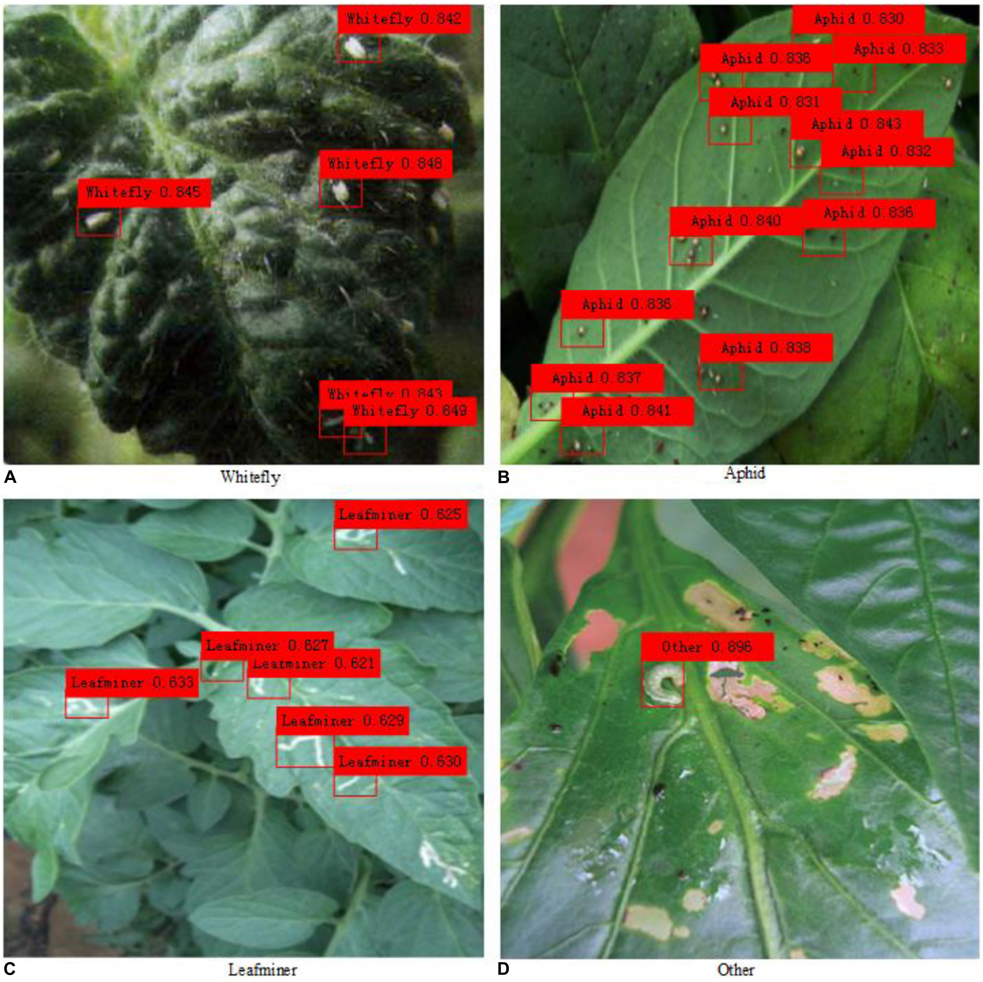


FIGURE 7 | Detection effect of practical images of healthy and unhealthy objects.

As shown by comparing the proposed algorithm with the other five algorithms in **Table 3**, the detection accuracy of the proposed algorithm is better than the other algorithms. Furthermore, in terms of detection speed, the proposed algorithm has an absolute advantage, which shows that the proposed algorithm can effectively carry out real-time detection.

Table 4 shows the proportion of detection errors for the six algorithms, with the proposed algorithm having the lowest error detection rate, only 0.36%. In consequence, the proposed algorithm in this study has a low false detection rate.

Algorithmic Performance on Practical Images of Healthy and Unhealthy Objects

The algorithmic performance on practical images of healthy and unhealthy objects is shown in **Table 5**.

As shown in **Table 5**, the AP of other pests is the highest and reaches 89.6%. However, the AP of leafminers is the lowest and only reaches 62.7%. The main reason for the large difference in detection accuracy between the two pests is the difference in pest image samples. The bodies of other pests are relatively large, and the number of pests in a single image is less, whereas the bodies of leafminers are relatively small, the number of pests in a single image is greater, and many are stacked together, resulting in greater detection difficulty and smaller AP. The mAP of the four pests reaches 78.1%, which has met the accuracy requirements of practical application and which shows that the proposed method is feasible for the detection of pests.

The actual detection effect comparison of pest images is shown in **Figure 7**. The detection results of all pest objects in the figure are marked with color rectangular boxes. It can be seen intuitively that the proposed algorithm has better detection results for images with large pests, while for images with dense small pests, the pest detection results are slightly worse, and some pests cannot be detected.

CONCLUSION AND FUTURE DIRECTIONS

Conclusion

In response to the problems of partial miss detection combined with poor detection accuracy that exists when using the YOLOv4 network to directly detect tomato pest images, this study proposes an improved YOLOv4 object detection method that employs a triplet attention mechanism and addresses the problem of imbalances in the number of positive and negative samples in the image by introducing a focal loss function. The experiment shows that the proposed model greatly improves the comprehensive performance on the image detection task of tomato pests based on not only increasing the complexity of the model on a small scale but also guaranteeing the real-time of the model, which is of great significance to reduce and prevent the incidence chance of tomato pests. Compared with other methods based on deep learning, this method can maintain high accuracy and has very prominent real-time performance, and can effectively identify the type and location of pests on the images with a small false detection rate and good robustness.

Future Directions

Although good experimental results have been achieved in this study for image recognition research of tomato pests, it is of great significance for tomato pest prediction and control. Because of the limited time, other things need further research:

(1) Current research focuses on the processing of static images, and how image recognition techniques can be applied in videos, integrated with monitoring devices is something to be investigated next. The application of image recognition technology in videos requires that the algorithms process fast, have high accuracy rates, and have requirements such as automation, continuity, and so on. It is difficult to meet the requirements only with the computational quantity of current algorithms. Borrowing from pedestrian detection methods is a feasible direction and requires further research.

(2) The sample size of the tomato pest image dataset established in this study is relatively large or far from that of standard-scale image datasets frequently used by the deep learning community, and the dataset size should be greatly expanded in future studies. It is also evident that the manual method cannot be adopted for the annotation of datasets alone, but in combination with existing detection models to automatically annotate new pest images, followed by corresponding manual corrections so that the combination of machine and manual annotation can greatly reduce the cost and time of work. Then the optimization and boosting of the object detection model should be studied in terms of a sufficiently capacitated dataset.

(3) The study of new algorithms need further research. It can be found that scientific development must have been helical. New algorithms can drive innovation of the whole technology, but there is always a validity period. There are many other ways to effectively optimize the model that still need to be attempted. In addition, how to solve the problem of pest adhesion and reduce the detected repeat box in the follow-up work will be the next research direction.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

JL and GL designed research and developed the detection dataset. JL and XW conducted the experiments, data analysis, and wrote the manuscript. GL and WM revised the manuscript. All authors read and approved the manuscript.

FUNDING

This study was supported by the Facility Horticulture Laboratory of Universities in Shandong (2019YY003), Key Research and Development Plan of Shandong Province (2020RKA07036 and 2019GNC106034), Shandong Social Science Planning Project (21CPYJ20), Natural Science Foundation of Shandong Province (ZR2021QC173), and Weifang Science and Technology Development Plan (2021 GX054).

REFERENCES

Barbedo, J. G. A., and Castro, G. B. (2019). Influence of image quality on the identification of psyllids using convolutional neural networks. *Biosyst. Eng.* 182, 151–158. doi: 10.1016/j.biosystemseng.2019.04.007

Bochkovskiy, A., Wang, C. Y., and Liao, H. (2020). Yolov4: optimal speed and accuracy of object detection. *arXiv* [Preprint]. doi: 10.48550/arXiv.2004.10934

Corbetta, M., and Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nat. Rev. Neurosci.* 3:201. doi: 10.1038/nrn755

He, Y., Zhou, Z., Tian, L., Liu, Y., and Luo, X. (2020). Brown rice planthopper (nilaparvata lugens stal) detection based on deep learning. *Precis. Agric.* 21, 1385–1402. doi: 10.1007/s11119-020-09726-2

Hu, J., Li, S., and Gang, S. (2018). "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2018.00745

Ju, M., Luo, J., Wang, Z., and Luo, H. (2021). Adaptive feature fusion with attention mechanism for multi-scale target detection. *Neural Comput. Applic.* 33, 2769–2781. doi: 10.1007/s00521-020-05150-9

Liu, L., Xie, C., Wang, R., Yang, P., Sudirman, S., Zhang, J., et al. (2020). Deep learning based automatic multi-class wild pest monitoring approach using hybrid global and local activated features. *IEEE Transact. Ind. Inform.* 17, 7589–7598. doi: 10.1109/TII.2020.2995208

Mique, E. L., and Palaoag, T. D. (2018). "Rice pests and disease detection using convolutional neural network," in *Proceedings of the 2018 International Conference on Information Science and System* (New York, NY: ACM). doi: 10.1145/3209914.3209945

Parsa, S., Morse, S., Bonifacio, A., Chancellor, T. C. B., Condori, B., Crespo-Pérez, V., et al. (2014). Obstacles to integrated pests management adoption in developing countries. *Proc. Natl. Acad. Sci. U. S. A.* 111, 3889–3894. doi: 10.1073/pnas.1312693111

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the Computer Vision & Pattern Recognition* (Las Vegas, NV: IEEE). doi: 10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2017). "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition* (Honolulu, HI: IEEE), 6517–6525. doi: 10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). Yolov3: an incremental improvement. *arXiv* [Preprint]. doi: 10.48550/arXiv.1804.02767

Shen, Y., Zhou, H., Li, J., Jian, F., Jayas, D. S., et al. (2018). Detection of stored-grain insects using deep learning. *Comp. Elect. Agric.* 145, 319–325. doi: 10.1016/j.compag.2017.11.039

Song, H., Willi, M., Thiagarajan, J. J., Berisha, V., and Spanias, A. (2018). Triplet Network with Attention for Speaker Diarization. *arXiv* [Preprint]. doi: 10.48550/arXiv.1808.01535

Tang, G., Zhuge, Y., Claramunt, C., and Men, S. (2021). N-yolo: a sar ship detection using noise-classifying and complete-target extraction. *Remote Sens.* 13:871. doi: 10.3390/rs13050871

Wang, D., and He, D. (2021). Channel pruned yolo v5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* 210, 271–281. doi: 10.1016/j.biosystemseng.2021.08.015

Wang, F., Jiang, M., Chen, Q., Yang, S., and Tang, X. (2017). "Residual attention network for image classification," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2017.683

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). "CBAM: convolutional block attention module," in *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, Vol. 11211, eds V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer). doi: 10.1371/journal.pone.0264551

Xin, M., and Wang, Y. (2021). Image recognition of crop diseases and insect pests based on deep learning. *Wireless Commun. Mobile Comput.* 2021, 1–15. doi: 10.1155/2021/5511676

Xu, H., Guo, M., Nedjah, N., Zhang, J., and Li, P. (2022). Vehicle and pedestrian detection algorithm based on lightweight yolov3-promote and semi-precision acceleration. *IEEE Transact. Intell. Transport. Syst.* 99, 1–12. doi: 10.1109/TITS.2021.3137253

Yang, G., Bao, Y., and Liu, Z. (2017). Localization and recognition of pests in tea plantation based on image saliency analysis and convolutional neural network. *Transact. Chinese Soc. Agric. Eng.* 33, 156–162.

Zha, M., Qian, W., Yi, W., and Hua, J. (2021). A lightweight yolov4-based forestry pest detection method using coordinate attention and feature fusion. *Entropy* 23:1587. doi: 10.3390/e23121587

Zheng, Y., Zemel, R. S., Zhang, Y. J., and Larochelle, H. (2015). A neural autoregressive approach to attention-based recognition. *Int. J. Comput. Vision* 113, 67–79. doi: 10.1007/s11263-014-0765-x

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

Zhou, L., Min, W., Lin, D., Han, Q., and Liu, R. (2020). Detecting motion blurred vehicle logo in iov using filter-deblurgan and vl-yolo. *IEEE Transact. Veh. Technol.* 69, 3604–3614. doi: 10.1109/TVT.2020.2969427

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.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 08 September 2022
DOI 10.3389/fpls.2022.957961

OPEN ACCESS

EDITED BY
Peng Chen,
Anhui University, China

REVIEWED BY
K. Ezzaeri,
Université Ibn Zohr, Morocco
Suneet Kr Gupta,
Bennett University, India

*CORRESPONDENCE
Saleh Albahli
salbahli@qu.edu.sa

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

RECEIVED 31 May 2022
ACCEPTED 12 August 2022
PUBLISHED 08 September 2022

CITATION
Albahli S and Nawaz M (2022) DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification. *Front. Plant Sci.* 13:957961. doi: 10.3389/fpls.2022.957961

DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification

Saleh Albahli[1]* and Marriam Nawaz[2,3]

[1]Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia, [2]Department of Computer Science, University of Engineering and Technology–Taxila, Taxila, Pakistan, [3]Department of Software Engineering, University of Engineering and Technology–Taxila, Taxila, Pakistan

Early recognition of tomato plant leaf diseases is mandatory to improve the food yield and save agriculturalists from costly spray procedures. The correct and timely identification of several tomato plant leaf diseases is a complicated task as the healthy and affected areas of plant leaves are highly similar. Moreover, the incidence of light variation, color, and brightness changes, and the occurrence of blurring and noise on the images further increase the complexity of the detection process. In this article, we have presented a robust approach for tackling the existing issues of tomato plant leaf disease detection and classification by using deep learning. We have proposed a novel approach, namely the DenseNet-77-based CornerNet model, for the localization and classification of the tomato plant leaf abnormalities. Specifically, we have used the DenseNet-77 as the backbone network of the CornerNet. This assists in the computing of the more nominative set of image features from the suspected samples that are later categorized into 10 classes by the one-stage detector of the CornerNet model. We have evaluated the proposed solution on a standard dataset, named PlantVillage, which is challenging in nature as it contains samples with immense brightness alterations, color variations, and leaf images with different dimensions and shapes. We have attained an average accuracy of 99.98% over the employed dataset. We have conducted several experiments to assure the effectiveness of our approach for the timely recognition of the tomato plant leaf diseases that can assist the agriculturalist to replace the manual systems.

KEYWORDS

CornerNet, classification, DenseNet, tomato plant diseases, localization

Introduction

In accordance with a report issued by the Food and Agriculture Organization (FAO) of the United Nations, the population of humans will undergo a tremendous increase around the globe to 9.1 billion by 2050. Such an increase in the number of humans will also raise the demand for food (Bruinsma, 2009). Meanwhile, the decrease in agricultural land and the unavailability of clean water will limit the progress of nutriment amounts. Therefore, there is an urgent demand for improving food yields by consuming minimum cultivation space to fulfill the necessities of humans. The occurrence of several crop abnormalities results in a substantial decline in both the yield and quality of food. Hence, the timely recognition of such plant diseases is required as these diseases can affect the profit of farmers and can increase the purchase cost of food. Such implications can introduce economic instability in the markets. Moreover, the plant diseases at their adverse stage can destroy the crops which can create a starvation scenario within a region, specifically in low-income countries. Plant inspections are generally carried out with the help of human experts. However, this is a cumbersome and time-consuming activity that relies upon the presence of area experts. These plant examination procedures are not considered very reliable and it is practically impossible for humans to inspect every plant separately (Pantazi et al., 2019). To enhance the quantity and quality of food, there is a need to timeously and correctly recognize the various plant diseases which can also force the farmers into using the costly spray methods. To tackle the above-mentioned problems of manual processes, the research community is focusing on the development of automated plant disease detection and classification systems (Wolfenson, 2013).

The focus of this paper is the recognition of several tomato plant diseases as tomato has the largest consumption rate, of 15 kg per capita within a year when compared to other vegetables such as rice, potato, and cucumber. Moreover, the tomato crop counts for 15% of the entire vegetable ingestion globally (Chowdhury et al., 2021). Further, tomatoes have the highest cultivation rate with an annual growth rate of 170 tons worldwide (Valenzuela and Restović, 2019). The leading countries for its production are Egypt, India, the United States, and Turkey (Elnaggar et al., 2018). In a study conducted by the FAO (Sardogan et al., 2018), the occurrence of several tomato plant diseases caused a severe reduction in its quantity and most of the abnormalities originated from the leaves of tomato plants. It has been observed that such diseases reduce the tomato food quantity from 8 to 10% annually (Sardogan et al., 2018). Farmers or agriculturalists can guard against these huge monetary losses by adopting automated systems which can assist them in the timely detection of plant diseases and taking proactive measures. At first, technology experts utilized the methods used in the field of molecular biology and immunology for locating the presence of tomato plant leaf diseases (Sankaran et al., 2010; Dinh et al., 2020). However, these techniques were not fruitful due to their high processing requirements and dependence on the expertise of humans. Most agriculturists belong to poor or under-developed countries where the adaptability of such an expensive solution is not affordable (Patil and Chandavale, 2015; Ferentinos, 2018). The rapid progression in the area of machine learning (ML) has introduced low-cost solutions for the recognition of tomato plant diseases (Gebbers and Adamchuk, 2010). Many researchers have tested the conventional ML methods, such as hand-coded approaches, in the field of agriculture (Gebbers and Adamchuk, 2010). The availability of economical image-capturing gadgets has assisted researchers to take pictures in real-time and then give intelligent predictions *via* using ML-based approaches. Examples of such approaches include K-nearest neighbors (KNN), decision trees (DT) (Rokach and Maimon, 2005), and support vector machines (SVM) (Joachims, 1998), which are heavily evaluated by researchers for plant disease classification. Such techniques are simple in their architecture and can work well with a small amount of training data. However, they are unable to contend with image distortions such as intensity variations, color changes, and brightness alterations of suspected samples. Furthermore, the conventional approaches always impose a trade-off among the classification performance and processing time (Bello-Cerezo et al., 2019).

The empowerment of DL frameworks has assisted the researchers in dealing with the problems of conventional ML approaches (Agarwal et al., 2021d, 2022). Several DL techniques such as CNN (Roska and Chua, 1993), recurrent neural networks (RNNs) (Zaremba et al., 2014), and long short-term memory (LSTM) (Salakhutdinov and Hinton, 2009) have been found to be reliable in recognizing plant leaf diseases. The DL approaches are inspired by the human brain and can learn to discriminate between a set of image features without relying on the intervention of domain experts. These frameworks recognize the objects in the same way as humans by visually examining several samples to accomplish a pattern recognition task. Because of such properties, the DL approaches are found to be more suitable in areas of agriculture, including plant disease classification (Gewali et al., 2018). Several well-known DL frameworks such as GoogLeNet (Szegedy et al., 2015), AlexNet (Yuan and Zhang, 2016), VGG (Vedaldi and Zisserman, 2016), and ResNet (Thenmozhi and Reddy, 2019) have been thoroughly tested for accomplishing several jobs in farming, i.e., estimating food yield, crop heads recognition, fruit totaling, plant leaf disease detection and categorization, among others. Such approaches show reliable performance by minimizing the processing complexity as well as by better analyzing the topological information of the input samples.

Numerous techniques have been evaluated to identify and classify tomato leaf diseases. However, the reliable and timely recognition of such abnormality is a complicated job because of the significant color resemblance between the healthy and diseased areas of plant leaves (Paul et al., 2020). Furthermore, the intense changes in the dimension of plant leaves, lightning conditions, the incidence of noise, and blurring in the input samples further problematize the disease recognition procedure. Hence, there is a need for a more reliable system to accurately perform the plant disease classification process with minimum time constraints. To deal with these issues, we have introduced a DL approach, namely the custom CornerNet model. We have utilized Dense-77 as the backbone of the CornerNet model for extracting the image features. These are later classified by the one-stage detection module of the CornerNet model. We have conducted extensive evaluation over a challenging dataset and confirm that our approach is proficient in classifying the numerous types of tomato plant leaf diseases. The major contributions of the proposed approach are listed as:

1. Modified an object detection approach named CornerNet for tomato plant leaf abnormality categorization which improves the classification performance with an accuracy value of 99.98%.
2. Exhibits robust performance for 10 classes of the tomato plant leaf diseases because of the empowerment of the custom CornerNet model to tackle the over-fitted model training data.
3. A cost-effective solution is presented for the classification of tomato plant leaf abnormalities which minimizes the test time to 0.22 s.
4. Efficient localization of diseased regions from the tomato plant samples due to the better keypoints calculation power of the Dense-77-based CornerNet model with the mean average precision (mAP) value of 0.984.
5. In contrast to several new methods, extensive experimentation has been carried out on a challenging database named the PlantVillage dataset to exhibit the robustness of the proposed work.
6. The presented work is capable of correctly identifying the abnormal area of the tomato plant leaves even from the distorted samples and under the influence of size, color, and light variations.

The article is structured as follows: existing studies are compared in section “Related work,” the details of the introduced approach are described in section “Materials and methods,” section “Results” contains the results, and the conclusion is drawn in section “Conclusion.”

Related work

In this section, we review existing studies that have attempted to classify tomato plant leaf diseases. Typically, the approaches for tomato plant leaf disease detection and classification are either conventional ML-based techniques or DL frameworks. Hand-coded features computation approaches with the ML-based classifiers were explored initially for the plant leaf disease classification. One such framework was presented in Le et al. (2020) where the suspected images were initially processed by applying the morphological opening and closing techniques to remove the undesired objects. Then, the filtered local binary pattern method, namely the k-FLBPCM, was used on the processed images to obtain the desired feature vector. Finally, the SVM classifier was trained on the computed features for classification. The technique in Le et al. (2020) improved classification results for the plant leaf diseases but was unable to show better results on the distorted samples. Another work, namely Directional Local Quinary Patterns (DLQP), was introduced in Ahmad et al. (2020) to extract the keypoints from the input images. The work also used the SVM classifier on the computed features for categorizing the several classes of plant leaf diseases. The solution introduced in Ahmad et al. (2020) was robust in classifying the affected areas of plant leaves into their respective groups but classification performance degraded for noisy images. Sun et al. (2019) proposed an automated solution to quickly locate the diseased portion of plant leaves. They used the Simple Linear Iterative Cluster (SLIC) algorithm for distributing the input images into numerous chunks. Then, for each block of the divided image, the GLCM approach was used to extract the features which were later combined and passed to the SVM classifier for classification. This approach (Sun et al., 2019) performed well in recognizing the several categories of plant diseases but suffered from extensive processing complexity. Another pattern recognition approach was used in Pantazi et al. (2019) where the input sample was initially segmented *via* applying the GrabCut method to locate the region of interest. Then, the LBP algorithm was applied for keypoints vector estimation. Finally, classification was carried out with the help of the SVM classifier. This technique (Pantazi et al., 2019) was proficient in locating the abnormal area of the plant leaves. However, detection performance degraded for the samples with intense noise attacks. Ramesh et al. (2018) proposed a computer-aided system for the automated detection and classification of several abnormalities of plant leaves. For feature estimation, the HOG filter was used on the input samples, and disease classification was performed using the Random Forest (RF) technique. This work, elaborated on in Ramesh et al. (2018), was found to be a lightweight solution for the recognition of plant leaf diseases but the classification accuracy required further improvements. Another technique was discussed in Kuricheti and Supriya (2019) where an ML-based approach was presented to classify the several abnormalities of the turmeric plant. In the first phase, the K-means clustering approach was used on the input sample to locate the area of interest. The GLCM algorithm was applied to this area to calculate the feature vector. Finally, the SVM classifier was adopted for classification using the computed keypoints. The work discussed in Kuricheti and Supriya (2019) showed better plant leaf disease classification results. However, detection performance degraded for images with large brightness changes. Another handcrafted feature estimation approach to recognize and categorize crop leaf diseases was found in Kaur and Education (2021). Several pattern-based approaches like the GLCM, LBP, and SIFT were used for feature vector estimation. Then, several well-known ML classifiers, named the SVM, RF, and KNN, were trained on the computed features to execute the classification task. The best results were reported for the RF classifier but the classification accuracy needed enhancement. A similar solution was elaborated on in Shrivastava and Pradhan (2021) where the fourteen color spaces approach was used to extract the keypoints from the test images with a length of 172. Then, the calculated keypoints were passed to the SVM algorithm to classify the samples into their respective classes based on the detected abnormal plant leaf areas. This solution (Shrivastava and Pradhan, 2021) provided superior plant leaf disease categorization results. However, this performance degraded for samples with significant color and light changes.

Due to the empowerment of DL frameworks and their ability to better deal with image transformations, researchers are now employing them for recognizing plant diseases.

The framework in Argüeso et al. (2020) used the DL technique named Few-Shot Learning (FSL) for recognizing the affected portions of crops and determining the related category. The InceptionV3 model was applied to capture the keypoints of the input image. The SVM classifier was used to classify the samples using the keypoints, according to the detected disease. The approach described in Argüeso et al. (2020) exhibited robust plant disease classification results but requires extensive data for the model training. Agarwal et al. (2020b) proposed a CNN framework containing 3 convolution layers as the feature extractor module before classification. The framework presented in Agarwal et al. (2020b) was a lightweight solution for the plant leaf disease classification but performance degraded for noisy samples. Another lightweight model was presented in Richey et al. (2020) to be used with cellphones. The ResNet50 approach was used as the end-to-end framework to compute the deep features and perform the classification task. The approach improved the processing complexity for plant disease classification. However, it was not supported by all mobile phones due to the memory requirements. Another framework was depicted in Batool et al. (2020) to classify the numerous types of tomato crop abnormalities. The AlexNet model was employed to extract the deep features of the plant images which were later passed as input to the KNN approach for the classification of the images into their respective category. This work was proficient in recognizing the various categories of tomato plant leaves. However, the KNN algorithm was a time-consuming approach. Similarly, an approach for categorizing the tomato plant leaf abnormalities was described in Karthik et al. (2020) that employed the residual method to compute the reliable feature set. A CNN-based classifier was introduced to categorize the samples based on the learned features of different classes. The approach (Karthik et al., 2020) classified the samples in the related categories better. However, it required a large number of samples for training, which further complicated the model. Dwivedi et al. (2021) applied the object detection approach named region-based CNN (RCNN) to automatically detect and localize the diseased area of grape plant leaves. The approach used the ResNet18 as the feature extractor unit which calculates the keypoints set from the plant images. In the next phase, the RCNN framework applied the region proposals approach to locate the affected portion and determine the associated class. The solution depicted in Dwivedi et al. (2021) worked well in recognizing the various diseases of the grape plant but was unable to generalize well from unseen training data. Another approach was discussed in Akshai and Anitha (2021) where several DL frameworks, namely VGG, DenseNet, and ResNet, were evaluated for the detection and classification of several types of plant leaf diseases. This approach (Akshai and Anitha, 2021) showed better results for the DenseNet model. Albattah et al. (2021) proposed an object detection approach, namely the CenterNet model, for the automated identification and classification of numerous types of plant leaf diseases. Initially, the dense model was used for the extraction of the keypoints set from the input images. These were then used to recognize the diseased portion of plant samples. This approach (Albattah et al., 2021) showed better plant leaf abnormality recognition ability. However, the model needed assessment on a more challenging dataset. Another DL approach was evaluated in Albattah et al. (2022) where the EfficientNetV2 model was tested for the classification of numerous types of plant diseases, that results in improving the classification performance. In Agarwal et al. (2021c), a DL approach, namely the VGG16 model, was used in the classification of tomato leaf diseases. The approach introduced the concept of model optimization, but the detection performance required extensive result improvements. Similarly, other works discussed the model optimization concept for the plant leaf diseases categorization (Agarwal et al., 2021a,b) but the recognition results needed improvement. Zhao et al. (2021) presented a model to recognize numerous tomato plant leaf abnormalities in which the CNN approach, merged with an attention mechanism, was utilized. The methodology attained classification results of 99.24%. Moreover, in Maeda-Gutiérrez et al. (2020), different DL networks, i.e., Inception V3, AlexNet, GoogleNet, ResNet-18, and SE-ResNet50 were tested for tomato plant disease classification. The GoogleNet approach worked well with classification results of 99.39%. Bhujel et al. (2022) also proposed a DL model, namely ResNet18, along with the CBAM for recognizing the tomato plant abnormalities and achieved an accuracy of 99.69%. The methods in Maeda-Gutiérrez et al. (2020), Zhao et al. (2021), and Bhujel et al. (2022) enhanced the tomato plant leaf diseases categorization results. However, these works accomplished classification at the image level and are incapable of identifying the precise diseased area.

A critical investigation of existing work is outlined in Table 1, which depicts that there is a performance gap that requires a more reliable model. This model must be proficient enough to recognize the numerous categories of tomato plant leaf disease and minimize the time complexity. In the presented work, we have tried to cover this gap by proposing a more accurate and robust approach for tomato plant leaf disease classification.

Materials and methods

In this section, an in-depth discussion of the proposed technique for tomato plant leaf disease localization and classification is presented. The basic motivation of this framework is to present an accurate and computationally efficient approach that is empowered to automatically nominate a representative feature vector independent from executing any manual examination. Our work comprises two main steps to accomplish the automated recognition of plant leaf diseases. First, the images from the PlantVillage dataset are employed to develop the annotations to correctly identify the affected portions and their associated classes. Then, these annotations are used in training the DenseNet-77-based CornerNet approach. During the test phase, the images from the test set are used to validate the model's performance. More precisely, we have customized the CornerNet model (Law and Deng, 2019) by introducing the DenseNet-77 network in its feature extraction unit. The DenseNet-77 approach as the base network computes the feature vector which is then passed to the one-stage detector of the CornerNet model to localize and classify the affected regions into 10 classes. Several standard evaluation measures are then used to quantitatively measure the performance of the introduced framework. The detailed model formulation of our framework is given in Algorithm 1, while the pictorial demonstrations showing the detailed steps of our approach are given in Figure 1.

```
 INPUT:
   TS, AI
OUTPUT:
   Bbx, CustomCoNet, C-score
     TS - total no of samples used
     for model training
     AI - annotated images showing the
     diseased area on the tomato plant
     leaves
     Bbx - rectangular box showing the
     diseased region on the output image
     CustomCoNet - CornerNet model with
       the DenseNet-77 backbone
       C-score - confidence score along
       with predicted class
SampleSize ← [x y]
Bbx computation
  β ← AnchorsCalculation (TS, AI)
CustomCustomCoNet-Model
  CustomCoNet ← CornerNetWithDenseNet-77
  (SampleSize, β)
  [$d_r$ $d_t$] ← Distribution of dataset into
  train and test sets
The training module for tomato
plant leaf disease detection and
classfication
  Foreach image m in → $d_r$
    Calculate DenseNet-77-based-
    deepFeatures ← df
  End For

  Train CustomCoNet on df, and measure
  network training time as t_d77
  β _dense ← EstimateDiseasedPos(df)
  V_dense ← Validate_Model
  (DenseNet-77, β_dense)

  Foreach images M in → $d_t$
    (i) Measure features with trained
        model €→V_dense
    (ii) [Bbx, C-score, class] ←
         Predict (€)
    (iii) Present output image with
          Bbox, class
    (iv) η ← [η bbox]
  End For

Ap_€ ← Test framework € using η
Output_class ← CustomCoNet (Ap_€).
```

Algorithm 1. Description of steps followed by the proposed work.

Data preparation for model training

The training of the object detection model was based on annotations development. This was focused on clearly localizing the affected region from the training samples and their associated category. Therefore, in the first step, we have used the images from the training set of the plant samples from the PlantVillage dataset and used the LabelImg software (Lin, 2020) for relevant annotation generation. These annotations assist in exactly outlining the diseased areas of leaves by developing the bounding box (*bbx*) around them. The dimensions of the annotations are saved as an XML file which is later employed for model training. A few examples of annotated samples are given in Figure 2.

TABLE 1 An analysis of existing methods.

Reference	Method	Accuracy (%)	Limitation
Hand-coded approaches			
Le et al., 2020	K-FLBPCM + SVM	98.63	The technique lacks the ability to classify distorted plant images.
Ahmad et al., 2020	DLQP + SVM	97.80	This approach is not efficient for noisy images.
Sun et al., 2019	GLCM + SVM	98.50	The technique entails high computational costs.
Pantazi et al., 2019	LBP + SVM	95	This approach is not efficient for noisy images.
Ramesh et al., 2018	HOGs + RF	70.14	The work requires classification result improvements.
Kuricheti and Supriya, 2019	GLCM + SVM	91	The technique lacks the ability to tackle the intensity and color variations found in the plant images.
Kaur and Education, 2021	SIFT, LBP, GLCM + SVM, KNN, and RF	82.12	The results need further improvements.
Shrivastava and Pradhan, 2021	Color spaces + SVM	94.65.	The approach is not robust for unseen data.
DL approaches			
Argüeso et al., 2020	InceptionV3 + SVM	91.40	The technique needs further assessment over a more complex database.
Agarwal et al., 2020b	CNN	91.20	The framework is facing the network over-fitting problem.
Richey et al., 2020	ResNet50	99	The approach requires high processing power.
Batool et al., 2020	AlexNet + KNN	76.10	The approach takes a long time to process samples.
Karthik et al., 2020	CNN	98	The work needs huge samples to train the network.
Dwivedi et al., 2021	RCNN	99.93	The approach does not perform well for unseen examples.
Akshai and Anitha, 2021	VGG, ResNet, and DenseNet	98.27	The approach requires high processing power.
Albattah et al., 2021	CenterNet	99.90	The framework needs to be evaluated on real-world examples.
Albattah et al., 2022	EfficientNetV2	99.93	Performance degrades for distorted samples.
Agarwal et al., 2021c	VGG16	98.40	The classification accuracy requires improvements.

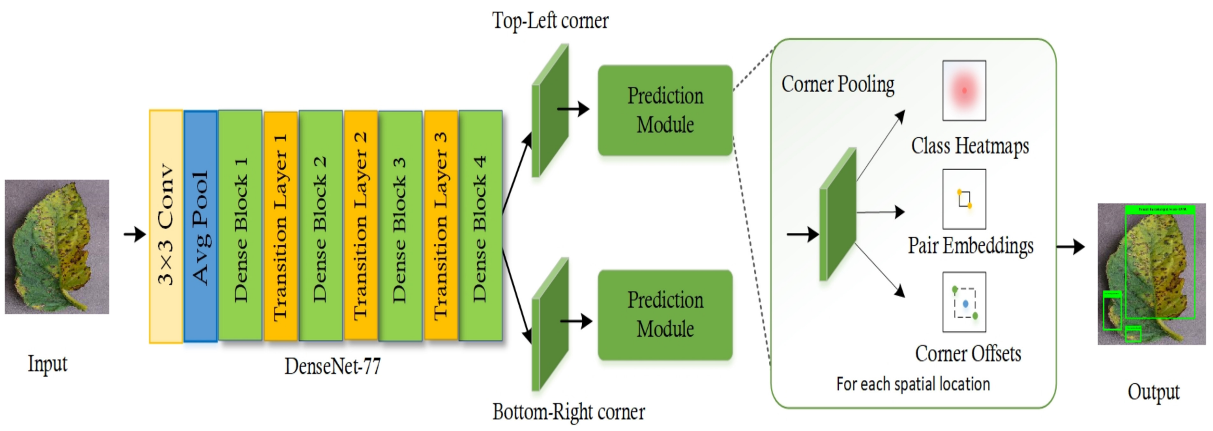


FIGURE 1
Pictorial depiction of the DenseNet-77-based CornerNet model for the tomato plant leaf diseases classification.

CornerNet model

The CornerNet (Law and Deng, 2019) is a well-known one-stage object detection model that recognizes the region of interest (ROI) (the diseased region of the tomato plants in this case) from the input samples through keypoint calculation. The CornerNet model estimates the Top-Left (T_L) and Bottom-Right (B_R) corners to draw the *bbx* with more accuracy when compared to other object detection models (Girshick, 2015; Ren et al., 2016). The CornerNet framework is comprised of two basic units: the feature computation backbone and the prediction module (**Figure 1**). At the start, a keypoints extractor unit is used which extracts the reliable feature vector that is employed to estimate the heatmaps (*Hms*), embeddings,

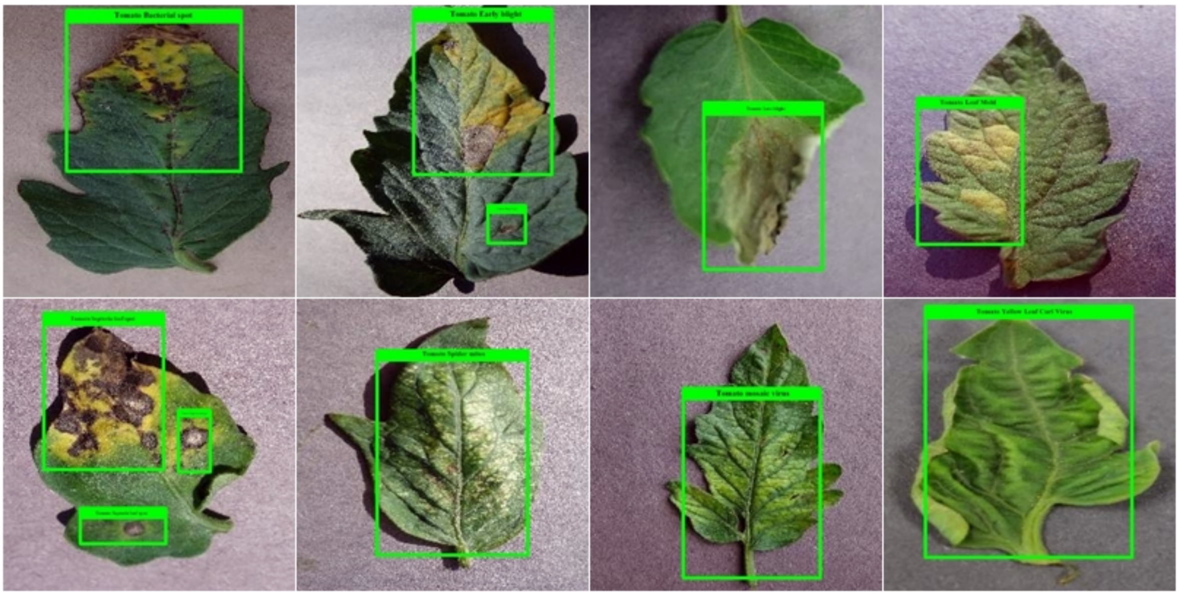


FIGURE 2
Example of annotated images of the tomato plant from the PlantVillage dataset.

offset, and class (*C*). The *Hms* give an approximation of a location in a sample where a T_L/B_R corner is associated with a particular category (Nawaz et al., 2021). The embeddings are used to discriminate the detected pairs of corners and offsets to fine-tune the *bbx* position. The corners with high scored T_L and B_R coordinates are used to determine the exact position of the *bbx*, whereas the associated category for each detected diseased region is specified by using the embedding distances on the computed feature vector.

The CornerNet framework shows robust performance in detecting and classifying several types of objects (Girshick, 2015; Raj et al., 2015; Redmon et al., 2016; Zhao et al., 2016). However, the abnormalities of tomato plant leaves have some distinct characteristics. These include leaves of different shapes and sizes and high color resemblance in the affected and healthy regions of plant leaves which complicates the classification procedure. Moreover, the existence of several image distortions such as differences in the light, color, and brightness of the samples and the incidence of noise and blurring effect further increase the complexity of the tomato plant leaf disease classification process. Therefore, to better tackle the complexities of samples, we have customized the CornerNet model by introducing a more effective feature extractor, namely the DenseNet-77, as its base network. The introduced base network is capable of locating and extracting the more relevant sample attributes which assist the CornerNet approach and enhance its recall ability in comparison to the conventional model.

The reason for selecting the CornerNet approach for classifying the diseases of tomato plants in this study is its capability for effectively detecting objects by utilizing keypoint approximation in comparison to earlier approaches (Girshick, 2015; Girshick et al., 2015; Liu et al., 2016; Ren et al., 2016; Redmon and Farhadi, 2018). The framework utilizes detailed keypoints and identifies the object by employing a one-stage detector. This eliminates the need to use large anchor boxes for diverse target dimensions as used in other one-stage object recognition models, i.e., single-shot detector (SSD) (Liu et al., 2016), and You Only Look Once (YOLO) (v2, v3) (Redmon and Farhadi, 2018). Moreover, the CornerNet model is more computationally robust than the other anchor-based two-stage approaches, i.e., RCNN (Girshick et al., 2015), Fast-RCNN (Girshick, 2015; Nazir et al., 2020), and Faster-RCNN (Ren et al., 2016; Albahli et al., 2021), as these techniques employ two phases to accomplish the object localization and categorization. Consequently, the DenseNet-77-based CornerNet framework efficiently deals with the issues of existing models by presenting a more proficient network that extracts more nominative sample features and reduces the computational cost.

Modified CornerNet framework

The base of a model is responsible for identifying and computing the reliable feature vector that gives the semantic information and reliable location of a target in an image. The affected regions of tomato plant leaves are small, therefore a robust and representative set of keypoints is mandatory to recognize the diseased portion from complex backgrounds such as changing acquisition positions, lightning conditions, noise, and blurring. The conventional CornerNet approach

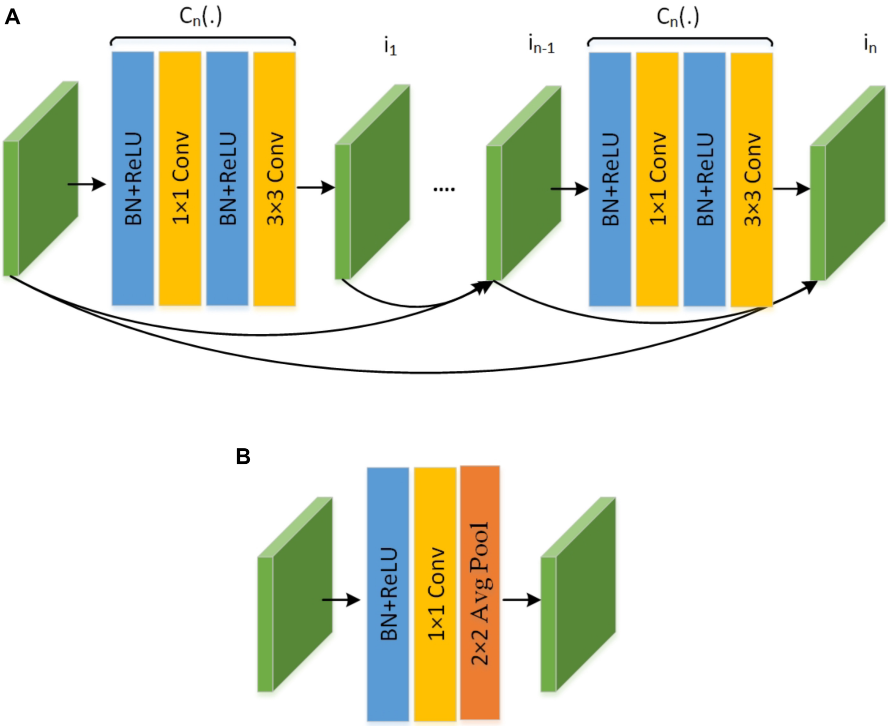


FIGURE 3
The pictorial representation of **(A)** dense block and **(B)** transition block.

was introduced along with the Hourglass104 as the base network (Law and Deng, 2019). The major drawback of the Hourglass104 network is its huge structural complexity. The larger number of framework parameters increases the computational burden on the CornerNet model and slows down the target identification procedure. Further, the Hourglass104 approach is inefficient when computing reliable keypoints for all types of image distortions, e.g., extensive changes in the size, color, and orientation of the affected areas (Zhao et al., 2019). Therefore, we have changed the feature extractor layer of the CornerNet model to enhance the identification and categorization performance for tomato plant leaf diseases. To this end, we have utilized the DenseNet-77 (Huang et al., 2017) as the base network of the CornerNet model in our proposed approach.

DenseNet-100 feature extractor

The DenseNet-77 network is a lightweight model from the DenseNet family and has two major benefits over the conventional DenseNet approach: first, the number of model parameters is smaller than the original DenseNet model (Masood et al., 2021); secondly, the layers within each dense block (D_b) are also reduced to further simplify its structure. The employed DenseNet-77 model is a shallower framework compared to the Hourglass104 approach and comprises four D_bs in total. A detailed demonstration of the architectural representation of the DenseNet-77 is given in **Figure 3**. The DenseNet-77 approach comprises a smaller number of model parameters (6.2M) in comparison to the Hourglass104 base network (187M). Such architectural settings give it a computational advantage over the original base network. In all D_bs, the convolution layers are directly linked and the computed feature maps from starting layers are communicated to the subsequent layers. The DenseNet model encourages the reemployment of the computed features and facilitates the communication of the computed data in the entire network structure. This empowers it to deal with the image distortions effectively (Huang et al., 2017). **Table 2** shows the network depiction of the DenseNet-77 model.

The network consists of numerous Convolutional Layers (Cn_L), D_bs, and Transition Layers (Tn_L). A pictorial depiction of the D_b is given in **Figure 3** and is the fundamental part of the DenseNet framework. In **Figure 3**, i_0 represents the input layer and k_0 depicts the feature maps. Furthermore, $C_n(.)$ is a compound function containing 3 consecutive actions: a 3 × 3 Cn_L filter, Batch Normalization (Bt_N), and ReLU. Each $C_N(.)$ operation produces keypoint maps (k), that are used as input i_N succeeding layers. The employment of all earlier computed features to the next layers introduces the $k \times (t-1)+k_0$ feature maps at the t-th layer of D_b, which increases the feature space immensely. Hence, the Tn_L is used between the D_b to lessen the computed features. The Tn_L is calculated as Bt_N and 1 × 1 Cn_L

and the average pooling layer is represented as Ap_L, as depicted in Figure 3.

Prediction module

The feature computation framework consists of two separate output units that denote the T_L and the B_R corners estimation branches, respectively. Each branch unit comprises a corner pooling layer (CP_L) positioned on the top of the backbone to pool keypoints and produces three results: *Hms*, embeddings, and offsets. The prediction module is an improved residual block (RB) containing two 3 × 3 Cn_L and one 1 × 1 residual network, followed by a CP_L. The CP_L assists the framework to identify the potential corners. The reduced keypoints are used as the input into a 3 × 3 Cn_L-Bt_N layer and then the reverse projection is performed. This improved RB is followed by a 3 × 3 Cn_L which produces *Hms*, embeddings, and offsets. The *Hms* give the approximation of a location in a sample, as a T_L/B_R corner, that is associated with a particular category. The embeddings are used to discriminate between the detected pairs of corners and offsets to fine-tune the *bbx* position. A suspected image can contain more than one affected region, therefore, embeddings assist the model to determine if the predicted corner points belong to a single disease class or different disease classes.

TABLE 2 Description of the DenseNet-77.

Layer	DenseNet-77	
	Size	Stride
CnL1	$7 \times 7\ cn$	2
$Pool_L1$	$3 \times 3_{max\_pooling}$	2
Db1	$\begin{bmatrix} 1 \times 1 & cn \\ 3 \times 3 & cn \end{bmatrix} \times 6$	1
TnL		
Cn_L2	$1 \times 1\ cn$	1
$Pool_L2$	$2 \times 2Ap_L$	2
Db2	$\begin{bmatrix} 1 \times 1 & cn \\ 3 \times 3 & cn \end{bmatrix} \times 12$	1
TnL		
Cn_L3	$1 \times 1\ cn$	1
$Pool_L3$	$2 \times 2_{ApL}$	2
D_b3	$\begin{bmatrix} 1 \times 1 & cn \\ 3 \times 3 & cn \end{bmatrix} \times 12$	1
Tn_L		
Cn_L4	$1 \times 1\ cn$	1
$Pool_L4$	$2 \times 2Ap_L$	2
D_b4	$\begin{bmatrix} 1 \times 1 & cn \\ 3 \times 3 & cn \end{bmatrix} \times 6$	1
Classification_layer	$7 \times 7\ Ap_L$	
	FCL	
	SoftMax	

Detection

The CornerNet model is a deep learning framework that is independent of the selective search and proposal generation techniques. The test image and the associated annotated sample are given as input to the trained model. The improved CornerNet model extracts the corner points for the diseased area of the tomato plants and computes the associated offsets to the *x* and *y* coordinates, the measurements of *bbx,* and the associated class.

Loss function

The employed framework for the detection and classification of tomato leaf disease is an end-to-end learning method that practices multi-task loss during the training to increase its recognition ability and precisely locate affected leaf regions. The total training loss, designated by L_t, is the combination of four different losses, given as:

$$L_t = L_d + \alpha L_{pl} + \beta L_{ps} + \gamma L_{off} \tag{1}$$

Here, the L_d signifies detection loss accountable for corner identification, while L_{pl} denotes the group loss of group corners of the same *bbx*. Moreover, L_{ps} is the corner separation loss used to separate the corners of different *bbx*, and L_{off} is the smooth *L*1 loss designated for offset adjustment. The symbols α, β, and γ are the constants for our approach, with the values of 0.1, 0.1, and 1, respectively. The mathematical description of the L_d is given in Eq. 2.

$$L_d = \frac{-1}{R} \sum_{j=1}^{c} \sum_{u=1}^{h} \sum_{v=1}^{w} \begin{cases} (1-t)^{\emptyset} \log(t) & if\ (g) = 1 \\ (1-g)^{\omega}\, t^{\emptyset} \log(1-t) & otherwise \end{cases} \tag{2}$$

In this equation, *R* is the total number of detected diseased areas in a given image. For a given image, *c*, *h*, and *w* designate its total channels, width, and height. Moreover, t_{juv} indicates the estimated score at a given position (*u*, *v*) for the diseased area of class (*j*) in the suspected sample, and g_{juv} is the related ground-truth value. The ∅ and ω indicates the model hyperparameters that govern the influence of every selected point and have the values of 2 and 4 for our framework, respectively.

In downsampling, the dimension of the output sample is reduced than the actual sample size. The position (*u*, *v*) of the diseased portion in the test sample is plotted to the position $(\frac{u}{N}, \frac{v}{N})$ in the *Hms*, where *N* indicates the downsampling factor. The remapping of *Hms* to the actual sample size introduces precision loss that eventually degrades the IoU performance for small *bbx*. To tackle this problem, the offsets for all locations are computed to fine-tune the corner dimensions as described in Eq. 3.

$$O_i = (\frac{u^i}{N} - \left\lfloor \frac{u^i}{N} \right\rfloor, \frac{v^i}{N} - \left\lfloor \frac{v^i}{N} \right\rfloor) \tag{3}$$

Here, O_i shows calculated offset, while for corner *i*, u^i, and v^i represents the coordinators of *u* and v. Furthermore, the

L_{off}, employs the smooth *L1* method for adjusting the corner positions and is represented as:

$$L_{off} = \frac{1}{M}\sum_{i=1}^{M} SmoothL1Loss(O_i,\ 0'_i) \tag{4}$$

There could be several affected regions on a single image. Therefore, several B_R and T_L corners are nominated. For all corners, the model estimates an embedding vector to decide whether a group of B_R and T_L corners is associated with the same disease class or different disease classes. For this purpose, the CornerNet model uses the "pull and push" losses for framework training and are given as:

$$L_{pl} = \frac{1}{M}\sum_{x=1}^{M}\left[(e_{lx} - e_x)^2 + (e_{rx} - e_x)^2\ \right] \tag{5}$$

$$L_{ps} = \frac{1}{M\ (M-1)}\sum_{x=1}^{M}\ \sum_{y=1,\ y\neq x}^{M} \max[0,\ \Delta - \left|e_x - e_y\right| \tag{6}$$

Here, e_{lx} shows the T_L while the e_{rx} denotes the B_R corners for a diseased region *x* and e_x is the average value of e_{rx} and e_{rx}. The distance value to declare two detected corners belonging to different categories is set as 1, while the value of Δ is also 1 for all experiments.

Results

In this section, we will outline detailed information about the dataset employed for the detection and classification of tomato plant leaf diseases. Moreover, the mathematical description of the used performance measures is also given. Finally, the results of the extensive experiments that have been conducted to show the efficacy of the proposed approach for tomato plant leaf disease recognition will be discussed.

Dataset

We have used the PlantVillage database (Hughes and Salathé, 2015), a large repository accessible online, to evaluate the effectiveness of the model in detecting and classifying tomato leaf diseases. This dataset is comprised of a total of 54,306 images for 14 crop types. As this study is focused on the diseases of the tomato plant, we have utilized the tomato plant samples belonging to 10 different diseases. The main reason to employ the PlantVillage dataset for our work is that its images contain severe alterations in the size, chrominance, and position of the affected leaf regions. Furthermore, the images contain noise, brightness changes, blurring, and color alterations. An in-depth demonstration of the employed dataset is elaborated in **Figure 4** while a few samples are shown in **Figure 5**.

Performance measures

For measuring the performance of the custom CornerNet model in detecting and classifying tomato plant leaf diseases, we have selected several standard metrics such as accuracy, mAP, intersection over union (IOU), precision, and recall. The mathematical description of accuracy and the mAP measure are given in Eqs 7, 8, respectively, while a graphical demonstration of precision, recall, and IOU is given in **Figure 6**.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$mAP := \sum_{i=1}^{T} AP(t_i)/T \tag{8}$$

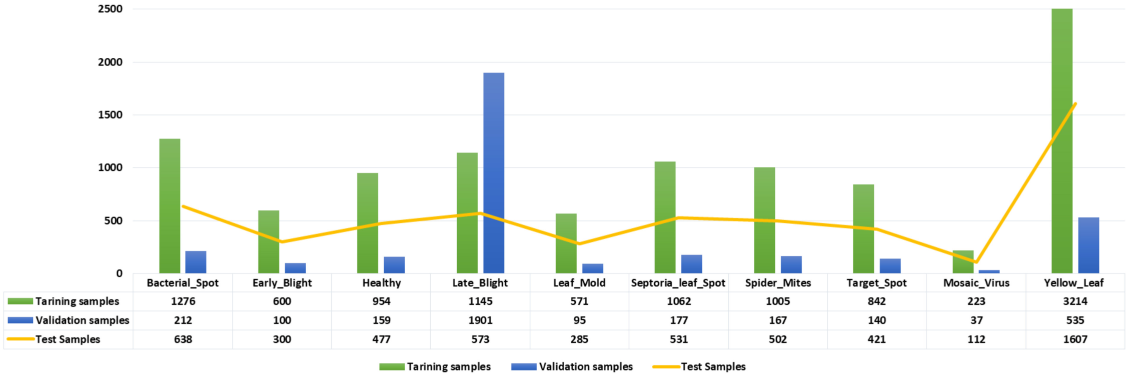
Localization results

The distinguishing attribute of a robust plant leaf disease classification framework is its ability to differentiate among the different classes of disease. To measure this, we designed an experiment. To visually elaborate on the detection performance of the custom CornerNet model, we have depicted the localized samples from the used dataset in **Figure 7**. The samples in **Figure 7** clearly show that our technique is quite efficient in detecting the affected portion of the plant leaves and recognizing the associated classes even under the incidence of color, size, light, chrominance, and brightness changes.

The high recall power of the custom CornerNet model allows it to appropriately identify and categorize the several classes of tomato plant abnormalities. To numerically show the robustness of the proposed solution for tomato plant leaf disease classification, we have used two measures, namely the mAP and IOU score. These are the standard and most heavily employed metrics by the research community for object detection models. The proposed CornerNet model has localized the diseased portion from the plant samples with mAP and IOU scores of 0.984, and 0.979, respectively, which shows the effectiveness of our approach.

Classification performance

An efficient plant leaf disease recognition system must be powerful enough to accurately discriminate among the different types of diseases. We tested the class-wise performance of the presented model with the help of several standard metrics such as precision, recall, accuracy, and F1-score. Initially, we computed the precision and recall values for the custom CornerNet model in locating and classifying the 10 categories of plant leaf abnormalities. We have used the boxplot to show the obtained results as these plots provide a better understanding of



	Bacterial_Spot	Early_Blight	Healthy	Late_Blight	Leaf_Mold	Septoria_leaf_Spot	Spider_Mites	Target_Spot	Mosaic_Virus	Yellow_Leaf
🟩 Tarining samples	1276	600	954	1145	571	1062	1005	842	223	3214
🟦 Validation samples	212	100	159	1901	95	177	167	140	37	535
— Test Samples	638	300	477	573	285	531	502	421	112	1607

FIGURE 4
Details of the tomato plant samples from the PlantVillage dataset.

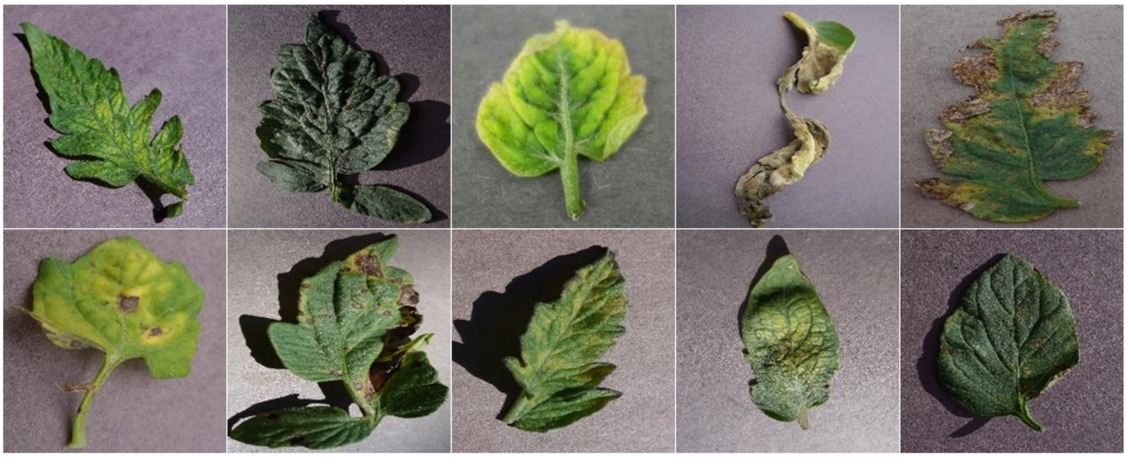


FIGURE 5
An example of tomato plant leaves samples from the PlantVillage dataset.

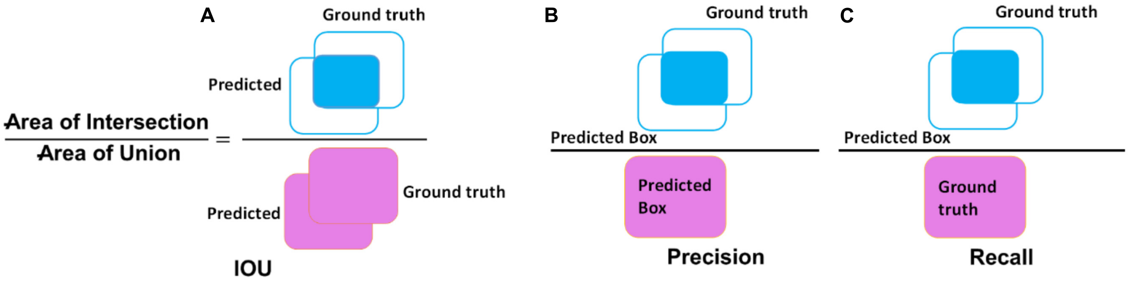


FIGURE 6
Visual demonstration of **(A)** IOU, **(B)** precision, and **(C)** recall.

the results by showing the minimum, maximum, and average values for the employed metrics (Figures 8, 9). The results reported in Figures 8, 9 show that the introduced approach is capable of correctly classifying the 10 classes of tomato plant leaf diseases.

Secondly, we show the calculated F1-score together with the error rate over the employed dataset and acquired values in Figure 10. The custom CornerNet model attains the average F1-score of 99.57% with the maximum and minimum error rates of 0.23 and 0.82%, respectively. The reported values demonstrate

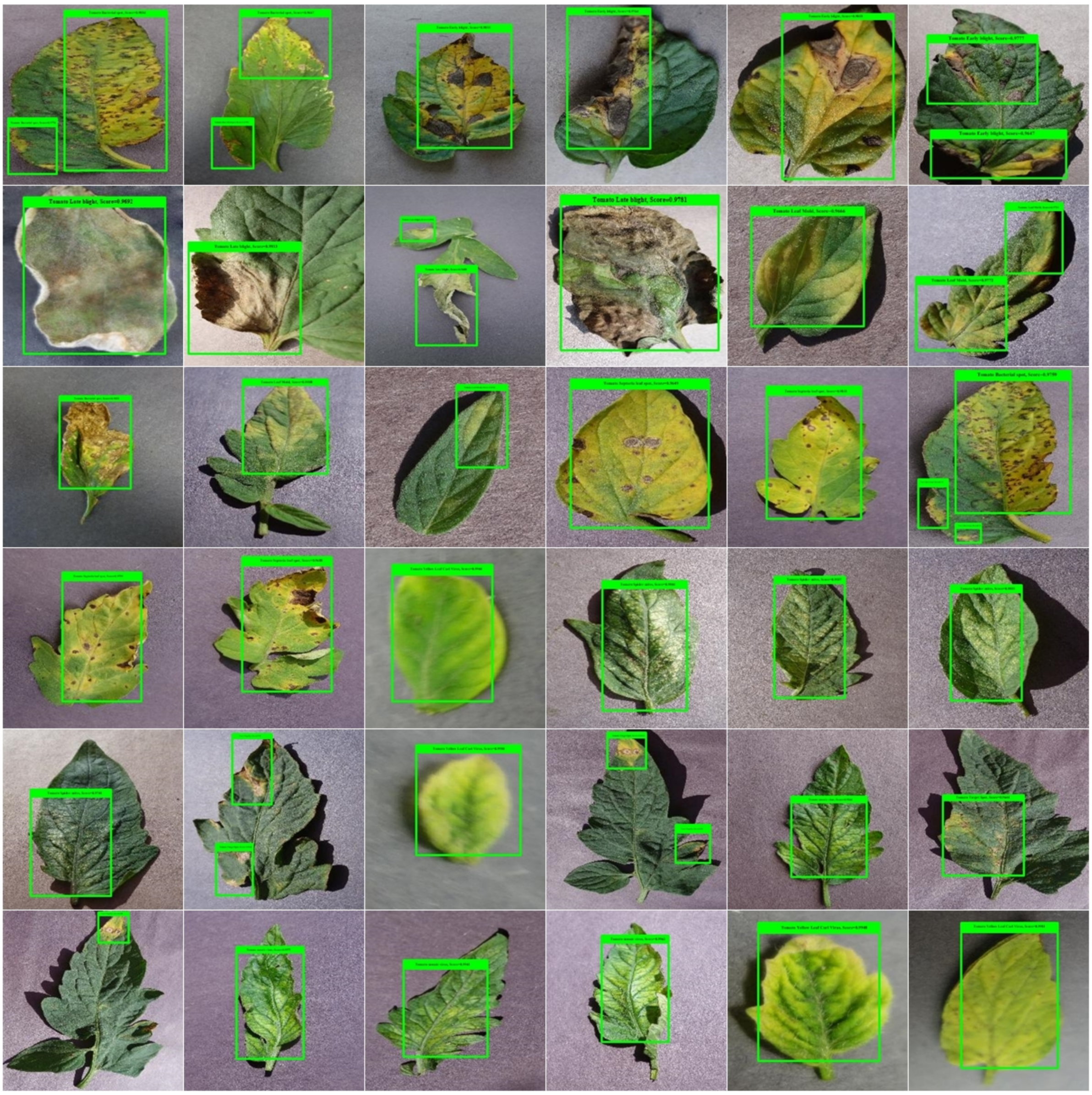

Tomato Early blight, Score=0.9777
Tomato Early blight, Score=0.9647
Tomato Late blight, Score=0.9692
Tomato Late blight, Score=0.9781
Tomato Leaf Mold, Score=0.9666

FIGURE 7
A pictorial depiction of the localized tomato plant leaf diseases samples.

the robustness of the custom CornerNet model in locating and classifying all classes of tomato leaf disease efficiently.

Additionally, we have measured the class-wise accuracy values of the proposed technique and the acquired results are demonstrated in Figure 11. The introduced DenseNet-77-based CornerNet model attains the accuracy values of % for the 10 disease categories of the tomato plant and confirms the effectiveness of our approach.

To further validate the class-wise accurateness of the introduced approach for distinguishing the numerous categories of plant leaf disease, we have created a confusion matrix (Figure 12). This plot can show the actual and estimated classes recognized by a model. The values shown in figure demonstrate that the custom CornerNet model is proficient at recognizing all classes of tomato plant leaf diseases due to its higher recall rate which empowered it to differentiate all categories reliably.

Comparison with base approaches

In this section, we outline an experiment to compare the tomato plant leaf disease recognition capability of the improved CornerNet model against the base networks. We chose

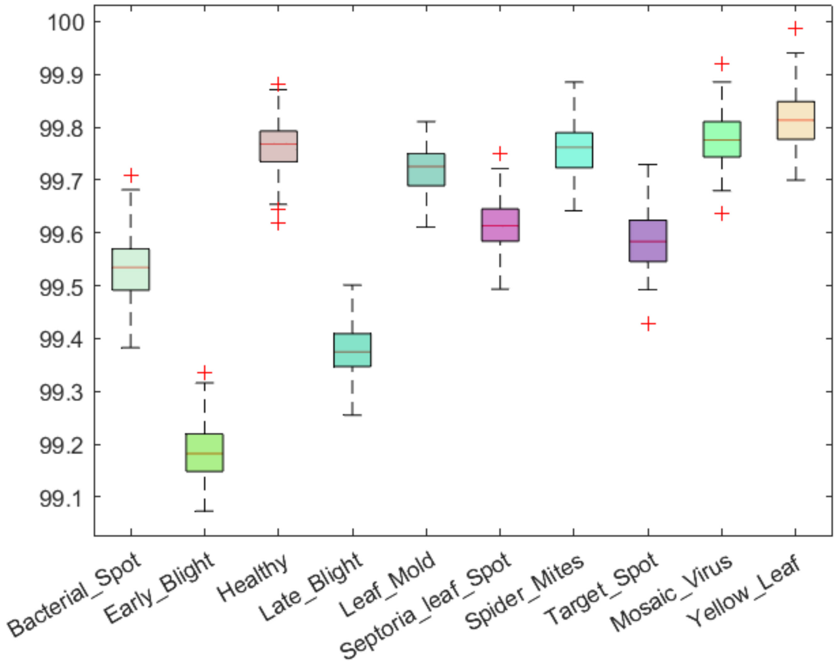


FIGURE 8
A pictorial depiction of the class-wise precision values obtained for the DenseNet-77-based CornerNet model.

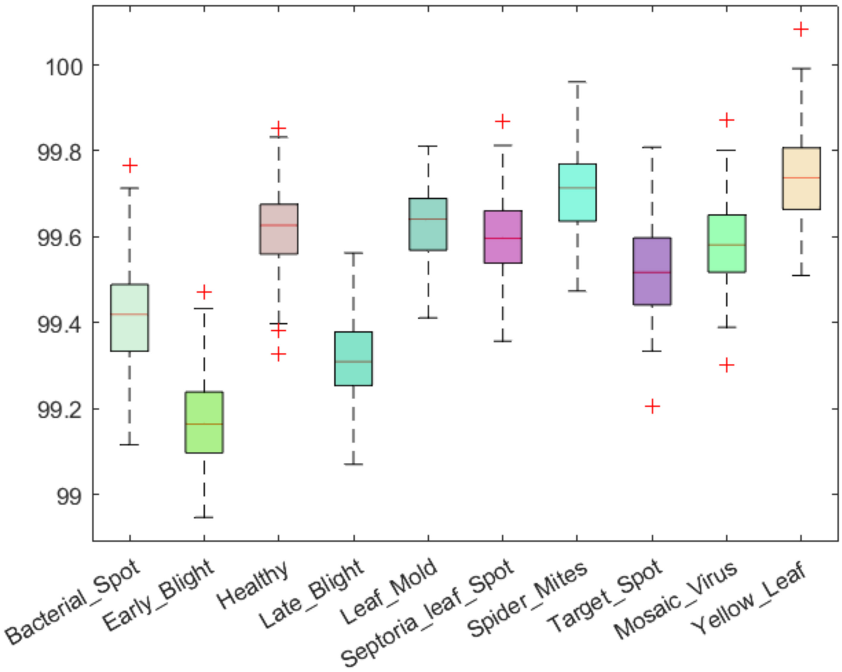


FIGURE 9
A pictorial depiction of the class-wise recall values obtained for the DenseNet-77-based CornerNet model.

several well-known DL frameworks, i.e., GoogleNet, ResNet-101, Xception, VGG-19, and SE-ResNet50. The comparison is depicted in Table 3. The performance analysis shown in Table 3 illustrates that our technique is more accurate than the peer approaches. The DenseNet-77-based CornerNet model attains the highest results for the precision, recall, F1-score,

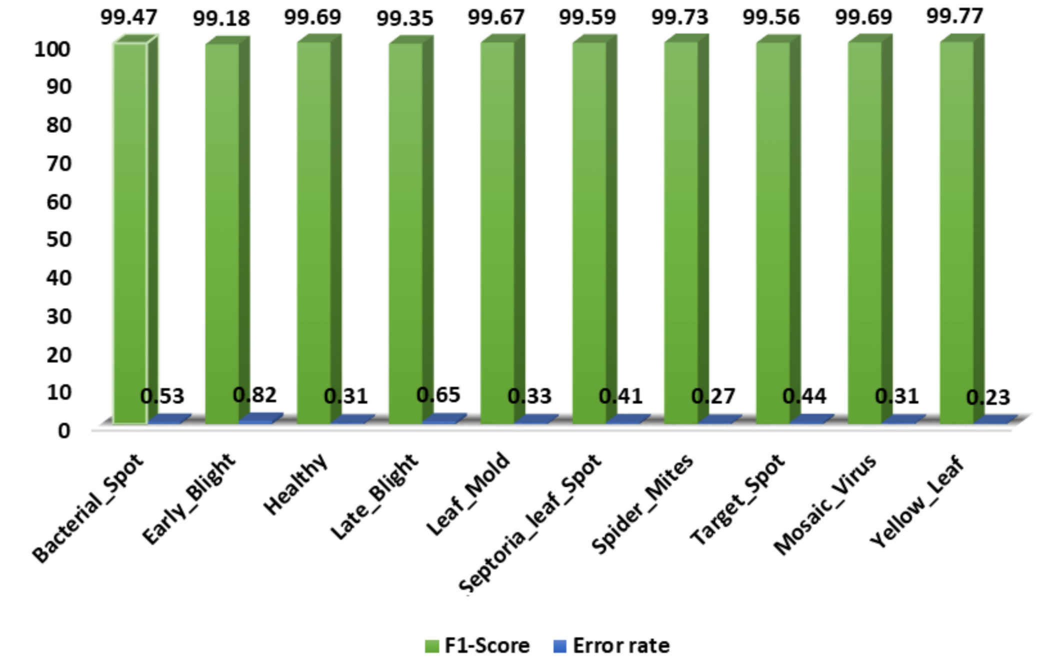


FIGURE 10
A pictorial depiction of the class-wise F1-score values obtained for the DenseNet-77-based CornerNet model.

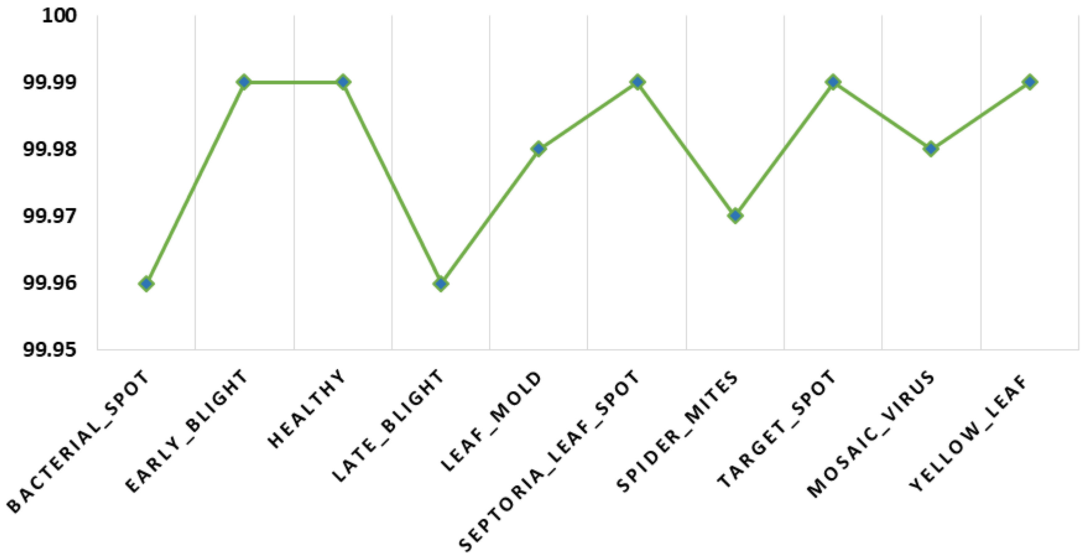
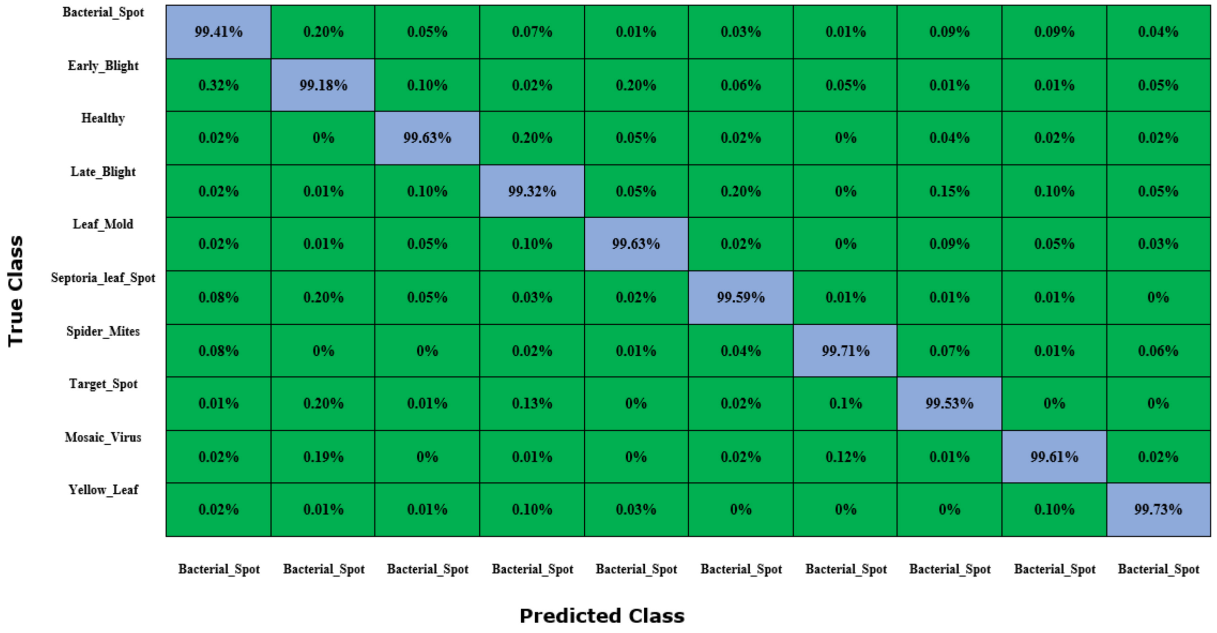


FIGURE 11
A pictorial depiction of the class-wise accuracy values obtained for the DenseNet-77-based CornerNet model.

and accuracy measures with the numeric count of 0.9962, 0.9953, 0.9957, and 99.98%, respectively. The second-highest results are reported by the SE-ResNet50 model with 0.9677, 0.9681, 0.9679, and 96.81% for the precision, recall, F1-score, and accuracy metrics, respectively. Moreover, the GoogleNet model attains the lowest results in classifying the leaf diseases of the tomato plant and attains the scores for precision, recall, F1-score, and accuracy measures of 0.8716, 0.8709, 0.8712, and 87.27%, respectively. The second-lowest values are attained by the Xception model with the numeric stats of 0.8825, 0.8814, 0.8819, and 88.16%. The comparison illustrates the effectiveness of our approach. Specifically, for the precision measurement, the selected methods have an average value of 0.9050, while the DenseNet-77-based CornerNet model acquires the value of



	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot	Bacterial_Spot
Bacterial_Spot	99.41%	0.20%	0.05%	0.07%	0.01%	0.03%	0.01%	0.09%	0.09%	0.04%
Early_Blight	0.32%	99.18%	0.10%	0.02%	0.20%	0.06%	0.05%	0.01%	0.01%	0.05%
Healthy	0.02%	0%	99.63%	0.20%	0.05%	0.02%	0%	0.04%	0.02%	0.02%
Late_Blight	0.02%	0.01%	0.10%	99.32%	0.05%	0.20%	0%	0.15%	0.10%	0.05%
Leaf_Mold	0.02%	0.01%	0.05%	0.10%	99.63%	0.02%	0%	0.09%	0.05%	0.03%
Septoria_leaf_Spot	0.08%	0.20%	0.05%	0.03%	0.02%	99.59%	0.01%	0.01%	0.01%	0%
Spider_Mites	0.08%	0%	0%	0.02%	0.01%	0.04%	99.71%	0.07%	0.01%	0.06%
Target_Spot	0.01%	0.20%	0.01%	0.13%	0%	0.02%	0.1%	99.53%	0%	0%
Mosaic_Virus	0.02%	0.19%	0%	0.01%	0%	0.02%	0.12%	0.01%	99.61%	0.02%
Yellow_Leaf	0.02%	0.01%	0.01%	0.10%	0.03%	0%	0%	0%	0.10%	99.73%

FIGURE 12
Confusion matrix results for tomato plant leaf diseases classification obtained using the DenseNet-77-based CornerNet model.

0.9962 and shows a performance gain of 9.12%. For the recall and F1-score, the selected models have attained the average numeric score of 0.9053 and 0.9091, while in comparative analysis the presented solution has shown the average recall and F1-score of 0.9953 and 0.9957, respectively. Therefore, we can demonstrate average performance gains for the recall and F1-score of 9 and 8.66%, respectively. Moreover, in terms of accuracy, the base models attain an average value of 90.56%. The proposed model attains 99.98% accuracy, representing a performance gain of 9.42%. Furthermore, we outline the time taken for each model. It should be noted that the proposed approach shows the minimum test time. The values show the efficacy of our work to better recognize the several classes of tomato plant leaf abnormalities. The basic cause of this better classification performance of the proposed improved CornerNet model is the employment of the DenseNet-77 model as the keypoints extractor. This uplifts the model to better select the image information to identify the affected areas of the plant leaves and better recognize the associated class.

Performance evaluation with object detection approaches

We have employed an object detection-based model for the localization and classification of the tomato plant leaf diseases and compared the performance of the proposed approach with other object detection techniques. The major reason for performing this simulation was to verify the reliability of the proposed DenseNet-77-based CornerNet model against other competitor techniques while locating the diseased areas from the tomato plant leaves under the occurrence of noise, light alteration, color changes, size variations, etc.

To execute this analysis, we have chosen numerous well-known object detection approaches, namely the Fast-RCNN (Girshick, 2015), Faster-RCNN (Ren et al., 2016) YOLO (Redmon and Farhadi, 2018), the SSD (Liu et al., 2016), and CornerNet (Law and Deng, 2019) models. To measure the performance of the model, the mAP metric is used as it is the standard evaluation measure used by the researchers to assess the classification performance of the object detection techniques. Furthermore, we have compared the test time of models as well to evaluate the time complexities of the comparative approaches as well. The comparison shows the efficiency and effectiveness of our approach and is illustrated in Table 4. The results in Table 4 show that the proposed approach has the highest mAP score and lowest test time with a numeric score of 0.984 and 0.22 s, respectively. The second highest mAP score is the Faster-RCNN model with a numeric count of 0.884. However, it is computationally inefficient and shows a time complexity of 0.28 s due to its two-stage classification network architecture. The SSD model has the lowest mAP score of 0.883 and a test time of 0.27 s. Furthermore, this approach does not perform well for very small plant leaf sizes. The conventional CornerNet model also has less promising results with a mAP score of 0.883 and a test time of 0.25 s. Whereas, the DenseNet-77-based CornerNet approach better tackles the issues of existing object detection approaches for identifying and

TABLE 3 Comparison with other DL frameworks.

Model	Precision	Recall	F1-score	Accuracy (%)	Time (second)
GoogleNet	0.8716	0.8709	0.8712	87.27	0.65
ResNet-101	0.8995	0.9013	0.9004	90.13	1.21
Xception	0.8825	0.8814	0.8819	88.16	0.77
VGG-19	0.9039	0.9047	0.9243	90.42	1.56
SE-ResNet50	0.9677	0.9681	0.9679	96.81	0.57
Proposed	0.9962	0.9953	0.9957	99.98	0.22

classifying the numerous categories of the tomato plant leaves and shows the highest results. The comparison object detection approaches have an average mAP value of 0.859, compared to 0.984 for the proposed algorithm. Therefore, we have attained an average performance gain of 12.42% for the mAP metric. The one-stage detection ability of the proposed approach reduces the network structure complexity which, in turn, gives it a computational advantage.

Model evaluation with the state-of-the-art methods

In this section, we have selected several new approaches (Tm et al., 2018; Kaur and Bhatia, 2019; Agarwal et al., 2020a) that worked for tomato plant leaf disease classification and have used analysis to compare the performance of the improved CornerNet model with them. For this purpose, we have utilized three standard measures: precision, recall, and accuracy. Agarwal et al. (2020a) proposed the EfficientNet model for the automated detection and classification of tomato plant leaf diseases and attained an average accuracy value of 91.20%. Tm et al. (2018) proposed a CNN framework for categorizing the affected area of plant leaves and demonstrated an accuracy value of 94%. Similarly, Kaur and Bhatia (2019) employed a deep learning framework for recognizing the 10 types of plant leaf diseases with an accuracy rate of 98.80%. Hence, the comparative analysis is depicted in **Table 5** and illustrates that our work has attained the highest results for all selected performance measures. From **Table 5**, it can be viewed that the techniques in Tm et al. (2018), Kaur and Bhatia (2019), and Agarwal et al. (2020a) achieve the precision of 0.90, 0.9481, and 0.9880, respectively, whereas the introduced improved CornerNet model obtains the precision of 0.9962. This is the highest of all the reported numeric scores for the selected works. The improved CornerNet model gains the largest value of 0.9953 for the recall performance measure, while the approaches in Tm et al. (2018), Kaur and Bhatia (2019), and Agarwal et al. (2020a) have recall scores of 0.92, 0.9478, and 0.9880, respectively. Moreover, with regards to accuracy, the proposed approach gains the numeric score of 99.98% while the approaches in Tm et al. (2018), Kaur and Bhatia (2019), and Agarwal et al. (2020a) have accuracy values of 91.20, 94, and 98.80%, respectively. The peer works (Tm et al., 2018; Kaur and Bhatia, 2019; Agarwal et al., 2020a) have the average precision, recall, and accuracy values of 0.9453, 0.9519, and 94.67%, respectively, as opposed to 0.9962, 0.9953, and 99.97%, respectively, for the presented work. Therefore, the DenseNet-77-based CornerNet model provides performance gains of 5.08, 4.34, and 5.30% for the precision, recall, and accuracy evaluation measures.

TABLE 4 Comparison with other object detection methods.

Models	mAP	Test time
Fast-RCNN	0.860	0.28
Faster-RCNN	0.884	0.28
YOLOv3	0.842	0.26
SSD	0.830	0.27
Hourglass-based-CornerNet	0.883	0.25
Proposed DenseNet-77-based CornerNet	0.984	0.22

The reason for the competent classification results of the improved CornerNet model is that the techniques in Tm et al. (2018), Kaur and Bhatia (2019), and Agarwal et al. (2020a) are quite complex in network structure. This creates a framework over-fitting problem. The proposed solution is quite simple in structure and the employment of DenseNet-77 as the base network further empowered the CornerNet model to nominate a more reliable set of the sample feature vector. Such a model setting enhances its recognition ability by eliminating redundant information and reducing the model complexity. Further, the one-stage detection and classification ability of the CornerNet model prevents the framework from over-fitting issues and enables it to robustly deal with several image distortions like color, size, brightness, light variation, etc.

TABLE 5 Comparison with the latest studies.

Approach	Precision	Recall	Accuracy (%)
Agarwal et al., 2020a	0.90	0.92	91.20
Tm et al., 2018	0.9481	0.9478	94
Kaur and Bhatia, 2019	0.9880	0.9880	98.80
Proposed	0.9962	0. 9953	99.97

Conclusion

The manual screening of tomato plant leaf diseases relies highly on domain experts to detect the detailed information from the samples under observation. AI-based solutions are trying to fill this gap by automating the manual screening system. However, excessive changes in the mass, color, and size of plant leaves, and the incidence of noise, blurring, and brightness variations in the images complicate the classification task. In this work, we have attempted to overcome the existing issues by proposing a deep learning-based approach namely the DenseNet-77-based CornerNet model. We have carried out extensive experimentations on a standard dataset, namely the PlantVillage, and have confirmed through both the visual and numeric computations that the proposed approach is both efficient and effective in recognizing tomato plant leaf disease. Furthermore, the proposed approach is capable of efficiently detecting the diseased area of the plant leaves from the distorted samples containing several image transformations. However, the approach shows small detection degradation for images with huge angular variations which will be a major focus of our future work. Moreover, we plan to test the proposed model on other plant diseases and evaluate other DL-based frameworks.

Data availability statement

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

Author contributions

SA: conceptualization, methodology, validation, software, supervision, and writing—reviewing and editing. MN: data curation, coding, validation, and writing—original draft preparation. Both authors contributed to the article and approved the submitted version.

Acknowledgments

We would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

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

Agarwal, M., Gupta, S. K., and Biswas, K. (2020a). Development of Efficient CNN model for Tomato crop disease identification. *Sustain. Comput. Inform. Syst.* 28:100407. doi: 10.1016/j.suscom.2020.100407

Agarwal, M., Gupta, S. K., and Biswas, K. (2021b). "A compressed and accelerated SegNet for plant leaf disease segmentation: a differential evolution based approach," in *Pacific-Asia Conference On Knowledge Discovery And Data Mining*, (Berlin: Springer). doi: 10.1007/978-3-030-75768-7_22

Agarwal, M., Gupta, S. K., and Biswas, K. (2021a). "Plant leaf disease segmentation using compressed UNet architecture," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, (Berlin: Springer). doi: 10.1007/978-3-030-75015-2_2

Agarwal, M., Gupta, S. K., Biswas, M., and Garg, D. (2022). Compression and acceleration of convolution neural network: A Genetic Algorithm based approach. *J. Ambient Intell. Humaniz. Comput.* 167, 1–11. doi: 10.1007/s12652-022-03793-1

Agarwal, M., Gupta, S. K., Garg, D., and Khan, M. M. (2021c). "A Partcle Swarm Optimization Based Approach for Filter Pruning in Convolution Neural Network for Tomato Leaf Disease Classification," in *International Advanced Computing Conference*, (Berlin: Springer). doi: 10.1007/978-3-030-95502-1_49

Agarwal, M., Gupta, S. K., Garg, D., and Singh, D. (2021d). "A Novel Compressed and Accelerated Convolution Neural Network for COVID-19 Disease Classification: A Genetic Algorithm Based Approach," in *International Advanced Computing Conference*, (Berlin: Springer). doi: 10.1007/978-3-030-95502-1_8

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., and Gupta, S. (2020b). ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Comput. Sci.* 167, 293–301. doi: 10.1016/j.procs.2020.03.225

Ahmad, W., Shah, S., and Irtaza, A. (2020). Plants disease phenotyping using quinary patterns as texture descriptor. *KSII Trans. Internet Inform. Syst.* 14, 3312–3327. doi: 10.3837/tiis.2020.08.009

Akshai, K., and Anitha, J. (2021). "Plant disease classification using deep learning," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, (Manhattan, NY: IEEE).

Albahli, S., Nawaz, M., Javed, A., and Irtaza, A. (2021). An improved faster-RCNN model for handwritten character recognition. *Arab. J. Sci. Eng.* 46, 8509–8523. doi: 10.1007/s13369-021-05471-4

Albattah, W., Javed, A., Nawaz, M., Masood, M., and Albahli, S. (2022). Artificial intelligence-based drone system for multiclass plant disease detection using an improved efficient convolutional neural network. *Front. Plant Sci.* 13.

Albattah, W., Nawaz, M., Javed, A., Masood, M., and Albahli, S. (2021). A novel deep learning method for detection and classification of plant diseases. *Complex Intell. Syst.* 8, 507–524. doi: 10.1007/s40747-021-00536-1

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., et al. (2020). Few-Shot Learning approach for plant disease classification using images taken in the field. *Comput. Electron. Agric.* 175:105542. doi: 10.1016/j.compag.2020.105542

Batool, A., Hyder, S. B., Rahim, A., Waheed, N., and Asghar, M. A. (2020). "Classification and Identification of Tomato Leaf Disease Using Deep Neural Network," in *2020 International Conference on Engineering and Emerging Technologies (ICEET)*, (Bellingham, WA: IEEE). doi: 10.1109/ICEET48479.2020.9048207

Bello-Cerezo, R., Bianconi, F., Maria, F. Di, Napoletano, P., and Smeraldi, F. (2019). Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions. *Appl. Sci.* 9:738. doi: 10.3390/app9040738

Bhujel, A., Kim, N.-E., Arulmozhi, E., Basak, J. K., and Kim, H.-T. (2022). A lightweight Attention-based convolutional neural networks for tomato leaf disease classification. *Agriculture* 12:228. doi: 10.3390/agriculture12020228

Bruinsma, J. (2009). *The resource outlook to 2050: By how much do land, water and crop yields need to increase by 2050: Expert meeting on how to feed the world in 2009*. Rome: Food and Agriculture Organization of the United Nations.

Chowdhury, M. E., Rahman, T., Khandakar, A., Ayari, M. A., Khan, A. U., Khan, M. S., et al. (2021). Automatic and reliable leaf disease detection using deep learning techniques. *AgriEngineering* 3, 294–312. doi: 10.3390/agriengineering3020020

Dinh, H. X., Singh, D., Periyannan, S., Park, R. F., and Pourkheirandish, M. (2020). Molecular genetics of leaf rust resistance in wheat and barley. *Theor. Appl. Genet.* 133, 2035–2050. doi: 10.1007/s00122-020-03570-8

Dwivedi, R., Dey, S., Chakraborty, C., and Tiwari, S. (2021). Grape disease detection network based on multi-task learning and attention features. *IEEE Sen. J.* 21, 17573–17580. doi: 10.1109/JSEN.2021.3064060

Elnaggar, S., Mohamed, A. M., Bakeer, A., and Osman, T. A. (2018). Current status of bacterial wilt (*Ralstonia solanacearum*) disease in major tomato (Solanum lycopersicum L.) growing areas in Egypt. *Arch. Agric. Environ. Sci.* 3, 399–406. doi: 10.26832/24566632.2018.0304012

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

Gebbers, R., and Adamchuk, V. I. (2010). Precision agriculture and food security. *Science* 327, 828–831. doi: 10.1126/science.1183899

Gewali, U. B., Monteiro, S. T., and Saber, E. (2018). Machine learning based hyperspectral image analysis: A survey. *arXiv* [Preprint].

Girshick, R. (2015). "Fast R-CNN," in *Proceedings Of The Ieee International Conference On Computer Vision*, (Santiago: IEEE). doi: 10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 142–158. doi: 10.1109/TPAMI.2015.2437384

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*, (Honolulu, HI: IEEE) doi: 10.1109/CVPR.2017.243

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].

Joachims, T. (1998). *Making Large-Scale SVM Learning Practical Technical Report*. Dortmund: Technical University Dortmund.

Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., and Menaka, R. (2020). Attention embedded residual CNN for disease detection in tomato leaves. *Appl. Soft Comput.* 86:105933. doi: 10.1016/j.asoc.2019.105933

Kaur, M., and Bhatia, R. (2019). "Development of an improved tomato leaf disease detection and classification method," in *2019 IEEE Conference on Information and Communication Technology*, (Manhattan, NY: IEEE). doi: 10.1109/CICT48419.2019.9066230

Kaur, N. J. T. J. O. C., and Education, M. (2021). Plant leaf disease detection using ensemble classification and feature extraction. *Turkish J. Comput. Math. Educ.* 12, 2339–2352.

Kuricheti, G., and Supriya, P. (2019). "Computer Vision Based Turmeric Leaf Disease Detection and Classification: A Step to Smart Agriculture," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, (Manhattan, NY: IEEE). doi: 10.1109/ICOEI.2019.8862706

Law, H., and Deng, J. (2019). CornerNet: Detecting objects as paired keypoints. *Int. J. Comput. Vis.* 128, 642–656. doi: 10.1007/s11263-019-01204-1

Le, V. N. T., Ahderom, S., Apopei, B., and Alameh, K. (2020). A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators. *GigaScience* 9:giaa017. doi: 10.1093/gigascience/giaa017

Lin, T. (2020). *Labelimg*. Available online at: https://github.com/tzutalin/ImageNet_Utils.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "Ssd: Single shot multibox detector," in *European Conference On Computer Vision*, (Berlin: Springer). doi: 10.1007/978-3-319-46448-0_2

Maeda-Gutiérrez, V., Galvan-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

Masood, M., Nazir, T., Nawaz, M., Mehmood, A., Rashid, J., Kwon, H.-Y., et al. (2021). A novel deep learning method for recognition and classification of brain tumors from MRI images. *Diagnostics* 11:744. doi: 10.3390/diagnostics11050744

Nawaz, M., Nazir, T., Masood, M., Mehmood, A., Mahum, R., Khan, M. A., et al. (2021). Analysis of brain MRI images using improved cornernet approach. *Diagnostics* 11:1856. doi: 10.3390/diagnostics11101856

Nazir, T., Irtaza, A., Javed, A., Malik, H., Hussain, D., and Naqvi, R. A. (2020). Retinal image analysis for diabetes-based eye disease detection using deep learning. *Appl. Sci.* 10:6185. doi: 10.3390/app10186185

Pantazi, X. E., Moshou, D., and Tamouridou, A. A. (2019). Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. *Comput. Electron. Agric.* 156, 96–104. doi: 10.1016/j.compag.2018.11.005

Patil, S., and Chandavale, A. (2015). A survey on methods of plant disease detection. *Int. J. Sci. Res.* 4, 1392–1396.

Paul, A., Ghosh, S., Das, A. K., Goswami, S., Choudhury, S. D., and Sen, S. (2020). "A review on agricultural advancement based on computer vision and machine learning," in *Emerging Technology In Modelling And Graphics*, eds J. Mandal and D. Bhattacharya (Berlin: Springer), 567–581. doi: 10.1007/978-981-13-7403-6_50

Raj, A., Namboodiri, V. P., and Tuytelaars, T. (2015). Subspace alignment based domain adaptation for rcnn detector. *arXiv* [Preprint]. doi: 10.5244/C.29.166

Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., and Vinod, P. (2018). "Plant disease detection using machine learning," in *2018 International Conference On Design Innovations For 3cs Compute Communicate Control (ICDI3C)*, (Manhattan, NY: IEEE). doi: 10.1109/ICDI3C.2018.00017

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

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*, (Las Vegas, NV: IEEE) doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2016). 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

Richey, B., Majumder, S., Shirvaikar, M., and Kehtarnavaz, N. (2020). "Real-time detection of maize crop disease via a deep learning-based smartphone app," in *Proceedings of the Real-Time Image Processing and Deep Learning 2020*, (Bellingham, WA: International Society for Optics and Photonics). doi: 10.1117/12.2557317

Rokach, L., and Maimon, O. (2005). "Decision trees," in *Data Mining And Knowledge Discovery Handbook*, eds O. Maimon and L. Rokach (Berlin: Springer), 165–192. doi: 10.1007/0-387-25465-X_9

Roska, T., and Chua, L. O. (1993). The CNN universal machine: An analogic array computer. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* 40, 163–173. doi: 10.1109/82.222815

Salakhutdinov, R., and Hinton, G. (2009). "Deep Boltzmann machines," in *Proceedings of the artificial intelligence and statistics (PMLR)*, Birmingham.

Sankaran, S., Mishra, A., Ehsani, R., and Davis, C. (2010). A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* 72, 1–13. doi: 10.1016/j.compag.2010.02.007

Sardogan, M., Tuncer, A., and Ozen, Y. (2018). "Plant leaf disease detection and classification based on CNN with LVQ algorithm," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, (Manhattan, NY: IEEE). doi: 10.1109/UBMK.2018.8566635

Shrivastava, V. K., and Pradhan, M. K. (2021). Rice plant disease classification using color features: A machine learning paradigm. *J. Plant Pathol.* 103, 17–26. doi: 10.1007/s42161-020-00683-3

Sun, Y., Jiang, Z., Zhang, L., Dong, W., and Rao, Y. (2019). SLIC_SVM based leaf diseases saliency map extraction of tea plant. *Comput. Electron. Agric.* 157, 102–109. doi: 10.1016/j.compag.2018.12.042

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*, (Berlin: IEEE). doi: 10.1109/CVPR.2015.7298594

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

Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., and Koolagudi, S. G. (2018). "Tomato leaf disease detection using convolutional neural networks," in *2018 Eleventh International Conference On Contemporary Computing (IC3)*, (Manhattan, NY: IEEE). doi: 10.1109/IC3.2018.8530532

Valenzuela, M. E. M., and Restović, F. (2019). "Valorization of Tomato Waste for Energy Production," in *Tomato Chemistry, Industrial Processing and Product Development*, ed. S. Porretta (London: Royal Society of Chemistry), 245–258. doi: 10.1039/9781788016247-00245

Vedaldi, A., and Zisserman, A. (2016). Vgg convolutional neural networks practical. *Dep. Eng. Sci. Univ. Oxford* 2016:66.

Wolfenson, K. D. M. (2013). *Coping With The Food And Agriculture Challenge: Smallholders' Agenda*. Rome: Food Agriculture Organisation of the United Nations.

Yuan, Z.-W., and Zhang, J. (2016). "Feature extraction and image retrieval based on AlexNet," in *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, (Bellingham, WA: International Society for Optics and Photonics). doi: 10.1117/12.2243849

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv* [Preprint].

Zhao, S., Peng, Y., Liu, J., and Wu, S. (2021). Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture* 11:651.

Zhao, X., Li, W., Zhang, Y., Gulliver, T. A., Chang, S., and Feng, Z. (2016). "A faster RCNN-based pedestrian detection system," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, (Manhattan, NY: IEEE). doi: 10.1109/VTCFall.2016.7880852

Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865

TYPE Original Research
PUBLISHED 15 September 2022
DOI 10.3389/fpls.2022.989086

Detection of unknown strawberry diseases based on OpenMatch and two-head network for continual learning

Kan Jiang[1], Jie You[1], Ulzii-Orshikh Dorj[1], Hyongsuk Kim[2,3] and Joonwhoan Lee[1]*

[1]Department of Computer Science and Engineering, Artificial Intelligence Lab, Jeonbuk National University, Jeonju, South Korea, [2]Division of Electronics and Information Engineering, Jeonbuk National University, Jeonju, South Korea, [3]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, South Korea

Check for updates

OPEN ACCESS

EDITED BY
Peng Chen,
Anhui University, China

REVIEWED BY
Yaqoob Majeed,
University of Agriculture, Faisalabad, Pakistan
Sijia Yu,
Rutgers, The State University of New Jersey - Busch Campus, United States

*CORRESPONDENCE
Joonwhoan Lee
chlee@jbnu.ac.kr

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

RECEIVED 08 July 2022
ACCEPTED 22 August 2022
PUBLISHED 15 September 2022

CITATION
Jiang K, You J, Dorj U-O, Kim H and Lee J (2022) Detection of unknown strawberry diseases based on OpenMatch and two-head network for continual learning. *Front. Plant Sci.* 13:989086. doi: 10.3389/fpls.2022.989086

For continual learning in the process of plant disease recognition it is necessary to first distinguish between unknown diseases from those of known diseases. This paper deals with two different but related deep learning techniques for the detection of unknown plant diseases; Open Set Recognition (OSR) and Out-of-Distribution (OoD) detection. Despite the significant progress in OSR, it is still premature to apply it to fine-grained recognition tasks without outlier exposure that a certain part of OoD data (also called known unknowns) are prepared for training. On the other hand, OoD detection requires intentionally prepared outlier data during training. This paper analyzes two-head network included in OoD detection models, and semi-supervised OpenMatch associated with OSR technology, which explicitly and implicitly assume outlier exposure, respectively. For the experiment, we built an image dataset of eight strawberry diseases. In general, a two-head network and OpenMatch cannot be compared due to different training settings. In our experiment, we changed their training procedures to make them similar for comparison and show that modified training procedures resulted in reasonable performance, including more than 90% accuracy for strawberry disease classification as well as detection of unknown diseases. Accurate detection of unknown diseases is an important prerequisite for continued learning.

KEYWORDS

continual learning, plant diseases, Open Set Recognition, Out-of-Distribution detection, two-head network, OpenMatch, strawberry disease classification

Introduction

Plant disease monitoring is a critical means of improving productivity and enhancing crop quality. The traditional methods for diagnosis of plant diseases–visual analysis by a professional farmer or inspection of a sample in a laboratory–generally requires extensive professional knowledge and high costs. For this reason, an automated disease monitoring process will prove to be a valuable supplement to the labor and skill of farmers (Kim et al., 2021).

A number of research studies have applied deep learning techniques to automatic plant disease monitoring (Liu and Wang, 2021). However, most of the studies have been based on closed set recognition (CSR), which is prone to erroneous decisions when an unknown disease sample is detected because it must be classified into one of known classes. Moreover, discriminating images of plant diseases (or disorders) is a difficult task for computer vision, categorized into a fine-grained task involving both easy and hard problems.

In contrast, a human expert can naturally accumulate knowledge to improve their ability to accurately recognize plant diseases or disorders in an increasing number of categories. In order to program a machine to be similar to a human expert, it is necessary to continuously increase the amount of data and the number of categories it has access to. Open Set Recognition (OSR) and Out-of-Distribution (OoD) detection technology are used for continual machine learning and can be applied to plant disease recognition in order to differentiate unknown diseases and disorders from known diseases.

Generally speaking, for continual learning of open world tasks, both detection of unknown diseases and incremental active learning with unknowns should be addressed. However, unknowns need to be correctly identified before commencing active and incremental learning, so that their detection is essential to lifelong or continual learning for the performance of open world tasks.

Figure 1 shows the continual learning process for plant disease monitoring. The unknowns should be identified in the inference stage, and then a second round of training is performed with additional known and unknown disease data, with (or without) increased number of categories.

Automatic detection of unknowns has been a traditional field of research (Scheirer et al., 2012) in computer vision and has recently received attention due to deep learning technology's increasing popularity (Cardoso et al., 2015). In general, however, unknowns are not available in the learning process. In conventional CSR, the unknowns must be classified into a known class during the inference process, which degrades performance. To avoid such degradation, OSR should have a proper structure and be carefully trained.

There has been a large volume of research on OSR since it was formalized by Scheirer et al. (2012). Unfortunately, OSR technology in its current state is unable to be practically applied to fine-gained plant disease monitoring due to poor performance without assuming outlier (sometimes called known unknowns) exposure. OoD detection technology is closely related to OSR, but outliers can be partly assumed and prepared for training differently from OSR. In general, OoD detection encompasses all forms of distributional shift, while OSR specifically refers to semantic novelty (Vaze et al., 2021). However, in plant disease recognition based on image analysis, OSR is similar to the OoD detection when there is no severe distribution shift in captured image data, and a set of outlier data is assumed in the training (outlier exposure). **Figure 1** assumes outlier exposure from the first round of training of the prototype model, because unknowns are incorporated with.

The goal of this paper is finding the practical solutions to detect unknowns for continual learning as shown in **Figure 1**, where a part of outliers is assumed to be prepared for training. For this purpose, the paper evaluates (Yu and Aizawa, 2019) a two-head network that uses OoD detection, and Saito et al. (2021) semi-supervised OpenMatch that uses OSR technology, both of which show reasonable performance for known plant disease recognition as well as unknown disease detection.

It is generally it is not appropriate to compare OoD and OSR because they require different settings for training. In order to change the semi-supervised OpenMatch into OoD detection similar to the two-head network, OpenMatch can be disassembled into two stages; one training stage to learn the One versus All (OVA) and softmax classifiers with labeled and OoD data, and another stage to learn the semi-supervised setting of OpenMatch with unlabeled samples including both inliers and outliers. Also, the two-head network can be retrained with FixMatch and fine-tuned after finding high confidence pseudo inliers and outliers from the inference process. After these modifications, the two different models can be comparable in terms of the performance for detecting and classifying both unknown and known diseases.

As shown in **Figure 1**, these two different modified models of OpenMatch and the two-head network are related with continual learning, because inliers and outliers of unknowns can be effectively recognized, and the results can be used for second round training to continuously improve the models' performance.

The contributions of this paper can be summarized as follows:

1. The difficulty recognizing unknown plant diseases is related to continual learning and the progressive evolution of machine performance. We chose two different types of techniques, OpenMatch using OSR, and a two-head network using OoD detection, which are closely related technologies. To the best of our knowledge, this might be the first article to examine OSR and OoD detection for plant disease monitoring. In addition, this paper shows that OSR outlier exposure is a necessary assumption to adequately detect unknowns.
2. Open Set Recognition and Out-of-Distribution detection are difficult to compare. After the proper modifications, we compared OpenMatch with the two-head network to classify unknown strawberry diseases and related both classification models with continual learning. In addition, our results show that the contrastive regularization in

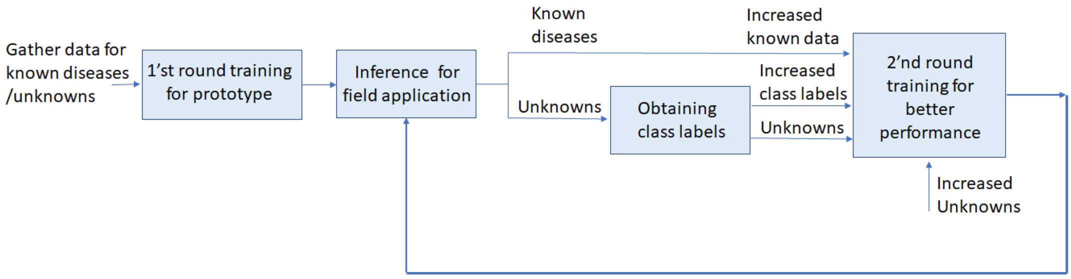

Gather data for known diseases /unknowns
1'st round training for prototype
Inference for field application
Known diseases
Unknowns
Obtaining class labels
Increased known data
Increased class labels
Unknowns
2'nd round training for better performance
Increased Unknowns

FIGURE 1
Unknown detection and continual learning for plant disease monitoring.

FixMatch developed for semi-supervised OpenMatch was successfully applied to the two-head network to improve its performance.

3. We constructed an image dataset of strawberry diseases to validate OSR with assumed outlier exposure. The result of our experiment shows that both the two-head network and OpenMatch can provide reasonable performance for classifying the aforementioned eight strawberry diseases as well as detecting unknowns.

Related works

In this section we summarize the use of OSR and OoD detection for continual learning and DNN-based plant disease monitoring.

Open Set Recognition and Out-of-Distribution detection for continual learning

Recently, open world vision has received considerable attention in the field of computer vision, because it has the potential to resolve many realistic problems such as open set recognition, long-tailed distribution, and limited ontology of labels for life-long or continual learning (Open World Vision, 2021). Open world vision is also related to active or incremental learning because unknowns can be grouped to obtain labels, or should be learned with increased number of categories without catastrophic forgetting (Parisi et al., 2019).

An important task in open world vision is properly differentiating unknowns from known classes. In the inference phase of CSR, a sample should be classified into known classes included in the training phase. When using OSR, however, a classification model must be able to distinguish between the training classes, and indicate if an image comes from a class it has not yet encountered (Scheirer et al., 2012). This implies that unknowns are not exposed to the model during OSR training.

There are several types of deep learning-based OSR models. OpenMax (Bendale and Boult, 2016) is an extension of SoftMax that uses probability adapting Meta-Recognition concepts to activate patterns in the penultimate layer to recognize unknown. There are many generative models of OSR based on auto-encoders or GANs (Generalized Adversarial Networks). G (Generative)-OpenMax is an extension of OpenMax, in which unknown unknown class samples are artificially generated with GANs and are used for fine-tuning OpenMax (Ge et al., 2017). A class-conditioned Auto-Encoder for OSR is another kind of generative model in which an encoder/decoder model is used to classify known classes and unknowns (Oza and Patel, 2019). Outlier exposure is a necessary assumption to improve OCR performance, but there is a risk of overfitting, because only a limited amount of the voluminous outlier data is available for training. OpenGAN is the most recent generative model in which outlier exposure is assumed, but additional GANs are applied to supplement outlier data to prevent overfitting (Kong and Ramanan, 2021). OpenHybrid framework consists of an encoder to encode the input data into a joint embedding space, a classifier to classify samples to inlier classes, and a flow-based density estimator to detect whether a sample belongs to the unknown category (Zhang et al., 2020). There are many recent papers continuously being published with tutorials in OpenSetRecognition_list (2022).

While OSR is closely related to OoD detection (Hendrycks and Gimpel, 2016), OoD settings permit the use of additional data as examples of "OoD" data during training (Chen et al., 2021). Many deep leaning-based OoD detection methods have been developed. The maximum softmax probability is the simplest one to decide if something is an inlier or outlier. Generalized ODIN (Hsu et al., 2020), an extended version of ODIN, uses the decomposed confidence model, temperature scaling, and modified input preprocessing strategies (Liang et al., 2017). Also, many OoD detection methods were introduced by Salehi et al. (2021) including the two-head network that we consider in this paper.

The two-head network in the paper was published by Yu and Aizawa (2019) to find OoD samples. In plant disease monitoring, the set of OoD samples can included unknown diseases or disorders, as well as other images irrelevant to the task. When unknowns are included in OoD detection training data, they are called known unknowns. The set of OoD data prepared for training is a type of bias (Hsu et al., 2020), and reasonable OoD data should be chosen in two-head network training.

The OSR algorithm OpenMatch in the paper was released in 2021 (Saito et al., 2021), and an advanced modified version was published which added contrastive loss (Lee et al., 2022). The networks in OpenMatch are trained in a semi-supervised setting, which is different from OoD-based detection of the two-head network. However, semi-supervised learning can be treated as a method to expose outliers for training, because unlabeled data can include OoD samples as well as unlabeled inliers.

Due to (*a priori*) known unknowns in the training phase, it is hard to directly compare OoD detection with OSR. However, in practice, the distinction between OSR and OoD detection is not important if the outlier images are well prepared.

Related works of deep learning-based plant disease monitoring

There are two types of deep learning models for plant disease monitoring: classification and deep object detection. The classification model can be used to find the name of a disease after an image is manually taken by a camera (Mohanty et al., 2016). In contrast, the deep object detection model can place the diseased area in a bounding box, so that it can be applied to automatic disease monitoring if the imaging apparatus is equipped with a mobile robot. There are excellent studies reported by Kim et al. (2021) and Liu and Wang (2021).

The following discussion focuses on the classification model, as we tried to apply said model to recognize the diseases with unknowns. In general, unknown object detection is a much more complicated task than object identification (Joseph et al., 2021).

There have been a number of deep neural network (DNN)-based classification approaches used to identify plant diseases and disorders. The DNN usually consists of a multilayer convolutional neural network (CNN)-based feature representation block (backbone), and a softmax classification block (head). **Table 1** displays several selected applications of plant disease classification. The backbone network can be used depending on requirements of the applications. If fast recognition speed is required to scarify the accuracy, then a light DNN model like MobileNet may be a prudent choice (You and Lee, 2020). If the accuracy is more important than the speed, then a complex DNN backbone like ResNet might be optimal (He et al., 2016). There are numerous CNN-based off-the-shelf DNN backbones one can choose according to specific requirements (Tan and Le, 2019). A transformer-based backbone is another option to select as a DNN backbone (Dosovitskiy et al., 2020). Note that the backbone can be constructed to obtain better performance by including multiscale methods (Lin et al., 2017).

The head structure of softmax classifiers is similar to each other, where the conditional probability distribution of class labels for given input image. Note that there might be multi-label classifiers which have more than one head. In this case, each separate head can be constructed using separate softmax classifiers to share the DNN backbone during multitasking and by sigmoid classifiers. In this paper, the K-OVA block in the OpenMatch structure has K separate softmax classifiers that share the backbone.

TABLE 1 Deep neural network (DNN)-based classification approaches for identification of plant diseases.

References	Network models	Dataset for pre-training	Plants	Dataset for fine-tuning	Disease classes
Barbedo, 2018	GoogleNet	ImageNet	12 spices		12
Ferentinos, 2018	AlexNet, GoogleNet, Overfeat, VGG16, AlexNetOWTBn		25 species	PlantVillage	58
Liu et al., 2017	AlexNet	ImageNet	apple	Collected from fields	4
Mukti and Biswas, 2019	AlexNet, VGG16,19, ResNet50	ImageNet	38 species	PlantVillage	38
Saleem et al., 2019	AlexNet, LeNet, VGG, GooLeNet, ResNet, DenseNet	ImageNet	38 species	PlantVillage	38
Kumar et al., 2020	ResNet34	ImageNet	14 species	New Plant Diseases Dataset	38
Rangarajan et al., 2018	AlexNet, VGG 16	ImageNet	7 species	Tomato crop	6
Aquil and Ishak, 2021	Vgg16,19, ResNet18,34,50,101, DenseNet120, SqueezeNet	PlantVillage	44 species	tomato leaves	9
Rao et al., 2022	VGG, ResNet based on Bi-CNN		38 species	PlantVillage	38
Rehman et al., 2022	MobileNetv2, DenseNet201	ImageNet	citrus	citrus diseases	6

Transfer learning is widely used due to the lack of training data in many application areas, including plant disease monitoring, where a pre-trained backbone with a huge amount of data in the general domain is initialized to be fine-tuned in a specific application domain. For this purpose, a set of pre-trained parameters for the specific backbone model with an ImageNet dataset is available for constructing the classifier. However, the ImageNet dataset is so general that the domain-specific dataset such as LifeClEF 2017 might be the better choice for a backbone to be used for a specific application (Joly et al., 2017).

Many initial DNN-based plant disease monitoring systems were developed using the PlantVillage dataset which included a diverse group of crops. However, the success of DNN-based monitoring has resulted in diverse datasets built for various crops.

However, it is difficult to find previous research concerning the detection of unknown diseases, except cassava disease classification using CropNet (CropNet, 2020), where the network tried to classify four major cassava diseases on diseased leaves, normal leaves, and unknown. The detection technology of CropNet cannot be identified in detail, but presumably it is not a very complex algorithm.

Meanwhile, there are more than 70 diseases and disorders introduced in Strawberry Diseases (2022), and it is difficult to paper sufficient data for all of them at once. Therefore, the probability of continual learning for detecting diseases and disorders increases with the increased number of classes and corresponding data. **Figure 2** shows images of the 8 classes of known diseases and several unknown disorders. Note that the plant parts including fruit, leaves, runners, and flowers are easy to differentiate, while diseases of the same plant part are difficult to discern. As a result, the disease recognition task is fine-grained, having both easy and hard problems.

Materials and methods

In this section, we introduce the two-head network (Yu and Aizawa, 2019) and OpenMatch (Saito et al., 2021) which were used in the experiments. We discuss how the two-head network can be implemented to recognize unknowns such as OoD, and how semi-supervised learning can be performed to better identify unknowns. In addition, we review how to change the networks so they can be compared, and how we can use them for continual learning.

Two-head network

The two-head network uses two different randomly initialized softmax heads, F_1 and F_2, that provide the same decision for labeled data, but different probability distribution for OoD data. **Figure 3** shows the structure of a two-head network that shares a backbone. Originally there are two stages of training: pre-training with only labeled inlier data (ID), and fine-tuning with unlabeled OoD data. The training loss for labeled ID in the first stage is given by the cross-entropy:

$$L^{two}_{cross}(X) = -\frac{1}{X}\sum_{x_b\in X}\sum_{i=1}^{2}\log(p_i(y_b|x_b)) \qquad (1)$$

where $\{x_b, y_b\}$ is the labeled ID samples, and index i is the head number.

In the second fine-tuning stage, the discrepancy loss is as follows:

$$L^{two}_{dis}(O) = max\left\{m - \frac{1}{\mu_O}\sum_{x_o\in\mu_O} d(p_1(y|x_o), p_2(y|x_o))\right\} \qquad (2)$$

$$d(p_1(y|x_o), p_2(y|x_o)) = \sum_{i=1}^{K}|p_1(y_i|x_o) - p_2(y_i|x_o)| \qquad (3)$$

where $d(\cdot)$ is the $L1$ loss, and $O = \{x_o\}_{o=1}^{\mu_O}$ is the set of unlabeled OoD data. In Eq. 2, m is a margin to prevent overfitting.

The OoD can be any irrelevant data to ID; it can be healthy leaves, fruit, runners or other images for strawberry disease recognition. Note that this OoD data is a type of bias that is inevitable in the OoD detector. Therefore, it is important to use them to increase the network's performance. In Section "Experimental results of the two-head network," we discuss the OoD data in more detail.

For continual learning, as displayed in **Figure 1**, the model can be retrained after performing an inference of unlabeled data. The inference process differentiates ID from OoD data. In the second-round training for continual learning, ID and OoD data are augmented by adding ID and OoD data.

Semi-supervised OpenMatch

OpenMatch uses semi-supervised learning to improve OSR, where labeled and unlabeled data are mixed to create training data. **Figure 4** shows the structure of the OpenMatch model. The base classifier consists of K one-vs-all (OVA) sub-classifiers $D^j(\cdot)$, $j \in \{1, \ldots, K\}$, that share the feature extractor $F(\cdot)$, each of which determines whether it is an inlier or not with respect to the class. There is one more closed set classifier $C(\cdot)$, which gives the class label $\hat{y}$ in one of K classes for an input sample. The final unknown decision of whether it is an inlier or outlier is based on $D^{\hat{y}}(\cdot)$. The training of OpenMatch includes several losses and tries to minimize them. One of the losses is the cross-entropy loss for a closed set classifier:

$$L_{cross}(X) = -\frac{1}{B}\sum_{x_b\in X} y_b\log(p(y|x_b)) \qquad (4)$$

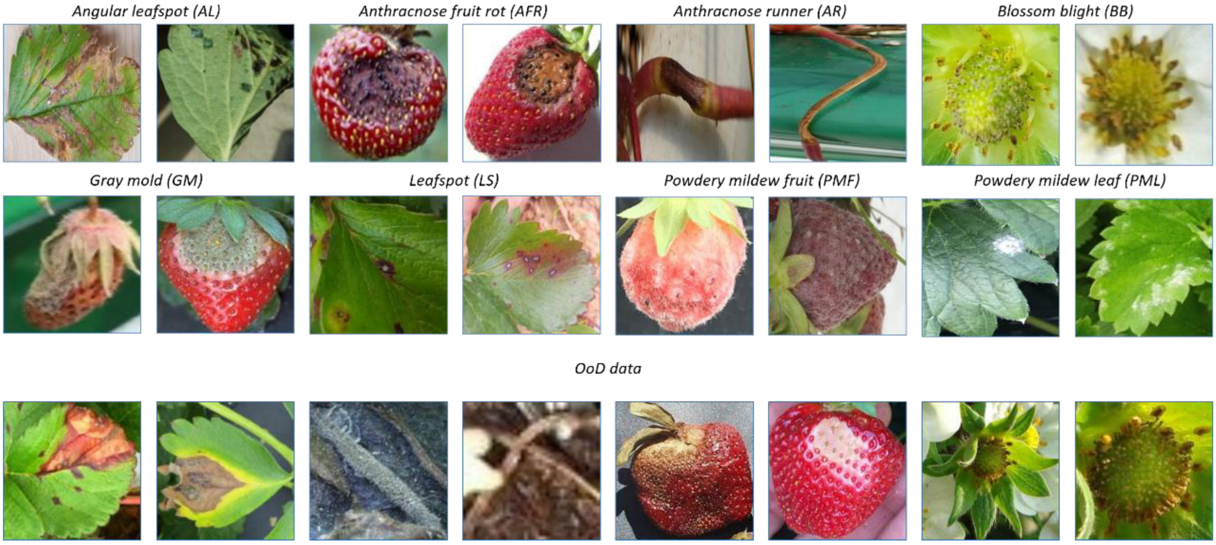


FIGURE 2
Prototypical images of known diseases and unknown diseases.

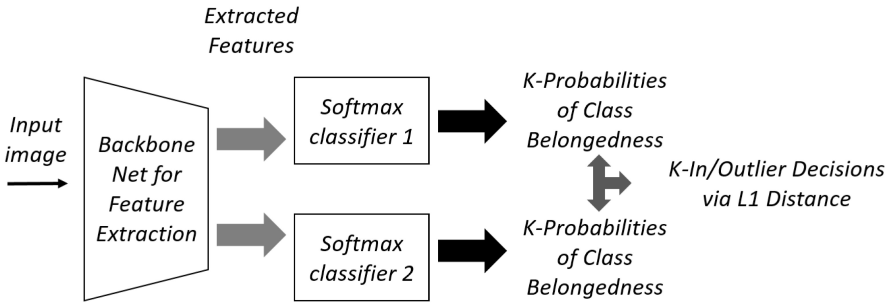

Extracted Features
Input image
Backbone Net for Feature Extraction
Softmax classifier 1
Softmax classifier 2
K-Probabilities of Class Belongedness
K-In/Outlier Decisions via L1 Distance
K-Probabilities of Class Belongedness

FIGURE 3
Two-head network for Out-of-Distribution (OoD) detection.

For a given batch of known data, $X = \{(x_b, y_b)\}_{b=1}^{B}$. In Eq. 4, $p(y_b|x_b)$ is the probability of softmax output y for x_b from closed set classifier $C(\cdot)$. Another loss for the OVA outlier detection is defined as:

$$L_{OVA}(X) = \frac{1}{B}\sum_{b=1}^{B} -\log\left(p^{y_b}\left(t = 0|x_b\right)\right) - min_{i\neq y_b} \log\left(p^{i}\left(t = 1|x_b\right)\right) \tag{5}$$

where $p^i(t = 0|x_b)$ and $p^i(t = 1|x_b)$ represents the probabilities of x_b being an inlier or outlier for class i. For unlabeled data $U = \{(u_b)\}_{b=1}^{\mu_B}$, there is another loss for OVA called entropy minimization, defined as:

$$L_{em}(U) = -\frac{1}{\mu_B}\sum_{b=1}^{\mu_B}\sum_{j=1}^{k} p^j\left(t = 0|u_b\right)\log\left(p^j\left(t = 0|u_b\right)\right) + p^j\left(t = 1|u_b\right)\log\left(p^j\left(t = 1|u_b\right)\right) \tag{6}$$

Equation 7 is the soft open set consistency regularization (SOCR) loss for the OVA classifier to encourage the consistency of the output logits over any augmentation A to enhance the smoothness:

$$L_{OC}(U, A) = -\frac{1}{\mu_B}\sum_{b=1}^{\mu_B}\sum_{j=1}^{k}\sum_{t\in\{0,1\}} \left|p^j\left(t|A_1(u_b)\right) - p^j\left(t|A_2(u_b)\right)\right| \tag{7}$$

which emphasizes the consistency of OVA for differently augmented A_1 and A_2 unlabeled data.

During semi-supervised learning, unlabeled samples are taken as pseudo inliers to supplement the set of labeled data, if $p^{\hat{y}}\left(t = 0|u_b\right) = \tau$, where $\hat{y} = argmax_j\, C(F(u_b)))$, after the training is stabilized.

The learning related to these pseudo inliers is called FixMatch (Sohn et al., 2020), and there is another corresponding

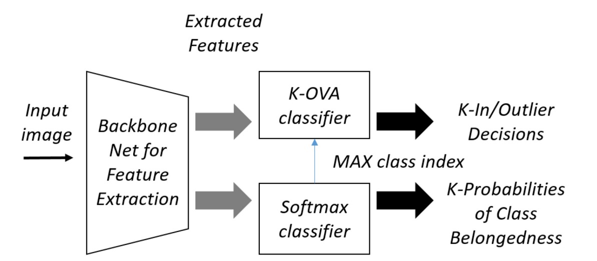


FIGURE 4
OpenMatch with softmax and One versus All (OVA) classifiers.

loss to be minimized L_{fm}. FixMatch is a combination of two approaches to semi-supervised learning: consistency regularization and pseudo-labeling (Sohn et al., 2020). Consistency regularization utilizes unlabeled data by relying on the assumption that the model should output similar predictions when fed perturbed versions of the same image (weak augmentation and strong augmentation). Pseudo-labeling leverages the idea of using the model itself to obtain artificial labels for unlabeled data. FixMatch progressively improves the performance of semi-supervised training (so-called curriculum learning) using pseudo-labeled data, where strong augmented pseudo inliers follow weak augmented ones. The FixMatch process can extend the decision boundary of known classes to allow the strongly augmented inliers to train models. Here, the corresponding loss can be described as:

$$L_{fm} = -\sum_{b=1}^{\mu_B} \mathbb{I}\left(p^{\hat{y}}\left(t = 0|u_b\right) > \tau\right) \log p\left(\hat{y}|A\left(u_b\right)\right) \quad (8)$$

where $\mathbb{I}()$ is a set indicator function, and $A\left(u_b\right)$ stands for the strong augmented data for the pseudo inlier. Note that L_{fm} is the same as the cross-entropy losses except that they are calculated for pseudo inliers labeled by $\hat{y}$.

A contrastive loss can also be applied to OpenMatch to improve the accuracy and speed of the FixMatch training process (Sohn et al., 2020). FixMatch only considers consistency regularization between each high confidence pseudo inlier $\left(p^{\hat{y}}\left(t = 0|u_b\right) > \tau\right)$ and its strong augmented version $A\left(u_b\right)$ by curriculum learning. On the other hand, contrastive regularization builds a pool of strong augmented samples of pseudo inliers where both positive and negative samples for pseudo-labeled data are included, and then tries to minimize the contrastive loss. In order to implement contrastive regularization, a pool of strong augmented unlabeled ID

$$A_m\left(U\right) = \left\{u'|u_b \in U, p^{\hat{y}}\left(t = 0|u_b\right) > \tau,\ u'_i = A\left(u_b\right),\right.$$
$$\left.1 \leq i \leq m\right\} \quad (9)$$

is first built, in which the average contrastive loss is calculated using the positive and negative pairs. In Eq. 9, m strong augmented data for each pseudo inlier is included in $A_m\left(U\right)$. The contrastive loss for a sample $u^{'}$ in $A_m\left(U\right)$ can be calculated by:

$$r\left(u^{'}\right) = \frac{-1}{\left|\hat{P}(u^{'})\right|} \sum_{p^{'} \in \hat{P}(u^{'})} \log \frac{\exp\left(\left\langle z_{u'}, z_{p'}\right\rangle / T\right)}{\sum_{v^{'} \in A_m(U)/u^{'}} \exp\left(\left\langle z_{u'}, z_{p'}\right\rangle / T\right)} \quad (10)$$

where $\hat{P}\left(u^{'}\right) = \left\{p^{'}|p^{'} \in A_m\left(U\right)/u^{'}, \hat{q}_{p'} = \hat{q}_{u'}\right\}$ is a set of $p^{'}$ which makes so-called pseudo positive pairs with $u^{'}$, that has the same pseudo label $\hat{q}_{p'}$ as $\hat{q}_{u'}$. In Eq. 10, T is temperature scaling parameter, and $z_{u'}$ is a normalized vector of the projection head.

Figure 5 shows OpenMatch with FixMatch-included contrastive regularization. In semi-supervised training, the degree of confidence in ID or OoD data is determined by OVA classifiers and its pseudo label assigned by the softmax classifier, as demonstrated in Eq. 8. In **Figure 5**, FixMatch uses the pairs of weak and strong augmented pseudo inliers for consistency regularization of the softmax classifier, and the pool of strong augmented pairs of pseudo-inliers are utilized for the contrastive regularization of feature embedding as demonstrated in Eqs 9, 10.

Discussions and comparison models

Outlier exposure

The two-head network explicitly includes OoD data with ID in its training for fine-tuning. On the contrary, OpenMatch improves OSR performance using semi-supervised learning, where the unlabeled data implicitly includes OoD data to better learn OVA according to the losses in Eqs 6, 7. OpenMatch assumes outlier exposure implicitly in unlabeled data.

As aforementioned, the set of OoD data prepared in the training process can be considered as a type of bias and a reason for overfitting, because it cannot include the large amount of OoD data; so-called unknown unknown space. Therefore, preparing an adequate and efficient set of OoD data for a specific domain is important. This is further discussed with the experimental results in the Section "Experimental results of the two-head network."

Comparison models and continual learning

Two-stage training of the two-head network may be merged into single stage semi-supervised training, which starts with labeled and unlabeled data in the same manner as semi-supervised OpenMatch. In this case of two-head network, unlabeled ID can be treated as pseudo-labeled inliers after stabilizing the second stage of the fine-tuning process. Also, FixMatch with additional loss L_{fm} may be applied in the two-head network with contrastive regularization. However, this semi-supervised alignment of two-head network and OpenMatch is not intuitive. Therefore, we considered another modification to make a comparison between the

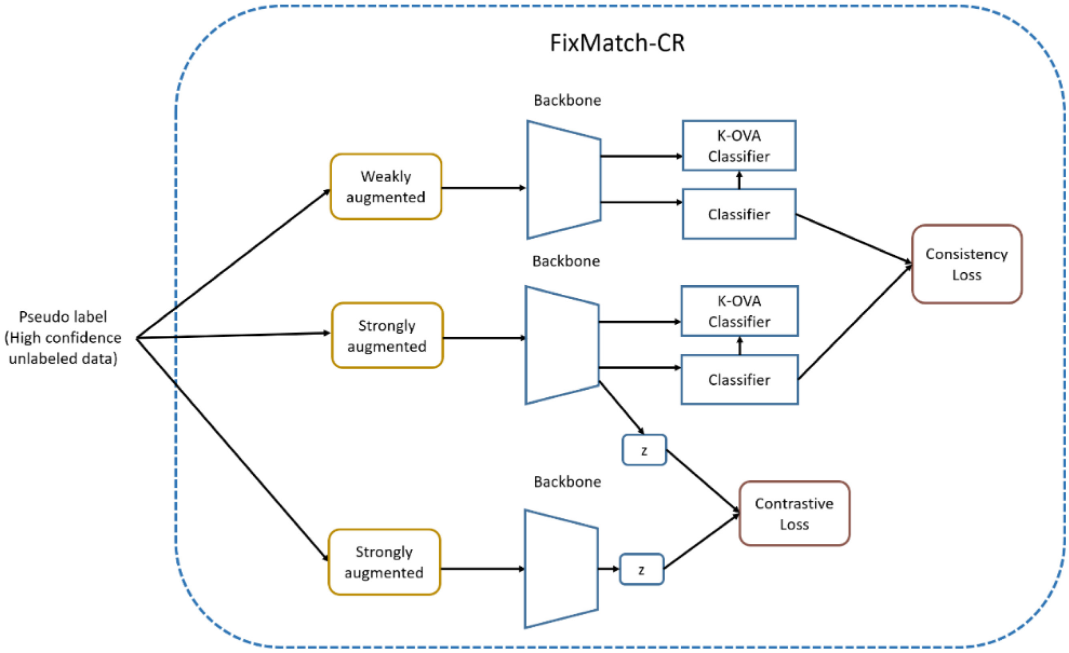


FIGURE 5
FixMatch with contrastive regularization in OpenMatch.

two-head network and semi-supervised OpenMatch, as shown in Algorithm 1.

```
Step 1: Train two classifiers with ID
and perform fine-tuning with OoD data.
Step 2: Inference unlabeled data
including ID and OoD data.
Step 3: Perform FixMatch with
pseudo-labeled data in Step 2.
Step 4: Perform fine-tuning with OoD
data in Step 1 and Step 2.
```

Algorithm 1: Modified two-head network.

In Algorithm 1, the two-head network is retrained using FixMatch similar to OpenMatch in Step 3. FixMatch can be performed with pseudo-labeled data to improve the performance of the two-head classifier after the inference process of unlabeled data in Step 2, in the same manner as in the semi-supervised OpenMatch. Note that the high confidence pseudo inlier can be detected by Eq. 3, however, the discrepancy must be smaller than the threshold for unlabeled ID data. The set of ID samples with pseudo labels obtained from the softmax decision can then be used for FixMatch (with contrastive regularization), as shown in Figure 6. Each softmax classifier head is separately adjusted for consistency loss in FixMatch, and the backbone can learn contrastive loss.

Finally, the outliers in the inference process of Step 2 can be used to fine-tune the two classifier heads in Step 4. Instead of inherent inference and FixMatch in the training loop of semi-supervised OpenMatch, the two-head network performs FixMatch and fine-tuning after explicit inferencing of unlabeled data.

Also, in order to compare semi-supervised OpenMatch with the two-head network, OpenMatch can be disassembled into two stages; one training stage to learn the One versus All (OVA) and softmax classifiers similar to OoD detection, and a second stage to conduct semi-supervised learning of OpenMatch with labeled and unlabeled samples. In the second stage, the same OoD data used in the first stage is included in the outlier data. OSR does not intentionally include OoD data in the training phase, but the data was prepared for the two training stages of the disassembled OpenMatch. So, OpenMatch can also be treated as an OoD detector. Algorithm 2 summarizes the disassembled training process of OpenMatch. Note that the first training stage is prepared only for comparing the OoD detection capability with the two-head network.

```
Step 1: Train softmax classifier
and $k$-OVA classifiers with labeled
ID and OoD data.
Step 2: Perform semi-supervised
training with labeled ID, unlabeled
data, and OoD data from Step 1.
```

Algorithm 2: Modified semi-supervised OpenMatch.

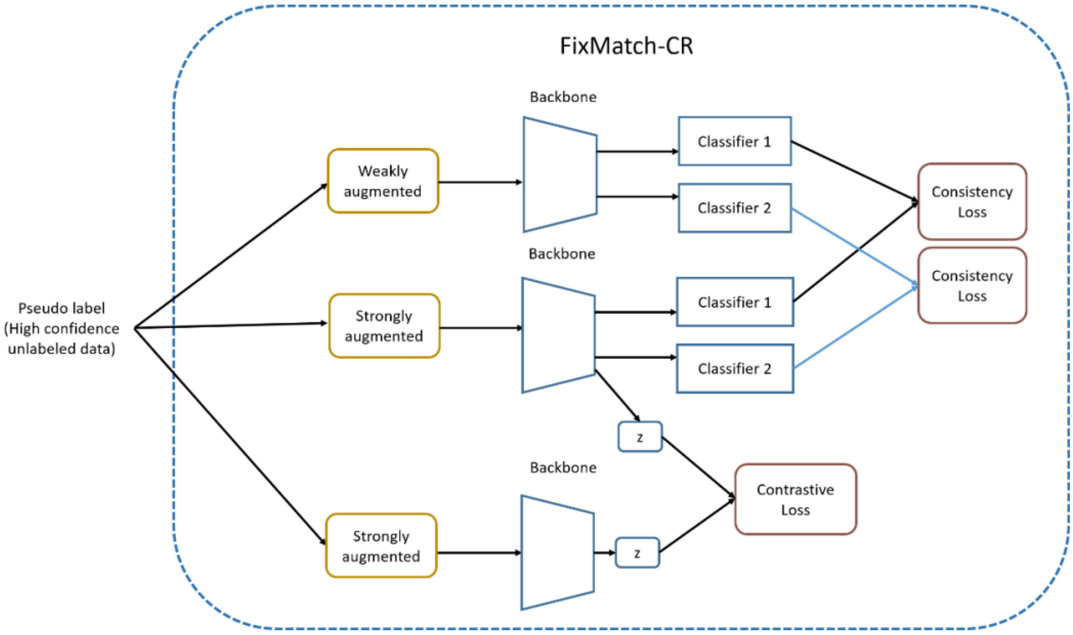


FIGURE 6
FixMatch with contrastive regularization in two-head network.

TABLE 2 Image dataset of strawberry diseases and unknown diseases.

Name of disease	Total no. of images	Training images	Validation images	Test images
Angular leafspot (ALS)	818	498	184	136
Anthracnose fruit rot (AFR)	188	137	32	19
Anthracnose runner (AR)	232	129	33	70
Blossom blight (BB)	1,898	1,410	264	224
Gray mold (GM)	1,303	1,003	171	129
Leaf spot (LS)	2,299	1,703	360	236
Powdery mildew fruit (PML)	397	236	77	84
Powdery mildew leaf (PML)	1,738	1,257	232	249
Unknown or OoD diseases or disorders	4,216	1,346	1,435	1,435
Total	13,089	7,719	2,788	2,582

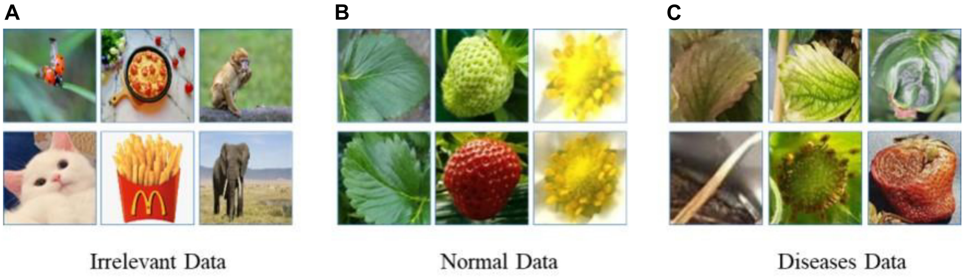


FIGURE 7
Three types of Out-of-Distribution (OoD) data.

After the modifications, OpenMatch and the two-head network had approximately the same conditions for comparison, including the same data for training, inference, and retraining. Note that Steps 2 through Step 4 in Algorithm 1 are included in the semi-supervised training loop of OpenMatch, which closely aligns the two models.

As well, both Algorithm 1, 2 are associated with continual learning, because they include an inference process of unlabeled data, and the results are utilized to improve the performance of OSR or OoD detection. The two-head network explicitly improves the performance by adding Step 2s through Step 4 in Algorithm 2, while semi-supervised OpenMatch includes continual learning inherent in Step 2 of Algorithm 2.

The continual learning process that utilizes each of the two different models as a whole is possible as follows: When training a two-head network using Algorithm 1, it can be used to determine inliers and OoD data during the actual inference performed in Step 2. We can then use the classified unlabeled data with high confidence as pseudo labeled data for performing FixMatch in Step 3. The supplemented OoD data can then be continuously added to fine-tune the model in Step 3 to improve the performance. If it is necessary to cluster OoD data to obtain new labels, then the retraining from Step 1 is possible with the new head structure.

On the contrary, OpenMatch trained by Algorithm 2 can be used to recognize unknowns of OoD data and ID in the real inference process. As shown in Figure 1, an increased amount of labeled and unlabeled data, including confident ID and OoD data, can then be prepared for Step 2 in Algorithm 2. In this process, the unknowns of OoD data can be clustered to give new labels and include them for incremental learning. In this case, retraining from Step 1 is necessary to adjust the extended structure of the model.

Complexity of the two models

The complexity of the two-head network and OpenMatch is comparable, because the two-head network includes two softmax classifiers, while OpenMatch includes one softmax classifier and *k*-OVAs. If the number of class

TABLE 3 Performance of the two-head network with different experimental settings.

1st						2nd	Improved 2nd	
Irrelevant	Normal	Unknowns (diseases/ disorders)	I + N	N + D	I + N + D	Using pseudo label	Without contrastive regularization	With contrastive regularization
Accuracy								
0.785	0.855	0.865	0.858	0.862	0.861	0.883	0.901	0.924
AUROC								
0.828	0.917	0.919	0.912	0.926	0.923	0.940	0.951	0.972

1st: 1st round training followed by fine-tuning. 2nd: 2nd round without FixMatch followed by fine-tuning. Improved 2nd: 2nd round FixMatch-CR followed by fine-tuning. I + N: Fine-tuning OoD data is a mix of irrelevant and normal data. N + D: Fine-tuning OoD data is a mix of normal and diseases data. I + N + D: Fine-tuning OoD data is a mix of irrelevant, normal, and diseases data.

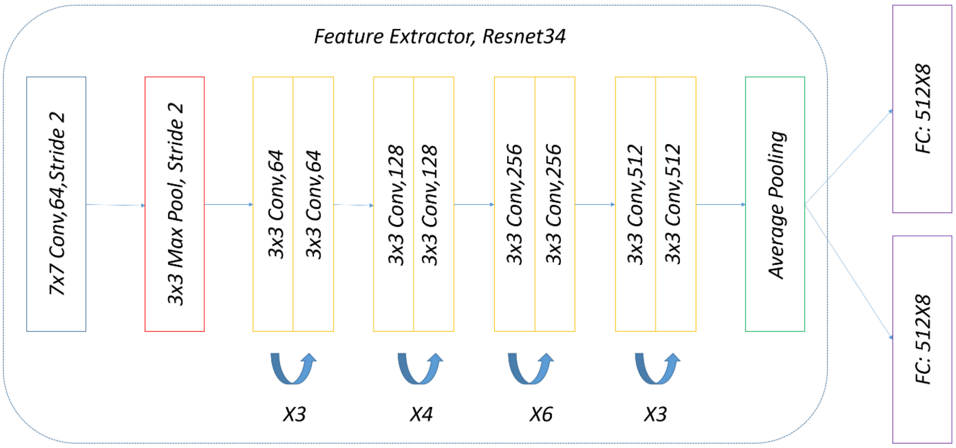


FIGURE 8
Two-head network for experiment.

K is large, then the two-head network is simpler than OpenMatch; otherwise, OpenMatch is preferable in term of complexity.

Experimental results

To conduct the experiment, we constructed a small dataset of strawberry diseases with unknowns that were used for training, validating the results, and testing. The experiments analyzed the effect of different types of OoD data, and the improvements of performance by adding technological components such as FixMatch with contrastive regularization.

Dataset of strawberry diseases with unknowns

This paper considers a two-head network and OpenMatch classifiers for monitoring strawberry diseases. For validation purposes, we built a strawberry disease dataset which included eight disease categories: angular leafspot, anthracnose (fruit rot,

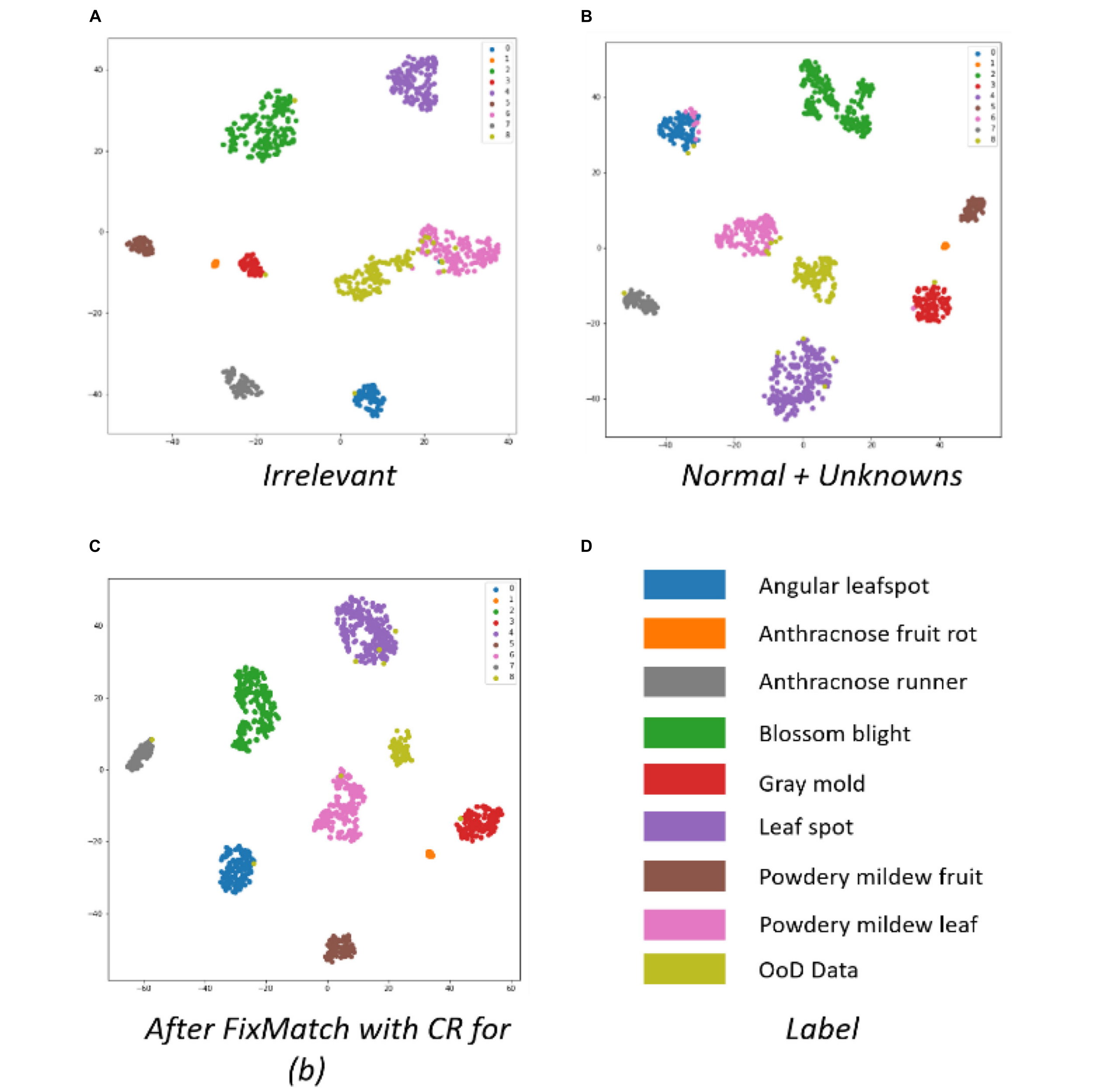


FIGURE 9
Comparison of cluster structures using t-SNE.

runner), blossom blight, gray mold (fruit), leafspot, powdery mildew (fruit, leaf); as well as unknown diseases and/or disorders. Reportedly there are more than 70 strawberry diseases or disorders (Strawberry Diseases, 2022); however, only eight diseases are considered as known diseases in our work. Other diseases or disorders we do not consider in the experiment were treated as unknowns for continual learning. There are 13,089 images including 8,873 known diseases and 4,216 unknown diseases, as displayed in **Table 2**. **Figure 2** shows prototypical images of known and unknown diseases. All the images were captured in more than 6 greenhouses by cellular phone cameras, because the system pursues a mobile application.

Experimental results of the two-head network

Training the two-head network for comparison with OpenMatch

The training of the two-head network consisted of two stages: the pre-training of each head of softmax classifiers, and fine-tuning with OoD data. To compare the two-head network with semi-supervised OpenMatch, and to show the applicability of continual learning, we added several steps in the training of the two-head network, as shown in **Algorithm 2**.

Step 1 trained the two heads of softmax classifiers with labeled ID and performed fine-tuning using OoD data to maximize the discrepancy between the decisions in the two heads. As explained in Section "Materials and methods" Step 2 performed the inferencing of unlabeled data in the same manner as in semi-supervised training of OpenMatch. After the inference, the pseudo inliers or outliers were obtained from the trained two-head network. The high confidence labeled pseudo inliers were then used by FixMatch with contrastive regularization, as displayed in **Figure 6**. Finally, Step 4 performed fine-tuning with the original OoD data one additional time.

Algorithm 1 used the same labeled and OoD data for training, inferencing to find the pseudo inliers, and fine-tuning, similarly to the disassembled OpenMatch in **Algorithm 2**, in order to compare the two different models.

To train model, the original dataset in the second column of **Table 2** was divided into training, validation, and test data. The training was performed using a random online selection of (weak) augmented data, visually rotated at 90, 180, and 270 degrees. For the intermediate inference stage, we used 2,659 unlabeled inliers and OoD data.

As previously discussed, in the fine-tuning stage in Step 1 of **Algorithm 1**, there were several possible ways to build OoD data, because it could draw from a large unknown data space. One way was to include only irrelevant data randomly selected from the ImageNet dataset, such as bugs, food, and trees. Another method was to include normal (healthy) strawberry data such as flowers, leaves, runners, and fruit. In addition, we could include unknown diseases or disorders that were not part of the known classes. We prepared the same amount of three types of OoD data: irrelevant data, healthy strawberry data, and unknown suspected disease data. **Figure 7** shows the different types of OoD data samples used to train the models. The effects of the three different types of OoD data on the performance of the models is compared in **Table 3** and discussed in Section "Experimental results and discussion."

For FixMatch, we required sets of weak and strong augmentation to gradually improve classification performance. In the experiment, the geometrically transformed images, as previously mentioned, were used for weak augmentation. For strong augmentation, images with color and brightness changes and different degrees of rotations were included, and an augmentation was randomly chosen among 36 different alternatives during FixMatch training.

The precise structure of the two-head network used in the subject experiment is shown in **Figure 8**. ImageNet pre-trained by ResNet34 was selected as a backbone for simplicity, and there were two eight-way softmax classifiers.

TABLE 4 Confusion matrix of the final two-head network (Fixmatch-CR).

	ALS	AFR	AR	BB	GM	LS	PWF	PML	OoD	Recall
Angular leaf spot (ALS)	122	0	0	0	0	0	0	0	14	0.897
Anthracnose fruit rot (AFR)	0	17	0	0	0	0	0	0	2	0.895
Anthracnose runner (AR)	0	0	68	0	0	0	0	0	2	0.971
Blossom blight (BB)	0	0	0	222	0	0	0	0	2	0.991
Gray mold (GM)	0	0	0	0	120	0	0	0	9	0.930
Leaf spot (LS)	0	0	0	0	0	235	0	0	1	0.996
Powdery mildew fruit (PMF)	0	0	0	0	0	0	76	0	8	0.905
Powdery mildew leaf (PML)	1	0	0	0	0	0	0	166	82	0.667
Unknowns (OoD)	8	0	1	0	10	24	0	30	1,362	0.949
Precision	0.931	1.000	0.986	1.000	0.923	0.907	1.000	0.847	0.919	

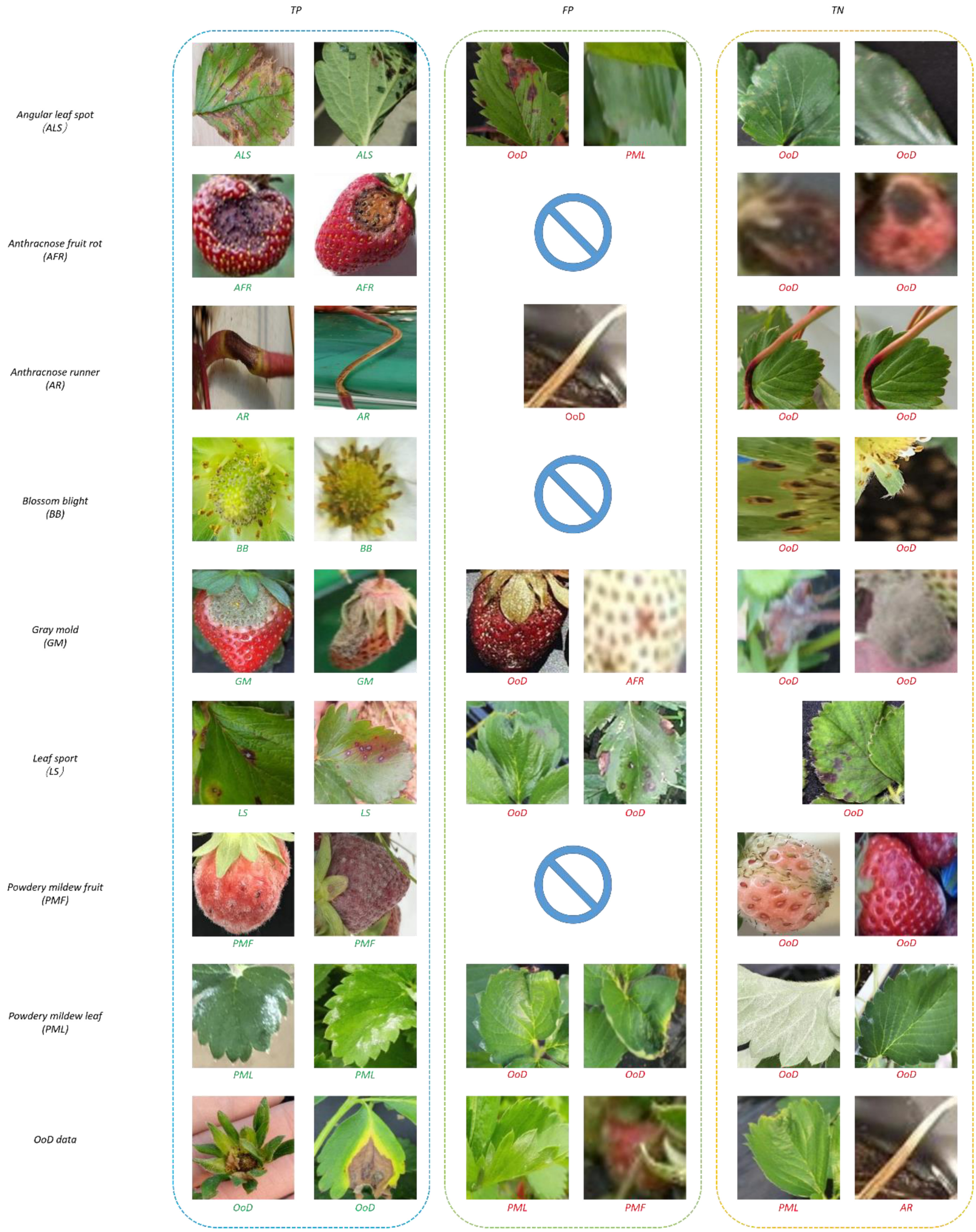


FIGURE 10
Recognition results of correct and incorrect classification.

To train the two-head softmax classifiers, an SGD optimizer was selected with a learning rate that decayed from 0.01. For fine-tuning, the fixed learning rate was set to $2 = 10^{-4}$ using the same SGD optimizer. The batch size was 64 and the number of epochs for pre-training and fine-tuning were 300 and 10, respectively.

Experimental results and discussion

We used classification accuracy with unknown disease and AUROC as evaluation metrics. TP, TN, FP, FN are used to denote true positives, true negatives, false positives, and false negatives, respectively. Accuracy is the ratio of correctly classified samples (TP + TN) to the total number of samples. AUROC is the Area Under the Receiver Operating Characteristic curve and can be calculated by the area under the FPR (FPR = FP/(FP + TN)) against the TPR (TPR = TP/(TP + FN)) curve. We also used precision and recall in the confusion matrix. Precision refers to the proportion of the true positive class (TP) among all judged positive classes (TP + FP). Recall refers to the proportion of all true positive classes (TP + FN) that are judged as positive classes (TP).

Table 3 shows the performance of the pre-training and fine-tuning of the two-head network. In Table 3 we compare performance from the different types of OoD data with the same labeled inlier training data. The combined OoD data, using normal (healthy) strawberry parts including leaves, flowers, fruit, and runners, as well as unknown diseases (or disorders), resulted in satisfactory performance recognizing diseases as well as unknowns. Note that irrelevant OoD data was not helpful to train the two-head network, even though it was included in the mixed OoD data of normal and unknowns, displayed in the sixth column of Table 3. The results in Table 3 show that OoD data selected from healthy plant parts can be helpful, which is useful for practical applications of plant disease monitoring.

Note that the selection of OoD samples is a bias in OoD detection. Biased will inevitably be introduced if the successful performance of the model requires outlier exposure. Generalized ODIN, a similar OoD detector without the bias, demonstrated 81.9% accuracy and 0.894 of AUROC using the same strawberry data. The set of OoD samples, composed of healthy parts and unknowns, might be an inevitable but reasonable bias to enhance the performance of fine-grained unknown disease detection in plants.

Figure 9 shows t-SNE images taken after different kinds of OoD data was trained. Figure 9B shows a more compacted cluster structure of different classes than Figure 9A, which correspond to the first and fifth columns of Table 3, respectively. The t-SNE images demonstrate why irrelevant outliers are not helpful in judging OoD, even if they are exposed during the training of an OoD detector. The irrelevant OoD samples cannot be used as hard negative samples to help the ID class become compact. Therefore, we built the OoD data using normal healthy parts and unknown diseases (or disorders) of strawberries for the rest of experiments.

In the inference stage, we prepare 2,659 images of inliers and OoD data, and select 1,350 high-confidence inliers with pseudo-labels. In order to find the required confident ID, we

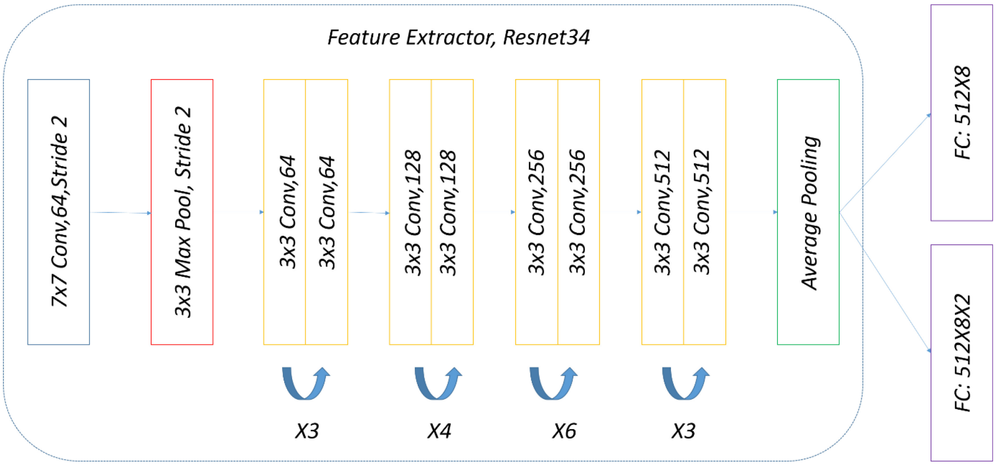


FIGURE 11
Experimental OpenMatch design.

TABLE 5 Performance of disassembled OpenMatch with different experimental settings.

	OpenMatch as OoD detector	OpenMatch without FixMatch	Semi-supervised training with FixMatch	
		Using pseudo label	Without CR	With CR
Accuracy	0.865	0.888	0.900	0.922
AUROC	0.928	0.944	0.951	0.971

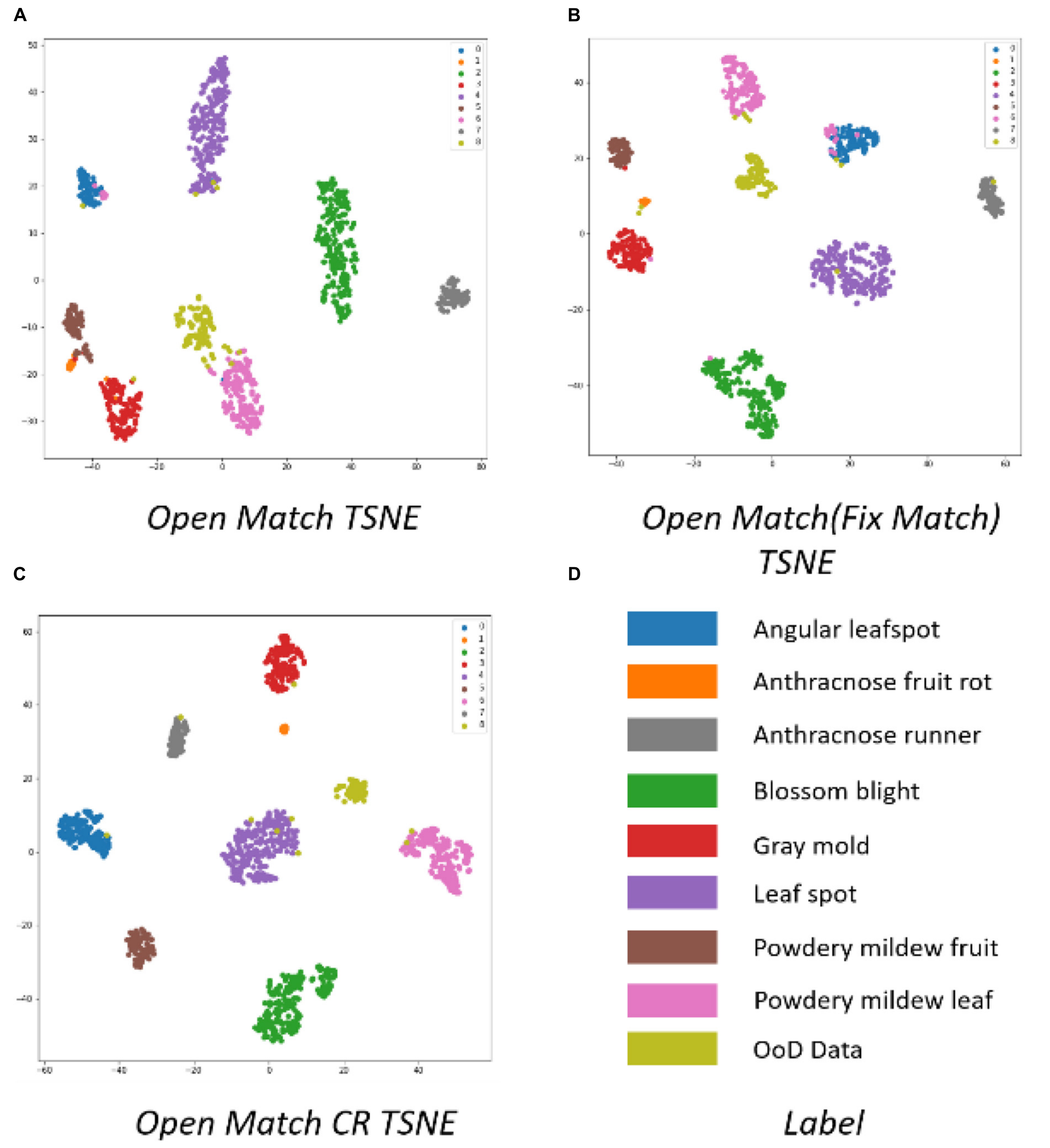


FIGURE 12
Comparison of cluster structures of disassembled OpenMatch using t-SNE.

used two thresholds: the threshold of L1 distance in Eq. 3, and the maximum class probability of two softmax classifiers. The former threshold was determined using a grid search on (0,1) to identify the maximum detection accuracy of OoD in the fine-tuning stage of Step 1, and the latter was 0.95; the same value as in Yu and Aizawa (2019).

The pseudo inliers were used to perform FixMatch with contrastive regularization in order to upgrade the performance of the two closed-set classifiers. Thereafter, the two-head network was fine-tuned with supplemented OoD data determined in the inference stage. The final result of the fine-tuning is shown in the last two columns of Table 3. Note that there was approximately a 3.6 (5.9) % gain using FixMatch (with contrastive regularization) and fine-tuning. When pseudo-ID obtained from the inference stage was used to train the two-head classifier without FixMatch of Step 3 in Algorithm 1, the performance decreased, as displayed in the seventh column of Table 3.

The t-SNE in Figure 9 shows that FixMatch with contrast regularization can make the intra-class distance more compact and the inter-class distance larger.

Note that this sequence of inferencing unknown data, using FixMatch with pseudo inliers, and fine-tuning with outliers, can be repeated to continuously improve the performance of the network.

Table 4 shows a confusion matrix after FixMatch with contrastive regularization followed by fine-tuning. Note that the class label was given only when the decisions from the two heads were consistent. Otherwise, the input image was treated as unknown. Leaf diseases like angular leafspots and powdery mildew (leaf) had reduced recall due to confusion with unknowns. Furthermore, the leaf diseases of angular leafspots, leaf spot, and gray mold (fruit) were inaccurately identified due to confusion with unknowns. The unknown detection results included 94.9% recall and 91.9% precision.

Figure 10 shows samples of recognition results. In Figure 10, all the true negatives (TNs) of leaf and fruit diseases were categorized as unknowns. Note that there were many false positives (FPs) and TNs due to image quality problems including bad illumination and blurring. In addition, some diseases featured small-sized symptoms which were difficult to discern and hard to differentiate, even by human eyes. There were no FPs of flower or runner diseases, due to their distinct shape compared to leaf or fruit diseases.

Experimental results of OpenMatch

Training of OpenMatch for comparison with the two-head network

We dissembled the end-to-end semi-supervised learning into Algorithm 2 in order to compare OpenMatch with the two-head network, as described in Section "Discussions and comparison models." In the first stage, OpenMatch was trained with the same labeled and unlabeled OoD data, similar to the first stage of the two-head network. The OpenMatch was initially treated as if it was an OoD detector. We then performed the semi-supervised OpenMatch training which included inferencing unlabeled data to find the pseudo inliers, as well as using FixMatch with contrastive regularization.

To ensure a fair comparison with the two-head network, the same data and the same weak (strong) augmentation methods at each training stage were used. The precise structure of OpenMatch used in the experiment is shown in Figure 11; ImageNet pre-trained ResNet34 was selected again as a DNN backbone, and there was an eight-way softmax closed-set classifier and 8 OVA classifiers, due to the identification of eight strawberry diseases.

Experimental results and discussion

To train OpenMatch classifiers, an SGD optimizer was selected with a learning rate that decayed from 0.01. For fine-tuning, the fixed learning rate was set to $2 = 10^{-4}$ using the same SGD optimizer. The batch size was 64 and the number of epochs for pre-training and fine-tuning were 300 and 10, respectively.

Table 5 shows the performance of the disassembled OpenMatch across different experimental settings. As a dataset of healthy parts and unknown diseases was identified as the most helpful OoD data, only those samples were used in order to simplify the experiment.

In the second semi-supervised training stage of OpenMatch, the same 2,659 images of inlier and OoD data used in the inference stage of the two-head network were prepared. During second stage training, the high confidence pseudo inliers were detected and applied to perform FixMatch with contrastive regularization, in order to upgrade the performance of the disassembled OpenMatch classifiers. In the experiment, the threshold τ in Eq. 8 was 0.95 to detect pseudo inliers. The result of the training is shown in Table 5.

Using OpenMatch without the semi-supervised method as the OoD detector yields an accuracy of 86.5%, as shown in

TABLE 6 Confusion matrix of the retrained OpenMatch-CR.

	ALS	AFR	AR	BB	GM	LS	PWF	PML	OoD	Recall
Angular leaf spot (ALS)	121	0	0	0	0	0	0	0	15	0.890
Anthracnose fruit rot (AFR)	0	16	0	0	0	0	0	0	3	0.842
Anthracnose runner (AR)	0	0	69	0	0	0	0	0	1	0.986
Blossom blight (BB)	0	0	0	222	0	0	0	0	2	0.991
Gray mold (GM)	0	0	0	0	119	0	0	0	10	0.922
Leaf spot (LS)	0	0	0	0	0	235	0	0	1	0.996
Powdery mildew fruit (PMF)	0	0	0	0	0	0	77	0	7	0.917
Powdery mildew leaf (PML)	5	0	0	0	0	0	0	153	91	0.614
Unknowns (OoD)	3	0	1	0	5	14	0	41	1,371	0.955
Precision	0.831	0.950	0.986	0.991	0.899	0.946	0.974	0.931	0.898	

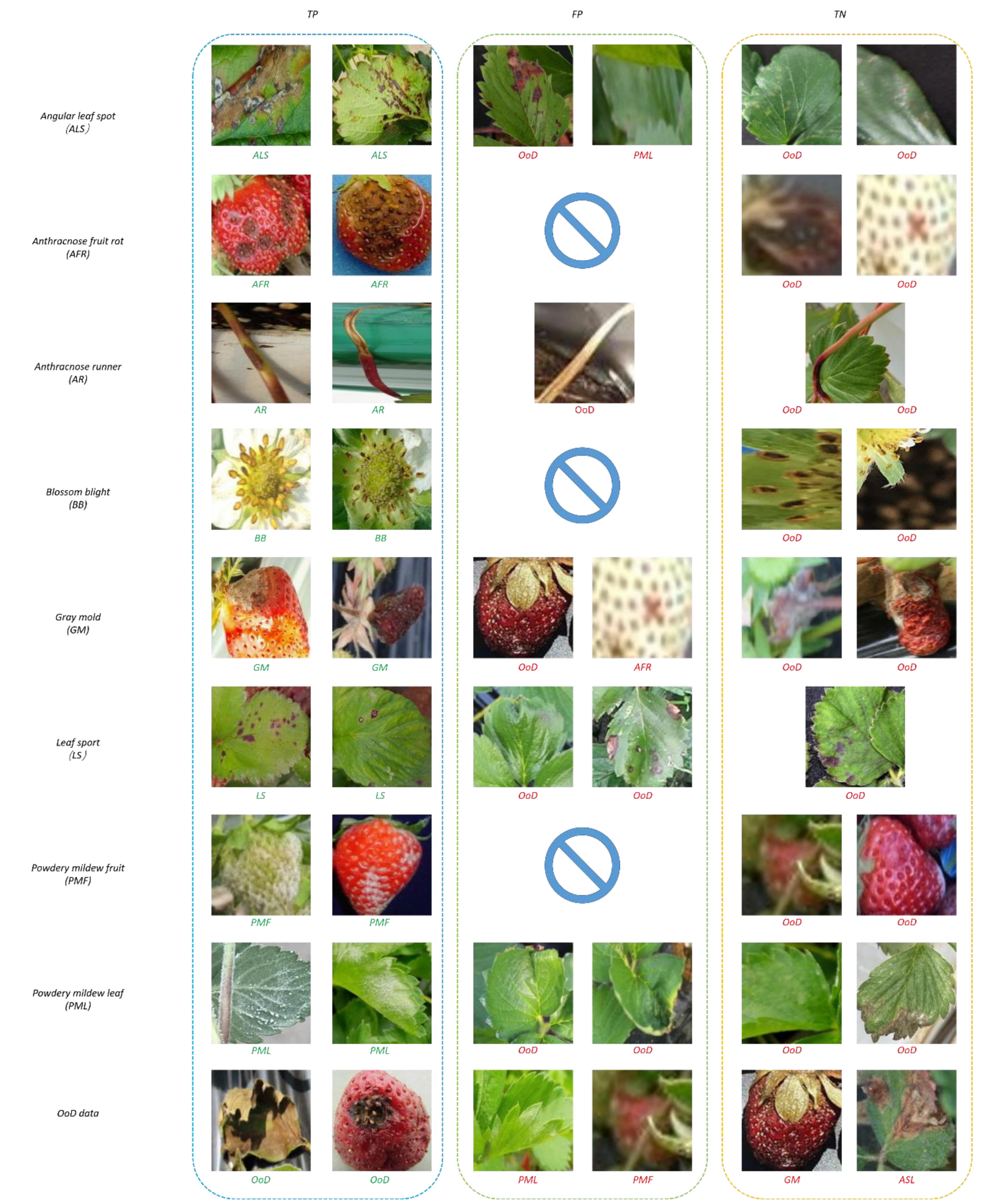


FIGURE 13
Recognition results of correct and incorrect classification.

the first column of Table 5. Note that only the OoD samples were fed as unlabeled data, similar to the fine-tuning stage of the two-head network. The performance of OpenMatch as an OoD detector was comparable with the 86.2% accuracy of the two-head network.

Usually, OSR does not make use of unknowns in the training phase, so that there is no bias regarding the type of unknowns. While OpenMatch with OoD data samples was biased due to unknown exposure during training, it was an inevitable but reasonable bias, similarly observed in the two-head network. When we applied OpenMax, a well-known OSR technique, the accuracy and AUROC were 70.1% and 0.812, respectively. The outlier exposure provided a significant 16.4% increase in accuracy, even though the OpenMax and OpenMatch structures were different.

By combining the semi-supervised training of OpenMatch with FixMatch (with contrastive regularization), accuracy improved as much as 3.5 (5.7) %, as displayed in Table 5, which was comparable with the accuracy of the retrained two-head network. The accuracy improvement might have been a result of semi-supervised learning with unlabeled inliers and outliers. The semi-supervised setting without FixMatch, where the high confident pseudo inliers were included in the semi-supervised OpenMatch, provided a small 2.3% gain in accuracy, as shown in the third column of Table 5. It can be seen from Table 5 that adding the contrast regularization technique can effectively improve the performance of FixMatch.

Figure 12 shows t-SNE images after OpenMatch training. The more compact cluster structure of classes was a result of the semi-supervised learning of OpenMatch and contrastive regularization in Figures 12B,C, respectively.

Table 6 shows the confusion matrix for the best experimental performance, which featured semi-supervised OpenMatch with contrastive regularization. The unknown detection results included 95.5% recall and 91.3% precision, which was comparable with the two-head network. Similar to the results of the two-head network, leaf diseases such as angular leaf spots, powdery mildew, and gray mold fruit were confused with unknowns.

Figure 13 shows samples of recognition results. In Figure 13, all the TNs of leaf and fruit diseases were categorized as unknowns, similar to the results of the two-head network. Image quality was the primary reason for misclassification of FPs and TNs, as seen in Figure 13.

Conclusion

For continuous learning in the plant disease identification process, an unknown disease or condition should first be distinguished from a known disease. This paper examined with two different but related deep learning-based techniques the detection of unknown plant diseases, including OSR and OoD detection. We chose the two-head network using OoD detection and semi-supervised OpenMatch using OSR technology, which explicitly and implicitly assume outlier exposure, respectively.

We carefully review the two models, and performed modifications in order to compare their performance classifying known diseases as well as detection of unknown diseases. For the experiment, we built an image dataset of eight strawberry diseases. Experiments on the dataset show that assuming outlier exposure during training is helpful for detecting unknown diseases. The experimental results also demonstrated that a careful selection of OoD samples for training is important to achieve better performance. Additionally, we demonstrated that FixMatch in semi-supervised OpenMatch can be successfully added into a two-head network, with contrastive regularization, to improve performance. Both OoD detection and OSR provided reasonable and comparable performance, as they were more than 92% accurate classifying the eight strawberry diseases and detecting unknown diseases. We believe the methods used in our experiment are general in nature, allowing them to be effectively applied to any type of plant disease monitoring.

Data availability statement

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

Author contributions

KJ and JL: conceptualization and writing—original draft preparation. KJ, JY, and JL: methodology. KJ, JY, and U-OD: formal analysis and investigation. KJ, JL, and HK: writing—review and editing. JL and HK: resources. JL: supervision. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1A6A1A09031717) and the Brain Korea 21 PLUS Project, National Research Foundation of Korea and (NRF-2021R1A2C2006895) Project.

Acknowledgments

The authors thank all the authors cited in this article and the referees for their helpful comments and suggestions.

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.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.989086/full#supplementary-material

References

Aquil, M. A. I., and Ishak, W. H. W. (2021). Evaluation of scratch and pre-trained convolutional neural networks for the classification of tomato plant diseases. *IAES Int. J. Artif. Intell.* 10:467. doi: 10.11591/ijai.v10.i2.pp467-475

Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* 153, 46–53. doi: 10.1016/j.compag.2018.08.013

Bendale, A., and Boult, T. E. (2016). "*Towards open set deep networks*," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2016.173

Cardoso, D. O., França, F., and Gama, J. (2015). "A bounded neural network for open set recognition," in *In 2015 International Joint Conference on Neural Networks, IJCNN 2015*, (Killarney: IEEE). doi: 10.1109/IJCNN.2015.7280680

Chen, G., Peng, P., Wang, X., and Tian, Y. (2021). Adversarial reciprocal points learning for open set recognition. *arXiv* [Preprint]. arXiv: 2103.00953

CropNet (2020). Available online at: https://www.tensorflow.org/hub/tutorials/cropnet_cassava

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* [Preprint]. arXiv: 2010.11929

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

Ge, Z., Demyanov, S., Chen, Z., and Garnavi, R. (2017). Generative openmax for multi-class open set classification. *arXiv* [Preprint]. arXiv: 1707.07418

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*, (Las Vegas, NV: IEEE). doi: 10.1109/CVPR.2016.90

Hendrycks, D., and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv* [Preprint]. arXiv: 1610.02136

Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. (2020). "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Seattle, WA: IEEE). doi: 10.1109/CVPR42600.2020.01096

Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., et al. (2017). "Lifeclef 2017 lab overview: Multimedia species identification challenges," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, (Berlin: Springer). doi: 10.1007/978-3-319-65813-1_24

Joseph, K., Khan, S., Khan, F. S., and Balasubramanian, V. N. (2021). "Towards open world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Nashville, TN: IEEE). doi: 10.1109/CVPR46437.2021.00577

Kim, B., Han, Y.-K., Park, J.-H., and Lee, J. (2021). Improved vision-based detection of strawberry diseases using a deep neural network. *Front. Plant Sci.* 11:559172. doi: 10.3389/fpls.2020.559172

Kong, S., and Ramanan, D. (2021). "*Opengan: Open-set recognition via open data generation*," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (Piscataway, NJ: IEEE). doi: 10.1109/ICCV48922.2021.00085

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)*, (India: IEEE). doi: 10.1109/ICESC48915.2020.9155585

Lee, D., Kim, S., Kim, I., Cheon, Y., Cho, M., and Han, W.-S. (2022). Contrastive Regularization for Semi-Supervised Learning. *arXiv* [Preprint]. arXiv: 2201.06247

Liang, S., Li, Y., and Srikant, R. (2017). Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv* [Preprint]. arXiv: 1706.02690

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. doi: 10.1109/CVPR.2017.106

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, 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

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

Mukti, I. Z., and Biswas, D. (2019). "Transfer learning based plant diseases detection using ResNet50," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, (Bangladesh: IEEE). doi: 10.1109/EICT48899.2019.9068805

Open World Vision (2021). Available online at: http://www.cs.cmu.edu/~shuk/open-world-vision.html

OpenSetRecognition_list (2022). Available online at: https://github.com/iCGY96/awesome_OpenSetRecognition_list#papers

Oza, P., and Patel, V. M. (2019). "C2ae: Class conditioned auto-encoder for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2019.00241

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Netw.* 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Rangarajan, A. K., Purushothaman, R., and Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Comput. Sci.* 133, 1040–1047. doi: 10.1016/j.procs.2018.07.070

Rao, D. S., Ch, R. B., Kiran, V. S., Rajasekhar, N., Srinivas, K., Akshay, P. S., et al. (2022). Plant disease classification using deep bilinear cnn. *Intell. Autom. Soft Comput.* 31, 161–176. doi: 10.32604/iasc.2022.017706

Rehman, M. Z. U., Ahmed, F., Khan, M. A., Tariq, U., Jamal, S. S., Ahmad, J., et al. (2022). Classification of citrus plant diseases using deep transfer learning. *CMC Comput. Mater. Contin.* 70, 1401–1417. doi: 10.32604/cmc.2022.019046

Saito, K., Kim, D., and Saenko, K. (2021). OpenMatch: Open-set consistency regularization for semi-supervised learning with outliers. *arXiv* [Preprint]. arXiv: 2105.14148

Saleem, M. H., Potgieter, J., and Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants* 8:468. doi: 10.3390/plants8110468

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. (2021). A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv [Preprint]*. arXiv: 2110.14051

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2012). Toward open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1757–1772. doi: 10.1109/TPAMI.2012.256

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., et al. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inform. Process. Syst.* 33, 596–608.

Strawberry Diseases (2022). Available online at: https://diagnosis.ces.ncsu.edu/strawberry/

Tan, M., and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, (Long Beach, CA: PMLR).

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. (2021). Open-set recognition: A good closed-set classifier is all you need. *arXiv* [Preprint]. arXiv: 2110.06207

You, J., and Lee, J. (2020). Offline mobile diagnosis system for citrus pests and diseases using deep compression neural network. *IET Comput. Vis.* 14, 370–377. doi: 10.1049/iet-cvi.2018.5784

Yu, Q., and Aizawa, K. (2019). "Unsupervised out-of-distribution detection by maximum classifier discrepancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (Piscataway, NJ: IEEE). doi: 10.1109/ICCV.2019.00961

Zhang, H., Li, A., Guo, J., and Guo, Y. (2020). "*Hybrid models for open set recognition*," in *Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*, eds A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm (Cham: Springer).

TYPE Original Research
PUBLISHED 19 October 2022
DOI 10.3389/fpls.2022.914829

Appearance quality classification method of Huangguan pear under complex background based on instance segmentation and semantic segmentation

Yuhang Zhang[1†], Nan Shi[2†], Hao Zhang[1], Jun Zhang[1], Xiaofei Fan[1] and Xuesong Suo[1*]

[1]College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding, China, [2]Key Laboratory of Microbial Diversity Research and Application of Hebei Province, College of Life Sciences, Hebei University, Baoding, China

Check for updates

OPEN ACCESS

EDITED BY
Peng Chen,
Anhui University, China

REVIEWED BY
Tonghai Liu,
Tianjin Agricultural University, China
Sezai Ercisli,
Atatürk University, Turkey

*CORRESPONDENCE
Xuesong Suo
13903120861@163.com

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection,
a section of the journal Frontiers in Plant Science

RECEIVED 07 April 2022
ACCEPTED 04 October 2022
PUBLISHED 19 October 2022

CITATION
Zhang Y, Shi N, Zhang H, Zhang J, Fan X and Suo X (2022) Appearance quality classification method of Huangguan pear under complex background based on instance segmentation and semantic segmentation.
Front. Plant Sci. 13:914829.
doi: 10.3389/fpls.2022.914829

The 'Huangguan' pear disease spot detection and grading is the key to fruit processing automation. Due to the variety of individual shapes and disease spot types of 'Huangguan' pear. The traditional computer vision technology and pattern recognition methods have some limitations in the detection of 'Huangguan' pear diseases. In recent years, with the development of deep learning technology and convolutional neural network provides a new solution for the fast and accurate detection of 'Huangguan' pear diseases. To achieve automatic grading of 'Huangguan' pear appearance quality in a complex context, this study proposes an integrated framework combining instance segmentation, semantic segmentation and grading models. In the first stage, Mask R-CNN and Mask R-CNN with the introduction of the preprocessing module are used to segment 'Huangguan' pears from complex backgrounds. In the second stage, DeepLabV3+, UNet and PSPNet are used to segment the 'Huangguan' pear spots to get the spots, and the ratio of the spot pixel area to the 'Huangguan' pear pixel area is calculated and classified into three grades. In the third stage, the grades of 'Huangguan' pear are obtained using ResNet50, VGG16 and MobileNetV3. The experimental results show that the model proposed in this paper can segment the 'Huangguan' pear and disease spots in complex background in steps, and complete the grading of 'Huangguan' pear fruit disease severity. According to the experimental results. The Mask R-CNN that introduced the CLAHE preprocessing module in the first-stage instance segmentation model is the most accurate. The resulting pixel accuracy (PA) is 97.38% and the Dice coefficient is 68.08%. DeepLabV3+ is the most accurate in the second-stage semantic segmentation model. The pixel accuracy is 94.03% and the Dice coefficient is 67.25%. ResNet50 is the most accurate among the third-stage classification models. The average precision (AP) was 97.41% and the F1 (harmonic average assessment) was

95.43%.In short, it not only provides a new framework for the detection and identification of 'Huangguan' pear fruit diseases in complex backgrounds, but also lays a theoretical foundation for the assessment and grading of 'Huangguan' pear diseases.

KEYWORDS

'Huangguan' pear disease, deep convolutional neural networks, instance segmentation, semantic segmentation, disease severity classification

1 Introduction

Pears are fruits produced and consumed around the world, growing on a tree and harvested in the Northern Hemisphere in late summer into October. The pear tree and shrub are a species of genus Pyrus, in the family Rosaceae, bearing the pomaceous fruit of the same name (Ikinci et al., 2014). Several species of pears are valued for their edible fruit and juices, while others are cultivated as trees. China is the world's largest producer and consumer of pears, and its pear cultivation area and output rank first in the world (Oyom et al., 2022). 'Huangguan' pear is a mid-early mature pear variety cultivated by China. It has the advantages of large fruit size, high quality, early fruit, and good yield. It can meet the demand for high-quality pears in the fruit market. After years of demonstration and promotion, 'Huangguan' pear has become one of the main pear tree varieties in most regions, providing huge economic benefits for 'Huangguan' pear producers and exporting countries. It is worth emphasizing that the economic value of 'Huangguan' pear fruit depends to a large extent on the aesthetics of its appearance. The best-looking fruits are for export, the less diseased ones are reserved for domestic consumption, and the worst ones are used for further processing to make canned fruits or jams. However, the quality grading of 'Huangguan' pear is a time-consuming and laborious process. So far, it has almost completely relied on human inspection and manual observation of disease symptoms to judge the grade of 'Huangguan' pear. This method is costly and has highly subjective and low efficiency and timeliness. However, early automated grading systems have extensively utilized image processing algorithms and relied on manually defined image features to build classifiers (Suykens, 2001; Zeng et al., 2020), limiting the robustness and generalization (Xu and Mannor, 2012) of detection performance due to the variance of fruits types, appearances, and damage defects.

In recent years. With the advancement of agricultural informatization, deep learning and machine learning are widely used different areas in agriculture (Dobrota et al., 2021; Yang et al., 2022a)in particular in crop disease detection (Liu et al., 2018; Pooja et al., 2018; Yu et al., 2018; Özden, 2021; Tassis et al., 2021; Wang et al., 2021; Wang et al., 2022; Yang et al., 2022b), experts and scholars have achieved fruitful results in the research of plant disease identification. Wu et al. (2020b) used Mask R-CNN and VGG models to judge whether 6400 mango images are good or bad, with an accuracy rate of 83.6% and expressed by PCA. Ren et al. (2020) used the tomato plant diseases in the Plant Village data set and the improved VGG to propose a model that can identify tomato leaf diseases, with an accuracy rate of more than 95%.In the food industry, a model based on CNN was introduced for identification of soft-shell shrimp. The proposed model attained an average accuracy of 97% (Liu, 2020). Ireri et al. (2019) introduced a tomato grading machine vision system. The proposed system performed calyx and stalk scar detection for both defected and healthy tomatoes based on regions of interest. The radial basic function support vector machine classifier achieved 97.09% accuracy rate for healthy and defected tomatoes. Farooq and Sazonov (2017) used CNN to classify different food groups. The classification accuracy for 7 and 61 different classes was 94.01% and 70.13%, respectively. Liang et al. (2019) proposed a plant disease severity estimation network PD2SENet, which achieves excellent comprehensive performances. Lu et al. (2017) developed an application for diagnosing diseases in wheat leaves using two steps: a disease location step and a classification step. Wu et al. (2020a) proposed an automatic and efficient apple defect identification method based on laser-induced light backscatter imaging and convolutional neural network algorithm. Sofu et al. (2016) proposed an automatic apple sorting and quality inspection system that apples were sorted into different classes by their color, size and weight. It also detected apples affected by scab, stain and rot. The average grading accuracy rate is 73–96%. Wang et al. (2017) applied 5 convolutional neural networks with different structures to estimate the severity of plant diseases, and fine-tuned the existing network models using transfer learning to improve the model accuracy. The above research has used traditional machine learning or deep learning to identify plant diseases, but the refinement and generalization capabilities need to be improved. Although some progress has been made in the research of fruit disease segmentation under complex background, the research of 'Huangguan' pear has not made significant progress. Convolutional neural networks have led to a series of breakthroughs for image classification (He et al., 2015). This article uses a convolutional neural network (CNN) to automatically extract data features by introducing local connections, pooling and other operations. In the first step, the strength segmentation model was used to remove the background of 'Huangguan' pear, which ensured the fineness of the next step of grading (He et al., 2018). Then, the semantic segmentation model was used to segment the disease of 'Huangguan' pear, and the proportion of disease pixels in 'Huangguan' pear was calculated (Chen et al., 2018). Finally, by performing transfer learning on ImageNet data. The over-fitting problem caused by the small sample data domain was optimized and the grading model was used to achieve the quality grading of 'Huangguan' pear (Akiba et al., 2017). Experiments show that this method can not only improve the recognition accuracy of 'Huangguan' pear disease, but also is suitable for classification of 'Huangguan' pear disease images in generalized scenarios.

The main contributions of this research are as follows:

For complex background images, a two-stage segmentation model of 'Huangguan' pear disease based on deep learning was proposed. The model achieved accurate segmentation of 'Huangguan' pear and disease. It provided the basis for establishing the classification model of 'Huangguan' pear disease severity.

By adopting a three-stage continuous segmentation and classification method, the complementary advantages of Mask R-CNN, DeepLabV3+ and Resnet50 models are fully utilized. Compared with the single-stage model, this model has better segmentation and classification effects.

A method for grading the severity of 'Huangguan' pear disease was proposed. By calculating the ratio of the area of diseased spots to the area of 'Huangguan' pear fruit, it provides technical support for the accurate classification of the appearance quality of 'Huangguan' pear in actual production.

It can effectively solve the problem of inaccurate grading of 'Huangguan' pears caused by manual sorting, which is time-consuming and laborious and easy to distract. It provides a new idea for the automatic grading of 'Huangguan' pear appearance quality.

2 Materials and methods

2.1 Data set production and processing

The data set used in this article has a total of 5562 images of 'Huangguan' pear. Taking into account the diversity of lighting conditions in practical applications, The data was collected in three different periods from July to December 2021: In the morning (8:30–10:00), noon (12:30–14:00) and afternoon (15:30–17:00) in the laboratory with camera. This leads to problems such as background noise, distance, location, and lighting conditions of 'Huangguan' pear. It is the existence of these problems that can improve the generalization ability of the model in different scenarios and improve the robustness of the model. Part of the 'Huangguan' pear image is shown in Figure 1. According to the 'Huangguan' pear samples displayed in the data set, the identification and segmentation of 'Huangguan' pear fruit disease mainly have the following difficulties: 1) 'Huangguan' pear background interferes with segmentation, and the different brightness of 'Huangguan' pear imaging caused by factors such as light can easily be mistaken for disease; 2) 'Huangguan' pear disease are irregular in shape, some are small, and the initial disease are difficult to detect with the naked eye, which increases the difficulty of disease segmentation; 3) 'Huangguan' pear have different shooting

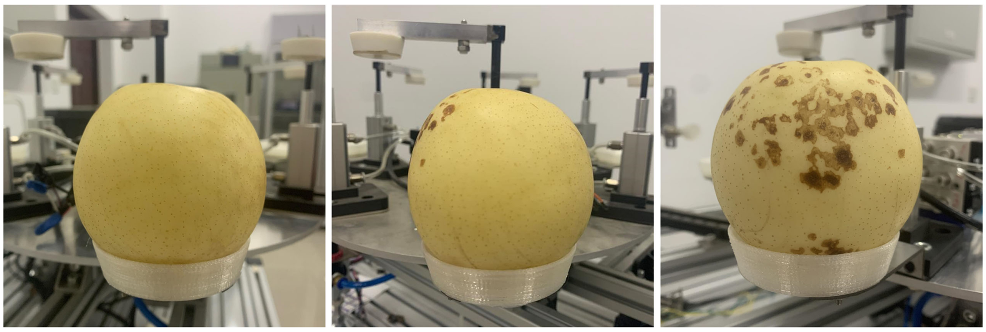


FIGURE 1
Some pictures of 'Huangguan' pear.

backgrounds, and the quality of the background processing directly affects the classification of 'Huangguan' pear.

2.2 Image data enhancement

The sample distribution of each type of disease in the data set is not uniform, and the limited training data is easy to overfit the deep learning model. In deep learning, the use of data augmentation methods to expand the data can improve the generalization ability of the model. The training data of this study uses the Image Data Generator online enhancement method under the Keras framework. That is, an enhancement method is randomly selected for each batch of data during the training process, without increasing the number of original data sets. In order to avoid changing the original data characteristics and better simulate the differences of samples under real shooting conditions, the training set of this research mainly adopts the following data enhancement methods: 1) Flip: Flip the image vertically to simulate the randomness of the shooting angle when the sample is collected, and will not change the shape of the diseased spot and the distribution of the diseased spot on the leaf. 2) Color jitter: Change the brightness of the image to randomly jitter between 0.8-1.2 times. Change the contrast of the image to randomly jitter between 0.6-1.6 times. Change the chromaticity of the image to jitter randomly between 0.7-1.4 times. Simulate lighting differences and ensure that the parameters conform to the actual shooting conditions to avoid image distortion. 3) Add noise: Add salt and pepper noise with a signal-to-noise ratio of 0.95 to the image to simulate the noise generated during the shooting process and weaken the high-frequency features to prevent the model from overfitting. The result of data enhancement is shown in Figure 2.

2.3 Labeling of diseased spots of 'Huangguan' pear fruit

To train the disease segmentation model, the disease need to be marked as shown in Figure 3. The labeling of 'Huangguan' pear disease is time-consuming and laborious, with a large number of small targets. The finer annotations help Mask R-CNN and DeepLabV3+ to perform finer segmentation of 'Huangguan' pears and disease, laying the foundation for the classification of 'Huangguan' pears. The labeling is divided into three scenes including background, pear and diseased spots, and labeling is carried out with LabelMe (Russell et al., 2008), an image semantic segmentation labeling tool.

2.4 Grading method for the severity of fruit diseases of 'Huangguan' pear

The classification of disease severity is the basis for formulating prevention and control strategies. Three methods are usually used in practice. The first method is to calculate the ratio of the number of infected fruits per unit area to the total number of fruits. The second method is to calculate the ratio of the number of diseased fruits to the total number of fruits on the same plant. The third method is to calculate the ratio of the area

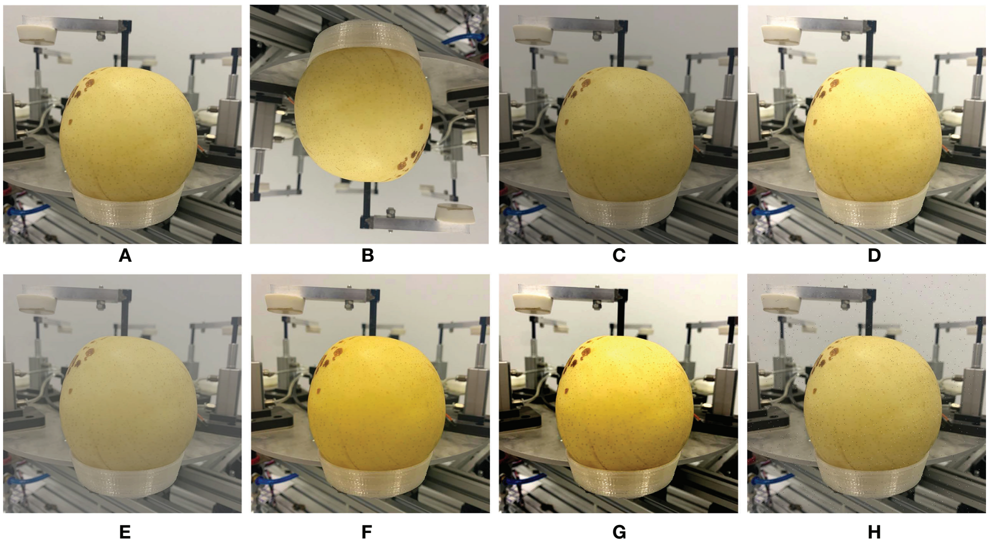


FIGURE 2
The image enhancement of (A) Original image, (B) Vertical flip, (C) 0.8 Brightness, (D) 1.2 Brightness, (E) 0.6 Contrast, (F) 1.6 Contrast, (G) Change chroma, and (H) Add salt noise.

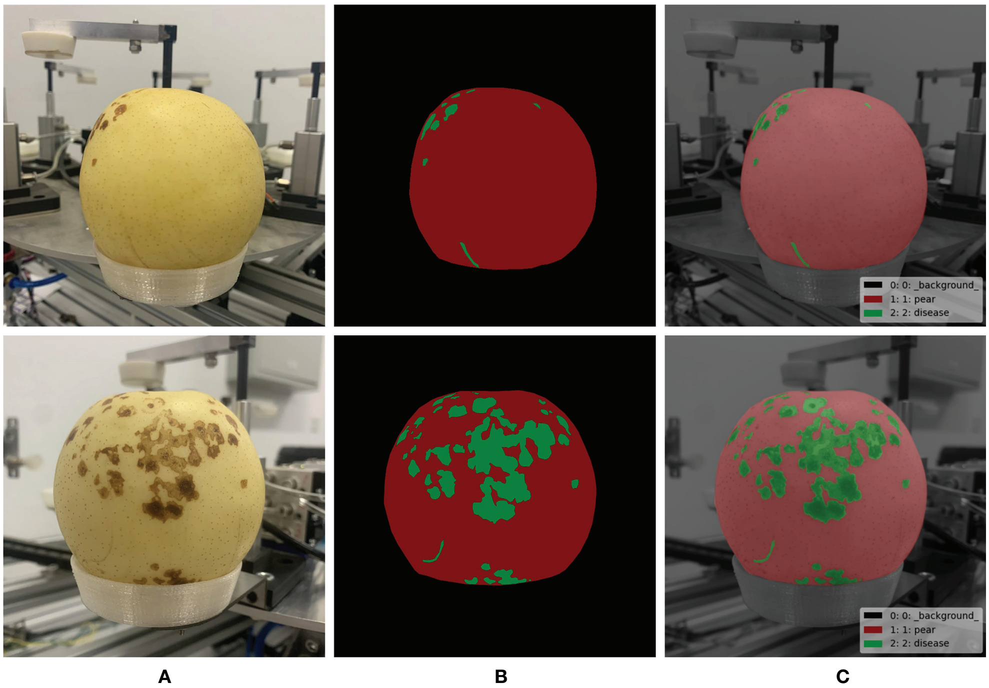


FIGURE 3
Pixel level labeling of 'Huangguan' pear fruit and disease spot. **(A)** Original images, **(B)** Pear-disease-labels and **(C)** Composite diagrams.

of spots on the same fruit to the total area of the fruit. The third method is the basis for accurately estimating the severity of crop diseases in a region. Therefore, we used the third method, which uses the ratio of the spot area to the total area of the same fruit as the basis for classification of disease severity. This method is mainly based on the opinions and practical experience of fruit farmers who have been engaged in fruit grading for many years.

By calculating the ratio of the area of the diseased spot to the area of the fruit, the severity of the disease of 'Huangguan' pear was classified. Since the 'Huangguan' pear fruit to be divided is located in a complex background, the target 'Huangguan' pear fruit and diseased spots are easily confused with other similar elements, resulting in over-segmentation or under-segmentation. Therefore, it is difficult to accurately segment 'Huangguan' pear fruit and diseased spots at the same time using a single-stage network. In order to ensure the accuracy of disease segmentation, the 'Huangguan' pear fruit in the complex background should be segmented first. Therefore, this study uses a two-stage segmentation network to classify the severity of 'Huangguan' pear diseases, and classifies the 'Huangguan' pear images according to the first, second and third levels. Specific steps are as follows. In the first stage, the segmentation target is the 'Huangguan' pear fruit and the complex background. The mask image obtained from the test is used to extract the 'Huangguan' pear fruit from the complex background, so as to obtain the 'Huangguan' pear fruit in the simple background. In the second stage of segmentation, the diseased spots in the 'Huangguan' pear fruit are taken as the target, and the proportion of the diseased spots in the 'Huangguan' pear fruit is obtained. As the basis for the classification of disease severity of 'Huangguan' pear. The formula is shown in formula (1).

$$P = \frac{S_{Disease}}{S_{Pear}} \tag{1}$$

Among them, S_{Pear} represents the fruit area of 'Huangguan' pear after segmentation; S_{Disease} represents the area of the disease after segmentation; P represents the proportion of diseased spots on 'Huangguan' pear fruit.

After calculating the area of 'Huangguan' pear by the disease, refer to the 'Huangguan' Pear Fruit Grade" DB 13/T 1571—2012 issued by China. According to local standards, the proportion of fruit diseases can be divided into three grades: good and bad. Among them, 0% of diseases are first-class fruits, 2% or less are second-class fruits, and diseases greater than 2% are third-class fruits.

2.5 Evaluation index

In order to reasonably evaluate the performance of the model, the first two segmentation stages of this study used 3 commonly used evaluation indicators: Pixel Accuracy (PA), dice and Intersection over Union (IoU). The pixel accuracy is the ratio of all correctly classified pixels to the total pixels, as shown in formula (2):

$$R_{PA} = \frac{\Sigma_{i=0}^{k} p_{ii}}{\Sigma_{i=0}^{k}\Sigma_{j=0}^{k} p_{ij}} \tag{2}$$

In the formula, k is the number of categories, p_{ii} is the number of pixels that are correctly predicted, and p_{ij} represents the number of pixels whose category i is predicted to be category j

The Dice coefficient is a function that measures the similarity of two sets, and is one of the most commonly used evaluation indicators in semantic segmentation. As shown in formula (3):

$$R_{dice} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3}$$

Where X is the predicted pixel and Y is the ground truth.

The intersection ratio is the ratio of the intersection and union of a certain type of prediction result and the true value of the model. The intersection ratio is the most commonly used evaluation index in semantic segmentation, and the expression is shown in formula (4):

$$R_{IoU} = \frac{A \cap B}{A \cup B} \tag{4}$$

When the value of IOU is between 0 and 1, it represents the degree of overlap of the two boxes. The higher the value, the higher the degree of overlap.

The third grading stage uses 5 evaluation indicators commonly used in grading models, recall, precision, average precision (AP), F1 score and speed. Recall is the ratio of the number of correctly detected targets to all actual targets (Equation (5)). Precision is the number of correctly detected targets in all detected targets The ratio of (Equation (6)). F1 is the harmonic average of precision and recall (Equation (7)).

$$\mathrm{Re}\,call(R) = \frac{TP}{TP + FN} \tag{5}$$

$$\mathrm{Pr}\,ecision(P) = \frac{TP}{TP + FP} \tag{6}$$

$$F1 = 2 \times \frac{\mathrm{Pr}\,e \times \mathrm{Re}\,c}{\mathrm{Pr}\,e + \mathrm{Re}\,c} \tag{7}$$

2.6 Model training

The hardware configuration used for training and testing in this research is as follows: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 16G RAM, NVIDIA GeForce GTX 1650SUPER graphics card, 64-bit Windows 10 operating system, CUDA version 10.0 and TensorFlow version 1.13.2. In order to avoid the influence of hyperparameters on the experimental results, the hyperparameters of each network are uniformly configured. After trial and error, the following hyperparameters have been determined: The learning rate is 1e-4, the epochs is 50, and the batch size is 16. If training for more than 5 generations does not further improve the accuracy, start early stopping and stop training.

3 Model construction

3.1 Input layer

The input image is a color 3-channel image of leaf disease, and the image size is uniformly adjusted to 416x416 pixels. In order to enhance the generalization ability of the model, a data enhancement method is randomly selected during the training process to process the original image, and the normalized and standardized data is used as the input of Mask R-CNN.

3.2 Model Mask R-CNN

Mask R-CNN is a new convolutional neural network proposed by Ren et al. (2015) based on Faster R-CNN, which realizes instance segmentation. This method can not only detect the target effectively, but also complete high-quality semantic segmentation of the target. The main idea is to add a branch to the original Faster R-CNN to achieve semantic segmentation of the target. Mask R-CNN uses FPN to improve the feature extraction network, which better solves the problem of serious loss of semantic information through the feature extraction layer of FCN and SegNet (Kendall et al., 2015), and greatly improves the segmentation of small target defects. For Deeplab-v3 defect contour segmentation is not clear, Mask R-CNN replaces the interest area pooling layer with the interest area alignment layer. That is, the spatial information on the feature map is further utilized through bilinear interpolation, so as to predict a more accurate defect contour. Mask R-CNN first uses the FPN based on Resnet50 to extract the feature map of the defect image, and then uses RPN to generate the target suggestion box, And use the Soft-NMS algorithm to filter the ROI (Bodla et al., 2017), and finally perform category prediction, bounding box prediction, and target binarization mask for each ROI. The structure of Mask R-CNN is shown in Figure 4.

3.3 Semantic segmentation model

DeepLabV3+ first uses Xception feature extraction network to perform feature extraction on the original image (Chollet, 2016), and then introduces several parallel Atos convolutions at different rates to obtain larger-scale image feature information. Then use

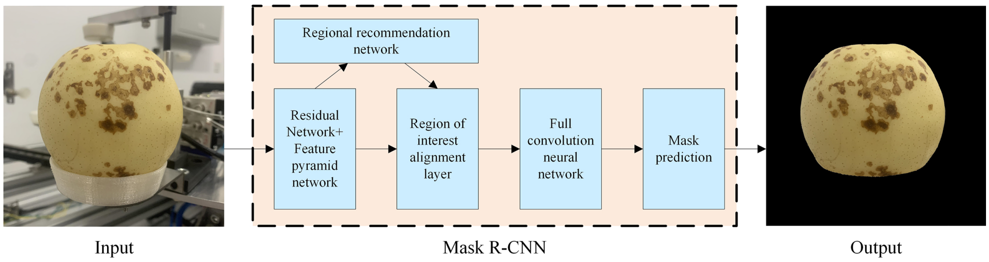

Regional recommendation network
Residual Network+ Feature pyramid network
Region of interest alignment layer
Full convolution neural network
Mask prediction
Input
Mask R-CNN
Output

FIGURE 4
The processing flow of 'Huangguan' pear by Mask R-CNN network.

the spatial pyramid pooling module Atrous Spatial Pyramid Pooling(ASPP) (Chen et al., 2016), respectively use a variety of different void rates for extraction. Obtain more semantic feature information, thereby improving segmentation accuracy. The Encode-Decode structure is the mainstream structure in the semantic segmentation network (Badrinarayanan et al., 2015). The so-called encoding process is to extract the features of the substation equipment through the feature extraction network, and then reorganize the feature information through decoding. In this process, the network is based on the image The label information is constantly modified parameters, and finally the object semantic segmentation of supervised learning is realized. The depth separable convolution can be added to the ASPP and decoder modules to make the overall model more efficient.

The UNet proposed by (Ronneberger et al., 2015) for semantic segmentation of biomedical images consists of two stages: a contraction stage and an expansion stage. The shrinking stage consists of the FCN architecture, including convolution, ReLU, and pooling operations. This step is responsible for extracting features from the image. The second step, also known as the expansion step, is the opposite of the previous step. It consists of a series of deconvolution operations followed by convolution and concatenation of the feature maps obtained in the first step. The last part of the network reconstructs the segmented image.

PSPNet (Zhao et al., 2016) adopts a spatial pyramid network architecture, which not only enhances the fusion of multi-scale information, but also reduces the local and global losses. This structure is a network architecture that integrates multi-scale scenarios, including 2 parts of convolutional layer and pyramidal pooling, which has multiple advantages, not only simple architecture but also high flexibility. Among them, the convolutional layer integrates different classical network architectures to achieve a progressive abstraction from low-level to high-level features.

3.4 Classification model

He et al. (2015) proposed resnet50 network. The main contribution is to solve the problem of the decline in classification accuracy as the depth of CNN deepens. The proposed residual learning idea accelerates the CNN training process and effectively avoids the problem of gradient disappearance and gradient explosion. Using the idea of residual learning. He et al. (2015) proposed a Shortcut Connections structure of identity mapping, as shown in Figure 5. Where *X* is the input, *F(X)* is the residual mapping, *Y(X)* is the ideal mapping, *Y(X)* = *F(X)* + *X*. By transforming the fitted residual mapping *F(X)* into the fitting ideal mapping *Y(X)*, the output can be changed into the superposition of the input and the residual mapping, so that the network changes between the input *X* and the output More sensitive. It does not add additional parameters and calculations to the network, but at *Y*

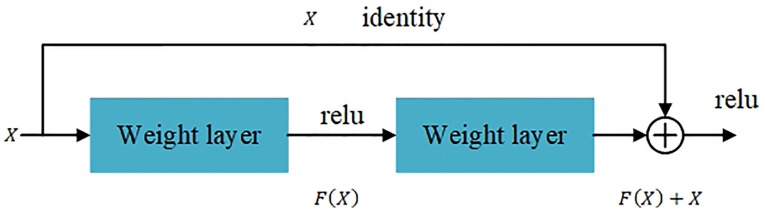


FIGURE 5
The residual block.

(X) the same time greatly increases the training speed of the model and improves the training effect. When the number of layers of the model is deepened, this simple structure can well solve the degradation problem. In recent years, the ResNet network has been widely cited in various computer vision tasks, and has achieved outstanding performance. So this article chooses ResNet50 as the third stage hierarchical network model.

The essence of the VGG16 model is an enhanced version of the AlexNet structure, with an emphasis on the depth of the CNN design (Simonyan and Zisserman, 2014). Furthermore, each convolutional layer is followed by a pooling layer. VGG16 has five convolutional layers, each with two or three convolutional layers. To better extract feature information, this experiment uses three convolutional layers per segment. In addition, VGG16 uses 3x3 convolution kernels instead of 7x7 convolution kernels. The 3x3 convolution kernel is the smallest receptive field size that can feel the focus of up and down, left and right. And, 2 3x3 convolution kernels are stacked. Their receptive field is equivalent to a 5x5 convolution kernel. When 3 stacks, their receptive field is equivalent to a 7x7 effect. Since the receptive field is the same, three 3x3 convolutions use three nonlinear activation functions to increase the nonlinear expression ability. Makes the dividing plane more separable. At the same time, a small convolution kernel is used, which greatly reduces the amount of parameters. Using the 3x3 convolution kernel stacking form not only increases the number of network layers but also reduces the amount of parameters. Due to the large number of layers and the relatively small convolution kernel, the entire network has better feature extraction effect.

MobileNetV3 replaces part of the 3×3 depth wise convolution by introducing a 5×5 depth wise convolution (Howard et al., 2019). Introduce Squeeze-and-excitation (SE) module and h-swish (HS) activation function to improve model accuracy. The last two layers of pointwise convolution do not use batch normalization. Use the NBN logo in the MobileNetV3 structure diagram. MobileNetV3 combines the following advantages. The first point is the depth wise separable convolution of MobileNetV1. The second point is the inverse residual structure of MobileNetV2 with a linear bottleneck. The third point is to use h-swish instead of the swish function.

3.5 Three-stage model structure

Different instance segmentation, semantic segmentation and classification models have different network structures, which will affect the classification accuracy of 'Huangguan' pear and disease. If the same model is used in the three stages, the feature extraction ability of the model may be affected due to different segmentation targets. Therefore, based on the different features to be extracted at each stage, compare various semantic segmentation and hierarchical models to determine a better model for each stage. Then, combined with the actual environment's requirements for segmentation speed, by adjusting the order of the model and changing the feature extraction network, segmentation accuracy can be improved and segmentation time can be shortened.

In this research, the fusion instance, semantic segmentation and classification network were used to segment the fruits and lesions of the 'Huangguan' pear in two stages through multiple experiments. Because the single-stage segmentation model of 'Huangguan' pear in complex background is difficult to accurately segment the fruit and lesions of 'Huangguan' pear at the same time, its segmentation accuracy is generally low. Based on the above ideas, through the comparison of multiple instance segmentation models, semantic segmentation models and hierarchical models. Finally, it is determined that the Mask R-CNN network with a preprocessing module is used to segment the 'crown' pear in the complex background in the first stage, and the image of the 'crown' pear in the simple background can be obtained. Then, the 'Huangguan' pear was segmented by DeepLabV3+, and the disease rate of the 'Huangguan' pear was calculated. Finally, use ResNet50 for training. The overall flow chart is shown in Figure 6.

4. Test results and analysis

4.1 Accuracy and effect of 'Huangguan' pear background segmentation

The models used in the first stage of this article are CLAHE-MASK R-CNN and Mask R-CNN. By default, the file with the best training effect will be saved as a weight file and then used for testing. In the algorithm of this paper, the CLAHE preprocessing module is added according to the characteristics of the 'Huangguan' pear background, which improves the local contrast of the edge of the 'Huangguan' pear and improves the network's ability to predict the details of the mask boundary. At the same time, fewer convolutional layers can ensure that the edge of the 'Huangguan' pear target will not be lost after multi-layer convolution. It can be seen from Figure 7. That CLAHE-Mask R-CNN can segment the background other than 'Huangguan' pear under the same label picture. Under the same conditions, when the segmented background color is similar to 'Huangguan' pear, Mask R-CNN gets the wrong result. The red box marks the background that Mask R-CNN has not segmented completely or the background is segmented excessively. In this paper, the Mask R-CNN of the CLAHE module has a better effect on the edge segmentation of 'Huangguan' pear. If the accuracy of the first stage segmentation is not high, it may result in segmentation of the wrong 'Huangguan' pear in the second stage, and the final accuracy will be reduced. For comprehensive comparison, CLAHE-Mask R-CNN is selected as the first stage segmentation model. It can be seen from Table 1 that

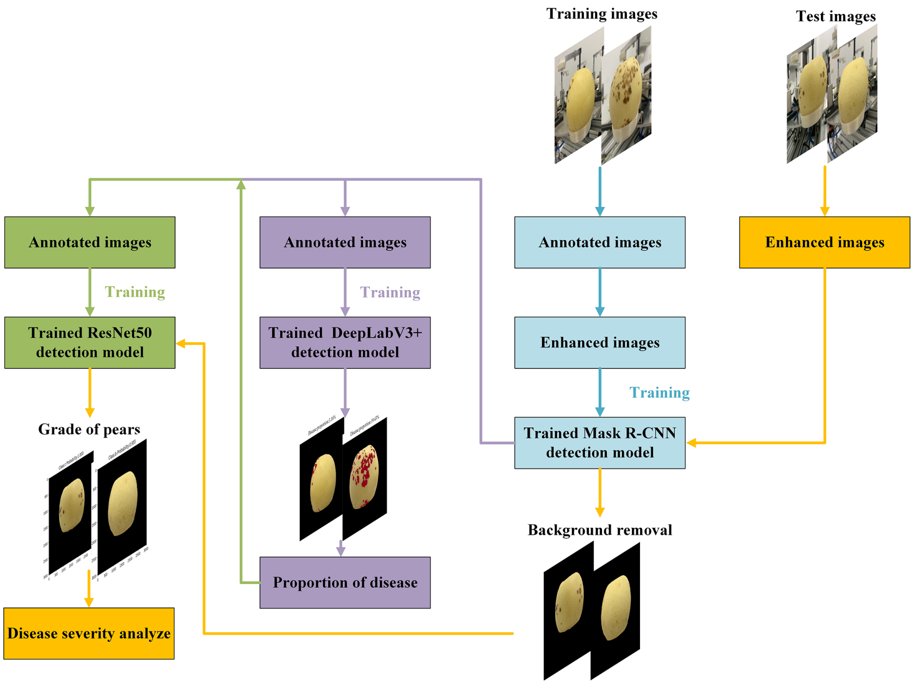


FIGURE 6
Three-stage model network architecture.

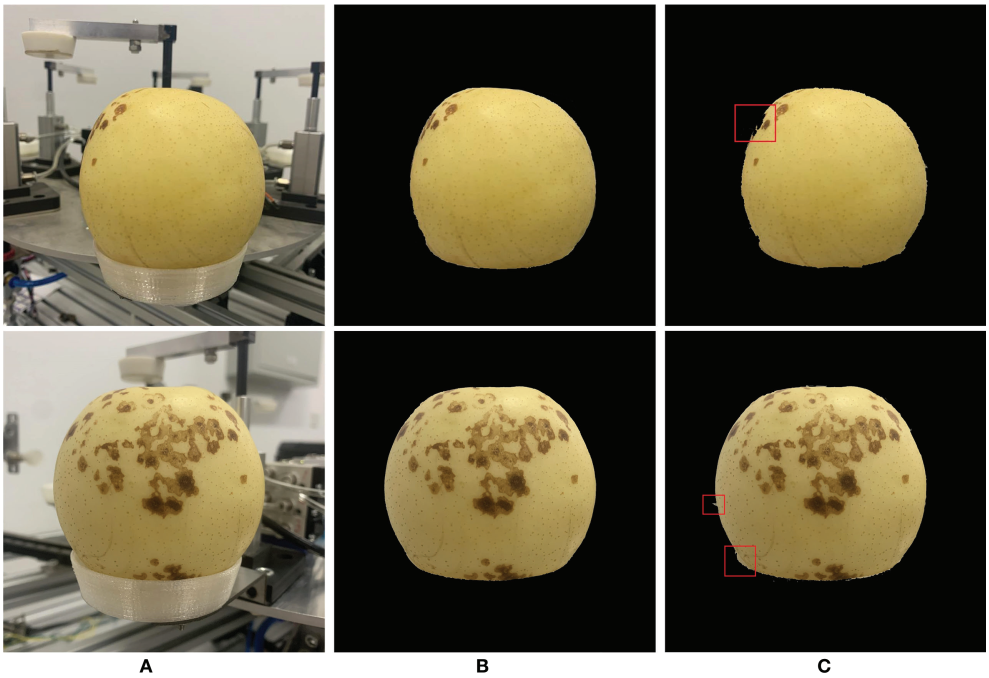


FIGURE 7
The prediction result of 'Huangguan' pear instance segmentation. **(A)** Original images, **(B)** CLAHE-Mask R-CNN and **(C)** Mask R-CNN.

TABLE 1 Performance of the first stage model on the test set.

Model	PA/%	Dice/%	IoU/%
CLAHE-Mask R-CNN	97.38	68.08	73.25
Mask R-CNN	94.84	67.72	69.92

in the first stage, the segmentation accuracy of Mask R-CNN with the addition of the CLAHE module is significantly higher than that of Mask R-CNN. The PA of CLAHE-Mask R-CNN reaches 97.38%, which can better provide 'Huangguan' pear pictures with complex background removed for the next stage and increase the accuracy of the overall model.

4.2 Comparison of segmentation accuracy and effect of 'Huangguan' pear disease

The overall structure of the semantic segmentation model used in the second stage is shown in Figure 8. Three models DeepLabV3+, UNet, and PspNet were used to segment 'Huangguan' pear disease. Divide the area of 'Huangguan' pear diseased spots by the area of 'Huangguan' pear to get the proportion of diseased spots, which provides accurate data support for the third-step classification model.

In the semantic segmentation stage, 448 images of 'Huangguan' pear were used as test samples, and the labels were only divided into 'Huangguan' pear and diseased spots without considering the disease category. The test result is the average of the test results of 448 images. Table 2 shows the comparison results of the segmentation accuracy of each algorithm. It can be seen from Table 2 that the segmentation accuracy of DeepLabV3+ is significantly higher than that of UNet and PspNet. The accuracy of DeepLabV3+ reached 94.03%. Compared with UNet and PspNet, the accuracy has increased by 2.81% and 0.62%. At the same time, the disease segmented by DeepLabV3+ obtained higher Dice coefficient (0.6725) and IoU coefficient (0.7436). Compared with UNet, it increased by 2.68% and 7.21%, and compared with PspNet by 0.86% and 3.25%. Various segmentation results are shown in Figure 9.

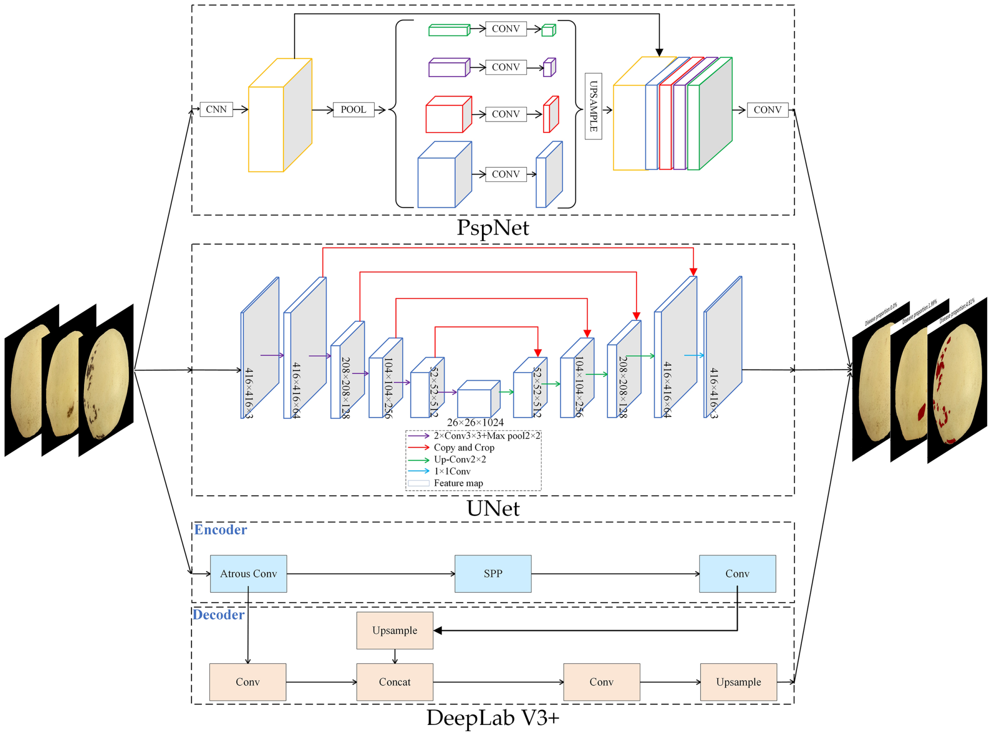


FIGURE 8
'Huangguan' pear semantic segmentation network structure diagram.

TABLE 2 Performance of the second stage model on the test set.

Model	PA/%	Dice/%	IoU/%
DeepLabV3+	94.03	67.25	74.36
UNet	91.22	64.57	67.15
PspNet	93.41	66.39	71.11

It shows the segmentation results of 'Huangguan' pear disease on the DeepLabV3+, UNet, and PspNet models. It can be seen that DeepLabV3+ can segment small disease. The segmentation result of UNet will lose some details, the segmentation boundary will be fuzzy, the similar disease area will be stuck, and the segmentation edge of UNet will appear jagged and there will be edge loss. This is because UNet cannot capture features at different levels, and integrates them through feature superposition. It is easy to lose data due to repeated downsampling and upsampling of the deep network. The convolution operation of the encoder-decoder of DeepLabV3+ can smoothly segment the edges of disease. The segmentation edge of PspNet is relatively smooth, but it is easy to miss some disease areas and excessive segmentation of disease areas. Which means that PspNet does not have obvious response to disease with similar colors to 'Huangguan' pear. It can be seen from the segmentation map that the difficulty of segmentation for different disease is different. For example, the chicken feet disease area of 'Huangguan' pear is dark yellow and the color is similar to that of 'Huangguan' pear, and the edge of the disease is not obvious, so the segmentation is more difficult. DeepLabV3 + can arbitrarily control the resolution of the extracted features of the encoder, and can effectively and accurately segment the 'Huangguan' pear disease by balancing the accuracy and time-consuming hole convolution. The proportion of diseased spots in 'Huangguan' pears is shown in Figure 10.

The DeepLabV3+ model is used to predict the disease area of 'Huangguan' pear, and the predicted disease area and the actual disease area have a higher IoU. It benefits from the early pixel-level disease labeling and the introduction of hole convolution in DeepLabV3+, which has strong semantic segmentation performance. According to the ratio of the number of pixels of

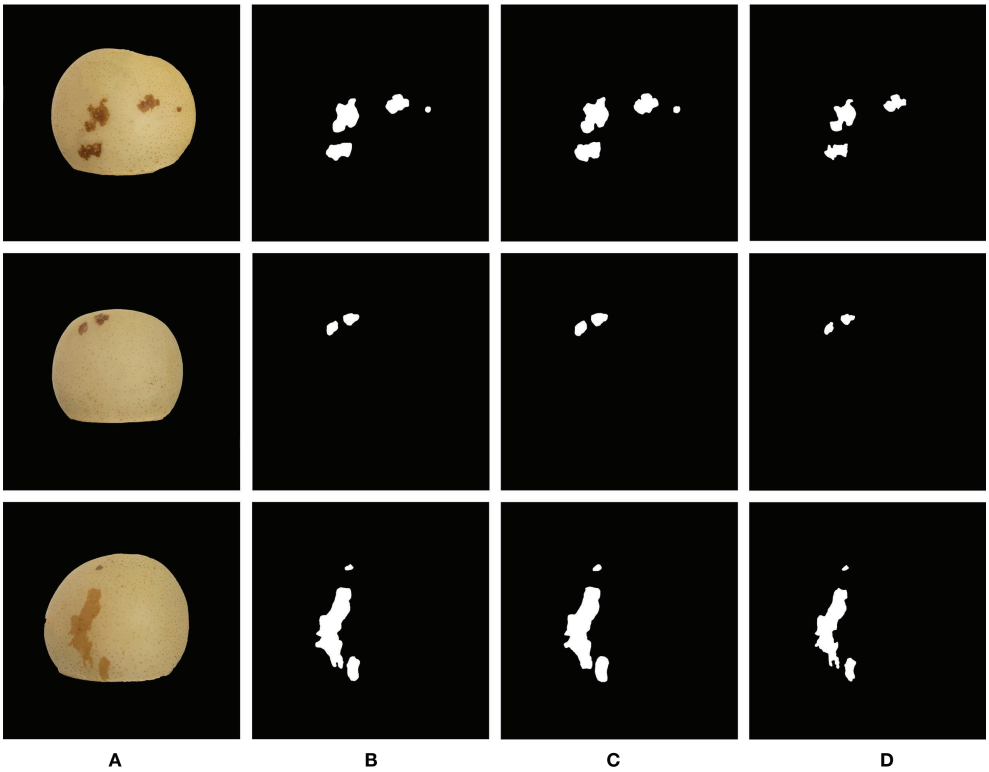


FIGURE 9
Comparison of the segmentation results of 'Huangguan' pear disease. (A) Original Image, (B) DeepLabV3+, (C) UNet and (D) PspNet.

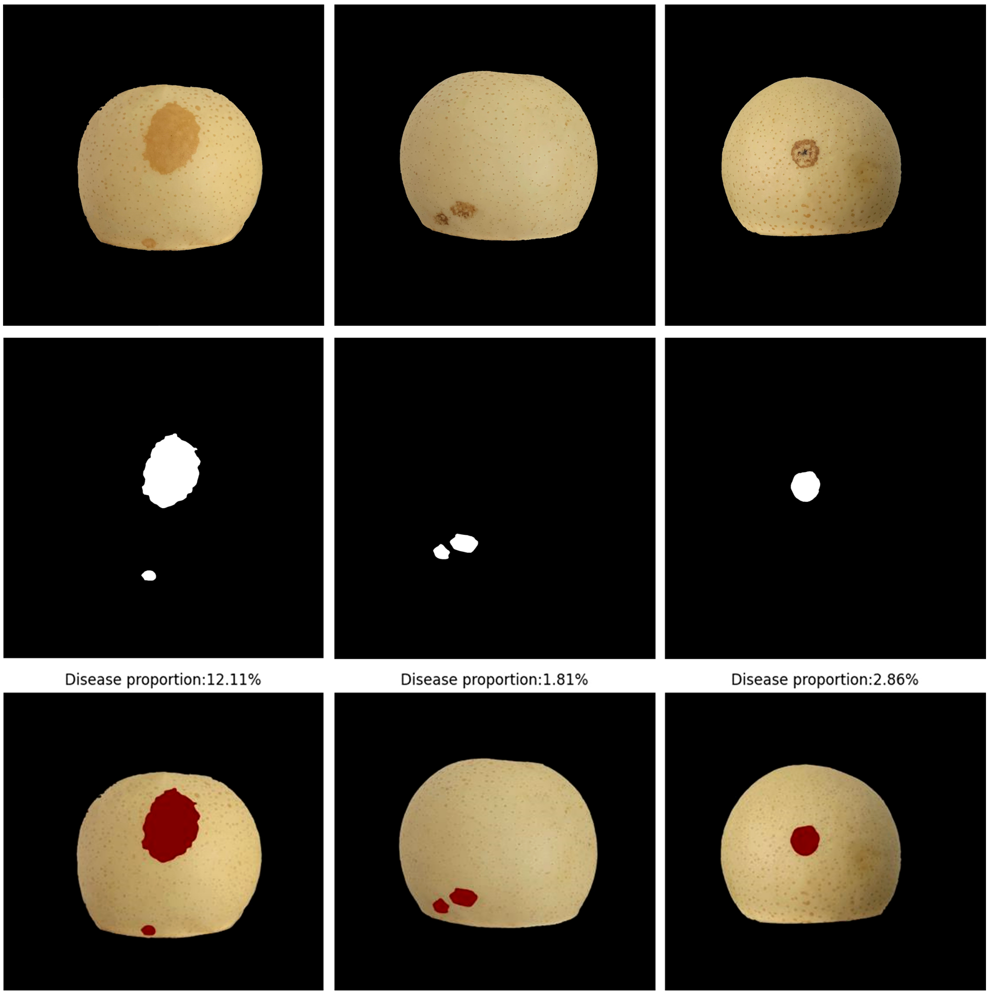


FIGURE 10
Proportion of 'Huangguan' pear disease.

the diseased spots to the number of pixels of 'Huangguan' pear, the accurate ratio of the diseased spots can be obtained, which provides an accurate data set for the third stage model.

4.3Analysis of the classification results of 'Huangguan' pear

4.3.1 Loss function

The fully connected layer uses the gradient descent algorithm as the parameter optimizer, and sets the average cross entropy as the loss function as follows:

$$L = \frac{1}{N}\Sigma_i L_i = -\frac{1}{N}\Sigma_{i=1}^{M} y_i \ln(p_i) \tag{8}$$

(8) Where: N is the total number of samples; M is the number of categories; y_i is the indicator variable (0 or 1), if the category is i, it is 1, otherwise it is 0; p_i is the probability that the observed sample is i; L_i Represents the loss value of category i.

4.3.2 Training process

The data set is classified according to the disease grades segmented by DeepLabV3+. There are a total of 5114 images in the training set and the verification set, which are allocated at a ratio of 9:1. There are 448 images in the test set. When training the classification model, three test models are designed: ResNet50, VGG16 and MobileNetV3. Among them, the classification of 'Huangguan' pear image is shown in Table 3.

Taking the ResNet50 model training as an example, first use a part of the third-class fruits in the image divided into 50 evenly and use equation (9) to train for one round, which can guide the network to pay attention to the disease part when extracting features. Then, after training 30 batches of samples without disease, use the training set with disease for one round to ensure

TABLE 3 Grade distribution of 'Huangguan' pear.

Dataset split	A	Grade B	C
Training set	1264 (27.46%)	1516 (32.95%)	1822 (39.59%)
Validation set	140 (27.35%)	169 (33.00%)	203 (39.65%)
Test set	126 (28.13%)	130 (29.02%)	192 (42.85%)

continuous supervision of the results of disease. An epoch training will be completed until all training samples with no disease are finished. At the end of each epoch training, record the training accuracy and average loss. Use the model trained in this round to make a prediction for all test samples, and record the test accuracy and average loss. After training for 150 epochs, the weight with the smallest disease recognition loss on the test set is selected as the final model.

4.3.3 Analysis of training results

Use VGG16 model, ResNet50 and MobileNetV3 respectively for training, and ensure that the parameter settings are the same. Because each iteration randomly uses an image enhancement method, the training recognition accuracy will fluctuate slightly. In the first 5 rounds of training, the training and recognition accuracy of ResNet50 increased rapidly, and the recognition accuracy on the test set reached more than 95% earlier than other models. ResNet50 has the highest recognition accuracy in the first round of testing. When the number of iteration rounds is about 60 rounds, the training recognition accuracy of ResNet50 first tends to 100%. It can be seen from the change of recognition accuracy that the ResNet50 model can converge faster, and its training accuracy and loss rate are shown in Figure 11.

Accurate and efficient 'Huangguan' pear appearance quality classification model is of great significance. The automatic scoring method will alleviate the problem of rural labor shortage. In addition, an accurate grading model will indirectly affect market segments and ensure the reliable and stable quality of 'Huangguan' Pear agricultural products. As shown in Table 4, the above experimental results clearly show the effectiveness of the ResNet50 model on the 'Huangguan' pear appearance quality classification model. The ResNet50 algorithm maintains a fairly high accuracy. The results show that this method can be used to realize the automatic grading of the appearance quality of 'Huangguan' pear. In our experiments, the ResNet50 model takes about 311.2 milliseconds to predict the appearance quality of each 'Huangguan' pear, and there is not much difference between VGG16 and MobileNetV3. This speed can fully meet the real-time requirements of classification. Compared with VGG16 and MobileNetV3, the average precision of ResNet50 has a higher advantage, which is 11.61% and 4.94% higher respectively. The prediction result of 'Huangguan' pear grade is shown in Figure 12.

It can be seen that the prediction result of ResNet50 on the appearance quality of 'Huangguan' pear is relatively accurate, and the prediction level and prediction probability are marked directly above each picture. It can be seen that the ResNet50 model can predict the 'Huangguan' pear images with different light intensity well, and the model has high robustness.

5. Conclusion and future work

In conclusion, this research proposes a three-stage model of 'Huangguan' pear disease in complex contexts that combines instance segmentation, semantic segmentation, and classification. In the first stage, the complete 'Huangguan' pear

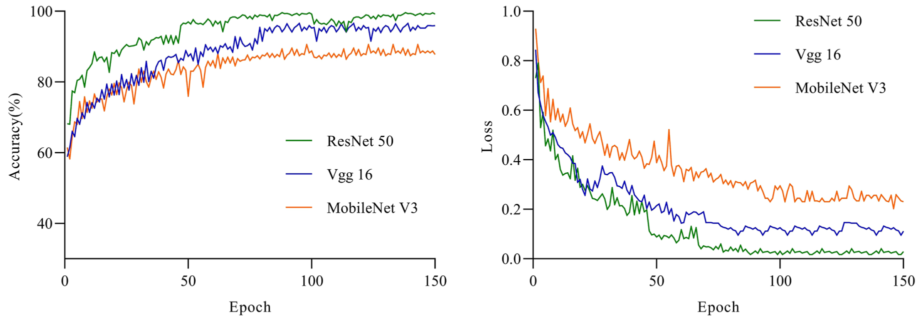


FIGURE 11
The accuracy and loss of the three models for the test set.

TABLE 4 Comparison results of ‘Huangguan’ pear grading data sets.

Model	Classes	Precision/%	Recall/%	AP/%	F1/%	Speed/ms
ResNet50	A	99.58	97.25	97.41	95.43	311.2
	B	95.26				
	C	97.39				
VGG16	A	88.21	85.54	85.80	86.10	430.6
	B	83.42				
	C	85.78				
MobileNetV3	A	95.97	93.68	92.47	92.88	198.5
	B	89.28				
	C	92.15				

fruit is segmented and extracted using Mask R-CNN with an preprocessing module. Then in the second stage, DeepLabV3+ was used to segment and extract the diseases of the simple background ‘Huangguan’ pear fruit extracted in the first stage, and the proportion was calculated. Through the data obtained in the second stage, the ‘Huangguan’ pears are divided into three grades: A, B, and C. In the third stage, the weights are obtained by training three grades of fruits through ResNet50. In the prediction stage, after the Mask R-CNN segmentation is completed, the ResNet50 model is used for prediction, and the grade of the ‘Huangguan’ pear can be directly obtained. Overall, the model can improve the accuracy of disease segmentation, thereby providing a reasonable classification opinion for the disease severity of ‘Huangguan’ pear fruits. Finally, the pixel accuracy of the Mask R-CNN model with preprocessing module is 97.38%. The pixel accuracy of the DeepLabV3+ model is 94.03%. The average precision of the ResNet50 model is 97.41%. Overall segmentation and classification performance is significantly improved compared to the one-stage model. This method based on machine vision and deep learning is harmless to ‘Huangguan’ pears and provides technical support for follow-up research. Currently, all diseases are roughly graded. ‘Huangguan’ pears suffer from a wide variety of diseases. The next step will be to subdivide the disease of the ‘Huangguan’ pear. Detect and identify various types of diseases to assess their severity. Thereafter, further work in this direction will continue.

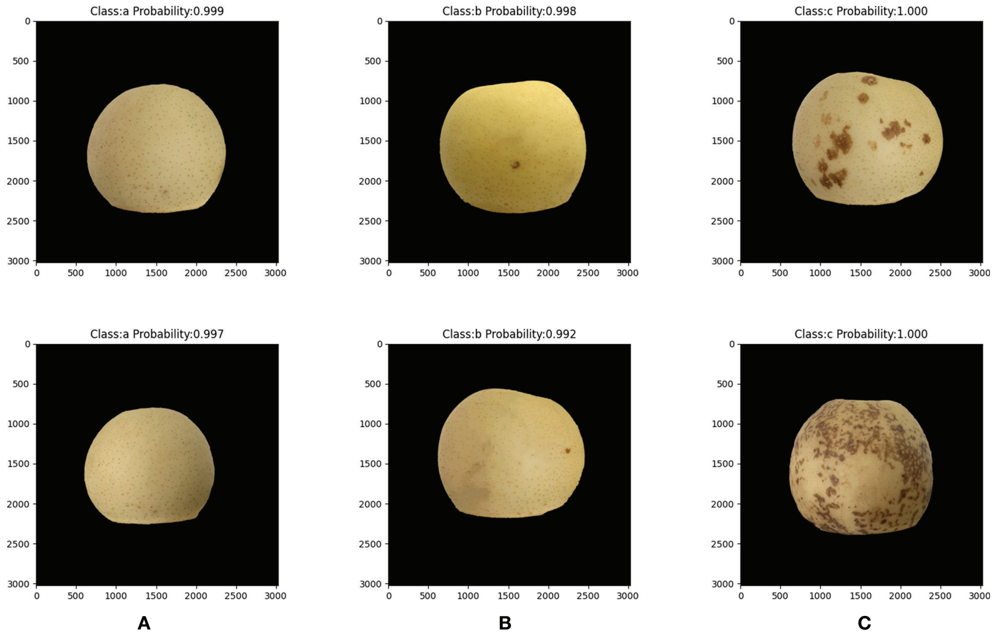

Class:a Probability:0.999
Class:b Probability:0.998
Class:c Probability:1.000
Class:a Probability:0.997
Class:b Probability:0.992
Class:c Probability:1.000
0
500
1000
1500
2000
2500
3000
A
B
C

FIGURE 12
Prediction grade results of ‘Huangguan’ pear. **(A)** grade A, **(B)** grade B and **(C)** grade C.

Data availability statement

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

Author contributions

YZ: Writing-original draft. NS: Guiding, Supervision. HZ: Data collection. JZ: Proofreading and polish manuscript. XF and XS: Editing, Supervision, Proofreading. All authors contributed to the article and approved the submitted version.

Funding

This study was supported by the Key Research and Development Program of Hebei Province (20327213D).

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

Akiba, T., Suzuki, S., and Fukuda, K. (2017) *Extremely Large minibatch SGD: Training ResNet-50 on ImageNet in 15 minutes*. doi: 10.48550/arXiv.1711.04325.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015) *SegNet: A deep convolutional encoder-decoder architecture for image segmentation*. doi: 10.48550/arXiv.1511.00561.

Bodla, N., Singh, B., Chellappa, R., and Davis, L. S. (2017) *Soft-NMS – improving object detection with one line of code*. doi: 10.48550/arXiv.1704.04503.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016) *DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs*. doi: 10.48550/arXiv.1606.00915.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018) *Encoder-decoder with atrous separable convolution for semantic image segmentation*. doi: 10.48550/arXiv.1802.02611.

Chollet, F. (2016) *Xception: Deep learning with depthwise separable convolutions*. doi: 10.48550/arXiv.1610.02357.

Dobrota, C. T., Carpa, R., and Butiuc-Keul, A. (2021). Analysis of designs used in monitoring crop growth based on remote sensing methods. *Turkish J. Agric. Forestry* 45, 730–742. doi: 10.3906/tar-2012-79

Farooq, M., and Sazonov, E. (2017). "Feature extraction using deep learning for food type recognition," in *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Springer Verlag), 464–472. doi: 10.1007/978-3-319-56148-6_41

He, K., Dollar, P., Girshick Presenters, R., Wang, X., and Shi, M. (2018). *Mask r-CNN*. 2017 IEEE International Conference on Computer Vision (ICCV). doi: 10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. Available at: http://arxiv.org/abs/1512.03385.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019) *Searching for MobileNetV3*. doi: 10.48550/arXiv.1905.02244

Ikinci, A., Bolat, I., Ercisli, S., and Kodad, O. (2014). Influence of rootstocks on growth, yield, fruit quality and leaf mineral element contents of pear cv. "Santa maria" in semi-arid conditions. *Biol. Res.* 47, 1–8. doi: 10.1186/0717-6287-47-71

Ireri, D., Belal, E., Okinda, C., Makange, N., and Ji, C. (2019). A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. *Artif. Intell. Agric.* 2, 28–37. doi: 10.1016/j.aiia.2019.06.001

Kendall, A., Badrinarayanan, V., and Cipolla, R. (2015) *Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding*. doi: 10.48550/arXiv.1511.02680

Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., and Sun, W. (2019). PD 2 SE-net: Computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron Agric.* 157, 518–529. doi: 10.1016/j.compag.2019.01.034

Liu, Z. (2020). Soft-shell shrimp recognition based on an improved AlexNet for quality evaluations. *J. Food Eng.* 266, 1–10. doi: 10.1016/j.jfoodeng.2019.109698

Liu, B., Zhang, Y., He, D. J., and Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry (Basel)* 10, 1–16. doi: 10.3390/sym10010011

Lu, J., Hu, J., Zhao, G., Mei, F., and Zhang, C. (2017). An in-field automatic wheat disease diagnosis system. *Comput. Electron Agric.* 142, 369–379. doi: 10.1016/j.compag.2017.09.012

Oyom, W., Li, Y., Prusky, D., Zhang, Z., Bi, Y., and Tahergorabi, R. (2022). Recent advances in postharvest technology of Asia pears fungi disease control: A review. *Physiol. Mol. Plant Pathol.* 117, 72–82. doi: 10.1016/j.pmpp.2021.101771

Özden, C. (2021). Apple leaf disease detection and classification based on transfer learning. *Turkish J. Agric. Forestry* 45, 775–783. doi: 10.3906/tar-2010-100

Pooja, V., Das, R., and Kanchana, V. (2018). "Identification of plant leaf diseases using image processing techniques," in Proceedings - 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development, TIAR 2017 (Institute of Electrical and Electronics Engineers Inc.). 130–133. doi: 10.1109/TIAR.2017.8273700

Ren, S., He, K., Girshick, R., and Sun, J. (2015) *Faster r-CNN: Towards real-time object detection with region proposal networks*. doi: 10.48550/arXiv.1506.01497

Ren, S., Jia, F., Gu, X., Yuan, P., Xue, W., and Xu, H. (2020). Recognition and segmentation model of tomato leaf diseases based on deconvolution-guiding. *Nongye Gongcheng Xuebao/Transactions Chin. Soc. Agric. Eng.* 36, 186–195. doi: 10.11975/j.issn.1002-6819.2020.12.023

Ronneberger, O., Fischer, P., and Brox, T. (2015) *U-Net: Convolutional networks for biomedical image segmentation*. doi: 10.48550/arXiv.1505.04597

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision* 77, 157–173. doi: 10.1007/s11263-007-0090-8

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

Sofu, M. M., Er, O., Kayacan, M. C., and Cetişli, B. (2016). Design of an automatic apple sorting system using machine vision. *Comput. Electron Agric.* 127, 395–405. doi: 10.1016/j.compag.2016.06.030

Suykens, J. A. K. (2001). Support vector machines: A nonlinear modelling and control perspective. *Eur. J. Control* 7, 311–327. doi: 10.3166/ejc.7.311-327

Tassis, L. M., Tozzi de Souza, J. E., and Krohling, R. A. (2021). A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images. *Comput. Electron Agric.* 186, 1–12. doi: 10.1016/j.compag.2021.106191

Wang, C., Du, P., Wu, H., Li, J., Zhao, C., and Zhu, H. (2021). A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-net. *Comput. Electron Agric.* 189, 1–13. doi: 10.1016/j.compag.2021.106373

Wang, L., Liu, J., Zhang, J., Wang, J., and Fan, X. (2022). Corn seed defect detection based on watershed algorithm and two-pathway convolutional neural networks. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.730190

Wang, G., Sun, Y., and Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Comput. Intell. Neurosci.* 2017, 1–9. doi: 10.1155/2017/2917536

Wu, A., Zhu, J., and Ren, T. (2020a). Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Comput. Electrical Eng.* 81, 1–9. doi: 10.1016/j.compeleceng.2019.106454

Wu, S.-L., Tung, H.-Y., and Hsu, Y.-L. (2020b) Deep learning for automatic quality grading of mangoes: Methods and insights. doi: 10.48550/arXiv.2011.11378

Xu, H., and Mannor, S. (2012). Robustness and generalization. *Mach. Learn* 86, 391–423. doi: 10.1007/s10994-011-5268-1

Yang, J., Lan, G., Li, Y., Gong, Y., Zhang, Z., and Ercisli, S. (2022a). Data quality assessment and analysis for pest identification in smart agriculture. *Comput. Electrical Eng.* 103, 1–11. doi: 10.1016/j.compeleceng.2022.108322

Yang, J., Yang, Y., Li, Y., Xiao, S., and Ercisli, S. (2022b). Image information contribution evaluation for plant diseases classification *via* inter-class similarity. *Sustainability* 14, 10938. doi: 10.3390/su141710938

Yu, X., Lu, H., and Wu, D. (2018). Development of deep learning method for predicting firmness and soluble solid content of postharvest korla fragrant pear using Vis/NIR hyperspectral reflectance imaging. *Postharvest Biol. Technol.* 141, 39–49. doi: 10.1016/j.postharvbio.2018.02.013

Zeng, X., Miao, Y., Ubaid, S., Gao, X., and Zhuang., S. (2020). Detection and classification of bruises of pears based on thermal images. *Postharvest Biol. Technol.* 161, 1–6. doi: 10.1016/j.postharvbio.2019.111090

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2016)Pyramid scene parsing network. doi: 10.48550/arXiv.1612.01105.

Frontiers in Plant Science

TYPE Original Research
PUBLISHED 20 October 2022
DOI 10.3389/fpls.2022.1023924

A wheat spike detection method based on Transformer

Qiong Zhou[1,2,3†], Ziliang Huang[1,2†], Shijian Zheng[1,4], Lin Jiao[1,5*], Liusan Wang[1*] and Rujing Wang[1,2*]

[1]Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, China, [2]Science Island Branch, University of Science and Technology of China, Hefei, China, [3]College of Information and Computer, Anhui Agricultural University, Hefei, China, [4]Department of Information Engineering Southwest, University of Science and Technology, Mianyang, China, [5]School of Internet, Anhui University, Hefei, China

OPEN ACCESS

EDITED BY
Po Yang,
The University of Sheffield,
United Kingdom

REVIEWED BY
Liu Liu,
Shanghai Jiao Tong University, China
Le Zou,
Hefei University, China

*CORRESPONDENCE
Rujing Wang
rjwang@iim.ac.cn
Liusan Wang
lswang@iim.ac.cn
Lin Jiao
ljiao@ahu.edu.cn

[†]These authors have contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

RECEIVED 20 August 2022
ACCEPTED 22 September 2022
PUBLISHED 20 October 2022

CITATION
Zhou Q, Huang Z, Zheng S, Jiao L, Wang L and Wang R (2022) A wheat spike detection method based on Transformer. *Front. Plant Sci.* 13:1023924. doi: 10.3389/fpls.2022.1023924

Wheat spike detection has important research significance for production estimation and crop field management. With the development of deep learning-based algorithms, researchers tend to solve the detection task by convolutional neural networks (CNNs). However, traditional CNNs equip with the inductive bias of locality and scale-invariance, which makes it hard to extract global and long-range dependency. In this paper, we propose a Transformer-based network named Multi-Window Swin Transformer (MW-Swin Transformer). Technically, MW-Swin Transformer introduces the ability of feature pyramid network to extract multi-scale features and inherits the characteristic of Swin Transformer that performs self-attention mechanism by window strategy. Moreover, bounding box regression is a crucial step in detection. We propose a Wheat Intersection over Union loss by incorporating the Euclidean distance, area overlapping, and aspect ratio, thereby leading to better detection accuracy. We merge the proposed network and regression loss into a popular detection architecture, fully convolutional one-stage object detection, and name the unified model WheatFormer. Finally, we construct a wheat spike detection dataset (WSD-2022) to evaluate the performance of the proposed methods. The experimental results show that the proposed network outperforms those state-of-the-art algorithms with 0.459 mAP (mean average precision) and 0.918 AP_{50}. It has been proved that our Transformer-based method is effective to handle wheat spike detection under complex field conditions.

KEYWORDS

deep learning, IoU loss function, transformer, wheat spike detection, agriculture

1 Introduction

Wheat is one of the most important food crops in the world, with an annual production of 730 million tons in around 215 million ha (Catherine et al., 2014). As the global yield supports approximately 30% of the world population, wheat production estimation has become a focus of agricultural research. It could provide key indicators for agricultural decision-making and field management. Since wheat spike is a major factor that reflects the grain number per unit area, it is significant to accurately detect the wheat spike for estimating crop yield.

Traditional field yield estimation methods are time-consuming, inefficient, and poorly representative, so they are not suitable for current large-scale yield forecasting tasks. With the development of computer vision, many researchers have conducted research through machine learning techniques. Fang et al. (2020) proposed to estimate the wheat tiller density based on terrestrial laser scanning data. Fernandez-Gallego et al. (2019) used zenithal/nadir thermal images to count the number of wheat spikes. Jin et al. (2017) adopted unmanned aerial vehicles (UAVs) to obtain high-resolution imagery for estimating wheat plant density. In these traditional machine learning studies, image texture, geometry, and color intensity are primarily used to discriminate spikes. However, the process is partly manually designed to define the range and threshold in the model. They are not robust enough for different situations with dense distribution, complex structural environments, and severe occlusion in the field (Zhang et al., 2020a). Convolutional neural networks (CNNs) have been introduced into the research of wheat spike detection in recent studies. Khoroshevsky et al. (2021) suggested that a network incorporates multiple targets in a single deep model, and the results show that the method is effective as a yield estimator. Misra et al. (2020) combined digital image analysis with CNN techniques to identify and count wheat spikes. CNNs are effective to extract local information, but they lack the ability to extract long-range features from global information. Due to the field environment of wheat being complex, *i.e.*, dense distribution, complex structural environment, and severe occlusion, it is hard for CNNs to perform well.

The evolution of Transformer (Vaswani et al., 2017) in natural language processing (NLP) provides an alternative path, and many researchers have subsequently transferred the NLP models to computer vision models. Compared with conventional CNN backbones, Transformers always produce global receptive fields rather than local receptive fields, which is more suitable for detecting objects in complex backgrounds. The Transformer architecture avoids repetition and instead relies entirely on the attention mechanism to map the global dependencies between inputs and outputs. The significant success in the natural language processing domain motivates researchers to investigate the application in classification (Dosovitskiy et al., 2021) and dense prediction tasks (Bochkovskiy et al., 2020; Carion et al., 2020; Xizhou et al., 2020). There are two main challenges in transferring the NLP Transformer to the visual domain Transformer. Firstly, unlike the word tokens that are the basic elements of a linguistic Transformer, the vision elements can be very different from the NLP in scale. Another is that Transformer has high computational and memory costs for prediction tasks.

Bounding box regression is a key operation to locate the target object in detection tasks. The loss function is to calculate the difference between the regression result and the true value and finally minimize the regression error. The $ln\text{−}norm$ loss function is widely adopted in bounding box regression, while the common $ln\text{−}norm$ loss (e.g. $l1\text{−}norm$ or $l2\text{−}norm$) is used for measuring the distance between bounding boxes. However, according to the research of Yu et al. (Yu et al., 2016; Rezatofighi et al, 2019), it is not tailored to the Intersection over Union (IoU) metric. IoU loss (Yu et al., 2016) and generalized IoU (GIoU) loss (Rezatofighi et al., 2019) have recently been suggested to improve the IoU metric. IoU loss can be effective only when the bounding boxes overlap, but it is useless for non-overlapping cases. GIoU adds a penalty term that the predicted bounding box will move to the target box without overlapping. Nevertheless, GIoU empirically has a lower convergence speed, and it will degrade to IoU loss for enclosing boxes (Zheng et al., 2020). Therefore, it is important to design an effective loss function for bounding box regression.

In this work, we aim to explore a Transformer-based network for wheat spike detection. To the best of our knowledge, this is the first attempt using Transformer in the wheat detection field. Inspired by the novel architecture of Swin Transformer (Liu et al., 2021) and exploring to overcome the above-mentioned limitations, we propose a Transformer-based network named MW-Swin Transformer. It has the following advantages: Firstly, compared with the conventional Transformer, the proposed Transformer occupies the hierarchical architecture that is essential for downstream tasks. Secondly, compared with Swin Transformer, we inherit the excellent network and design of a multi-window Transformer block to extract target features with different scales. Thirdly, our method has three variants according to the number of stacked layers, which is flexible to fit the actual requirements. Furthermore, we propose a WIoU loss for bounding box regression. Specifically, we add a penalty term on IoU loss, considering the overlap area, Euclidean distance, and aspect ratio. The three geometric indicators are important, *e.g.*, the Euclidean distance is used to minimize the distance of central points in two bounding boxes, and the consistency of aspect ratios is also bringing about an impact on IoU loss. We incorporate the proposed methods into the FCOS and name the new model WheatFormer, as illustrated in Figure 1. WheatFormer contains two major parts: the multi-window Swin (MW-Swin) Transformer and the wheat detector. The input image is split into non-overlapping patches, and each

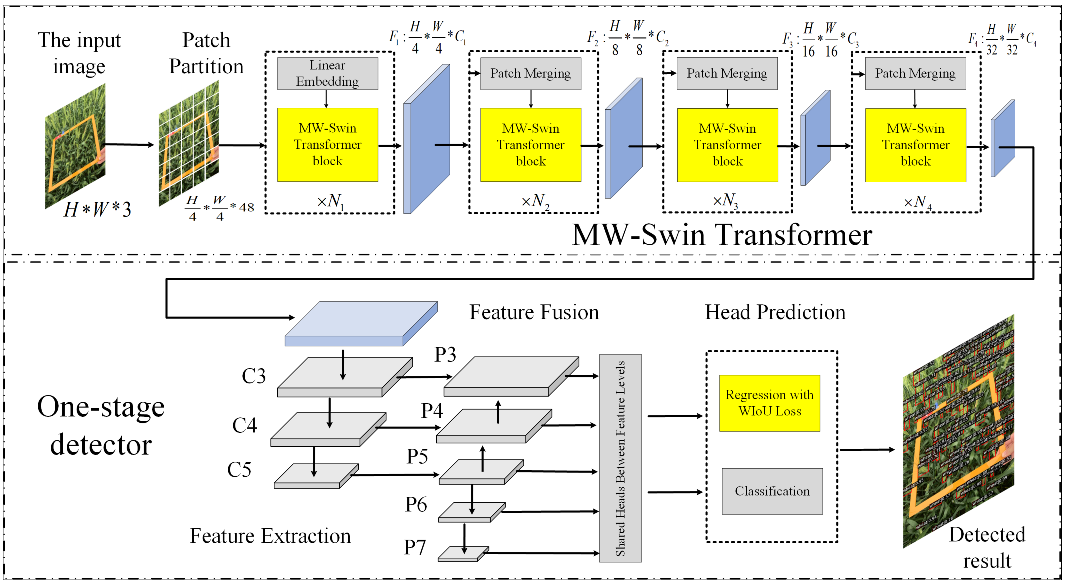

The input image
Patch Partition
$H*W*3$
$\frac{H}{4}*\frac{W}{4}*48$
Linear Embedding
MW-Swin Transformer block
$\times N_1$
$F_1:\frac{H}{4}*\frac{W}{4}*C_1$
Patch Merging
MW-Swin Transformer block
$\times N_2$
$F_2:\frac{H}{8}*\frac{W}{8}*C_2$
Patch Merging
MW-Swin Transformer block
$\times N_3$
$F_3:\frac{H}{16}*\frac{W}{16}*C_3$
Patch Merging
MW-Swin Transformer block
$\times N_4$
$F_4:\frac{H}{32}*\frac{W}{32}*C_4$
MW-Swin Transformer
One-stage detector
Feature Fusion
Head Prediction
C3
C4
C5
P3
P4
P5
P6
P7
Feature Extraction
Shared Heads Between Feature Levels
Regression with WIoU Loss
Classification
Detected result

FIGURE 1
The main architecture of WheatFormer.

patch is regarded as a token and fed into the MW-Swin Transformer backbone to learn long-range features from global information. Then, the extracted feature maps are fed into the one-stage detector to locate the wheat spike. Finally, we construct a wheat spike detection dataset named WSD-2022 to evaluate the performance of the proposed WheatFormer. The dataset contains 6,404 images from two data sources, the first was from the Global Wheat Head Detection (GWHD) dataset (David et al., 2021) and the second was collected in the field environment by our collaborators. The major contributions of our work are as follows:

- We propose the MW-Swin Transformer with multiple windows for different scale objects, which inherits from the shifted windows in Swin Transformer. This strategy brings a much lower latency than those previous Transformer models, leading to strong performance due to the global receptive field.
- A WIoU loss function is proposed for bounding box regression, considering three important geometric indicators. WIoU helps the network achieve a better performance than normal IoU loss and other improved IoU loss functions.
- We build the WSD-2022 dataset for detecting wheat spikes. This dataset contains wheat spike images from different regions and different developmental stages. Our work provides a richer benchmark dataset for wheat spike detection tasks.

2 Related work

2.1 CNN-based methods in wheat spike detection

CNNs have been widely used in computer vision tasks, such as image classification (Huang et al., 2017), object detection (Ren et al., 2017), and semantic segmentation (He et al., 2017), which have achieved excellent achievements. Differently from traditional machine learning methods, CNNs can automatically abstract features without manual intervention. Sadeghi-Tehran et al. (2019) proposed a low-computational-cost system to automatically detect the number of wheat spikes, which used simple linear iterative clustering with CNN. Hasan et al. (2018) introduced a robust R-CNN model for the accurate detection, counting, and analysis of wheat ears for yield estimation. Wang et al. (2019) provided a method based on a fully convolutional network and Harris corner detection, solving the problem of counting wheat ears in field conditions. Madec et al. (2019) used Faster R-CNN to provide accurate ear density using RGB images taken from the UAV. Pound et al. (2017) investigated a deep learning method capable of accurately localizing wheat ears and spikelets. Gong et al. (2020) proposed a novel object method of wheat head detection based on dual SPP networks to enhance the speed and accuracy of detection. Yang et al. (2021) combined the convolutional neural network and attention mechanism technology to propose a CBAM-YOLOv4 wheat ear detection and counting method.

2.2 Object detection

Object detection methods can be divided into two groups: with two stages and with one-stage. For two-stage detectors, the first stage is to produce lots of high-quality region proposals by a proposal generator, and the second stage is classifying and refining the proposals by region-wise subnetworks. R-CNN (Girshick et al., 2014) and Fast R-CNN (Girshick, 2015) are the typical networks of two-stage detectors, which combined the region proposals and CNN for object detection. Faster R-CNN (Ren et al., 2017) was proposed to speed up Fast R-CNN and promote detection accuracy by using region proposal network. Other two-stage detectors mainly include Mask R-CNN (He et al., 2020), Libra R-CNN (Pang et al., 2019), and Cascade R-CNN (Cai and Vasconcelos, 2018). However, two-stage detectors show a weakness in detection efficiency (Redmon et al., 2016). For one-stage detectors, they drop the process of generation region proposals, treating the object detection task as a single shot problem, such as the YOLO series networks: YOLO (Redmon et al., 2016), YOLOv3 (Redmon and Farhadi, 2018), and YOLOv4 (Bochkovskiy et al., 2020). Tian et al. (2019) proposed a fully convolutional one-stage object detector. This method avoided the complex computation by eliminating the predefined set of region proposals. SSD (Fu et al., 2017) introduced additional context into the popular general object detection.

2.3 Vision Transformer

The Transformer is proposed by Vaswani et al. (2017), which is widely used in NLP tasks. Recently, the pioneering work of vision Transformer ViT (Dosovitskiy et al., 2021) demonstrated that the pure Transformer-based model can also achieve competitive performance in vision tasks. Based on the success of ViT, many studies have on designing more advanced Transformer base networks been published, including image processing (Wan et al., 2021), classification (Wang et al., 2021), object detection (Carion et al., 2020), and semantic segmentation (Zheng et al., 2021). However, the normal ViT-based models are not compatible with many downstream tasks due to the high computational cost. To alleviate the limitations, an efficient and effective hierarchical Transformer named Swin Transformer (Liu et al., 2021) was proposed as a unified vision backbone. Swin Transformer designed the shifted windows mechanism, achieving state-of-the-art performance in many downstream tasks. We introduce Swin Transformer due to its excellent characteristics, and the hierarchical architecture is designed to reduce the complex computation by progressively decreasing the shape of feature maps.

3 Materials and methods

3.1 Dataset

We built a wheat spike detection dataset named WSD-2022, containing a total of 6,404 images, of which 978 images we collected ourselves in the field environment. We conducted wheat image collection in four locations, including Dangtu County, Ma'anshan; Feidong County, Hefei; Guizhi District, Chizhou; and Susong County, Anqing. The images were collected from April 18 to May 10, 2021 from the flowering stage to the milk stage of maturity. We collected the wheat spikes of varieties with different colors, shapes, and densities, thus increasing the diversity of the data. We shot the images using different types of cameras at different shooting angles and distances to collect image data under different lighting conditions to enhance the robustness of the model. About 80% of the images were captured at a resolution of over 3,000*4000 pixels. The captured images need to label each wheat spike, and we use LabelImg software to annotate the bounding boxes around the wheat spikes. Each wheat spike is labeled with a bounding box, the annotation is represented as a vector (x,y,w,h) where (x,y) are the coordinates of the upper left and (w,h) are the width/height of the bounding box. Figure 2 shows some examples of WSD-2022. Due to the different shooting angles, different lighting conditions, different wheat growth periods, different wheat distribution densities, and different wheat spike sizes, we can find the diversity and complexity of the dataset. We randomly split the WSD-2022 into training and validation subsets at a ratio of 8:2. The details of the two subsets are summarized in Table 1.

3.2 MW-Swin Transformer

3.2.1 Overall architecture

This section describes the design of MW-Swin Transformer. The pyramid structure was introduced based on the Transformer model to generate hierarchical feature maps for downstream tasks. The overall architecture of MW-Swin Transformer is similar to CNN networks. As shown in (Figure 1). For an input image with size of $H*W*3$, we follow Swin Transformer to split the image into patches at first (we treat each patch as a "token"); the patch size is 4*4. By such approach, the feature dimension of each patch becomes 4*4*3 = 48. Then, a linear embedding layer is employed to project the feature dimension to arbitrary dimension (set as C). To produce hierarchical feature representation, the model architecture consists of four stages; a patch merging layer is added after each stage for down-sampling (reduce the number of tokens, which is similar to the pooling layer in CNN).

In the first stage, we divide the input image into $HW/4^2$ patches, with a size of 4*4*3 for each of them. Through the linear

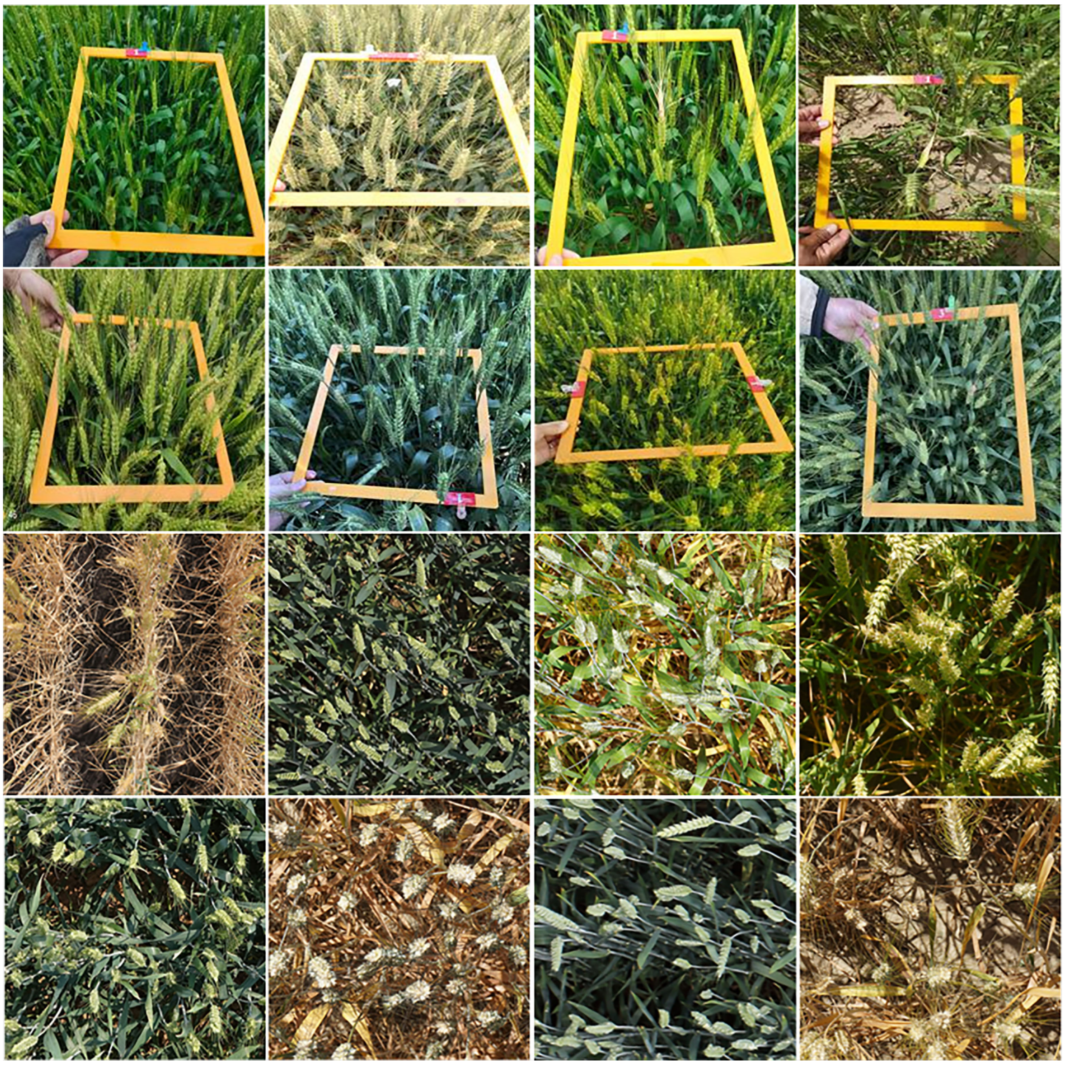


FIGURE 2
Samples of the WSD-2022 dataset. The first and second rows of the figure show the images that we acquired, while the third and fourth rows of the figure come from GWHD.

embedding layer, we feed the flattened patches to MW-Swin Transformer blocks (the number of blocks is represented by N), and the output is reshaped to a feature map with a size of $H/4*W/4*C_1$ (represented as F_1). The patch merging layer down-sampled each feature map F_i,i={1,2,3,4} with strides [4, 8, 16, 32] with respect to the size of the input image. The output dimensions of F_i is set to C_i,i={1,2,3,4} . Therefore, the output resolution of each stage is $H/4*W/4*C_1$, $H/8*W/8*C_2$, $H/16*W/16*C_3$, and $H/32*W/32*C_4$, respectively. With the hierarchical structure, our model possesses the progressive shrinking strategy that adjusts the output scale of each stage so that we can easily apply the model to downstream tasks.

TABLE 1 Number of images in the WSD-2022 dataset.

WSD-2022	Train	Validation	Total
Ours	782	196	978
GWHD	4,309	1,117	5,426
Total	5,091	1,313	6,404

3.2.2 MW-Swin Transformer block

Transformer obtains the powerful ability of long-range context modeling, but the computation complexity of conventional Transformer is quadratic to feature map size. For dense prediction tasks with high-resolution images as input, using conventional Transformer is expensive. Therefore, Swin Transformer is proposed to perform self-attention by non-overlapping local windows and shifted windows. However, the window size of Swin Transformer is fixed, which is not conducive to detecting objects of different sizes. To enlarge the receptive field and obtain global self-attention more flexibly, we propose the MW-Swin Transformer; the architecture is similar to the feature pyramid network, using different-sized windows to detect objects across a large range of scales.

As shown in Figure 3, two consecutive MW-Swin Transformer blocks are presented. Each block contains two LayerNorm (Bosilj et al. 2020) layers, a multi-head self-attention (MSA), and a multilayer perceptron (MLP). The multi-window MSA (MW-MSA) and the shifted multi-window MSA (SMW-MSA) are adopted in the consecutive Transformer blocks, respectively. With the MW-MSA module and the SMW-MSA module, consecutive MW-Swin Transformer blocks can be represented as:

$$\bar{z}^l = MW - SMA(LN(z^{l-1})) + z^{l-1}$$

$$\bar{z}^l = SR(\bar{z}^l)$$

$$z^l = MLP(LN(\bar{z}^l)) + \bar{z}^l \qquad (1)$$

$$\bar{z}^{l+1} = SMW - SMA(LN(z^l)) + z^l$$

$$\bar{z}^{l+1} = SR(\bar{z}^{l+1})$$

$$z^{l+1} = MLP(LN(\bar{z}^{l+1})) + \bar{z}^{l+1}$$

where $\bar{z}^l$ and z^l represent the outputs of (S)MW-SMA module and the MLP for the block, respectively. MW-MSA equals $Concat(W{-}MSA(z^{l-1})_1, W{-}MSA(z^{l-1})_2, W{-}MSA(z^{l-1})_3)$, where $W{-}MSA(\bullet)_i, i{=}1,2,3$ indicates the i_{th} window with size X , and we set $X{=}[7,9,11]$ in experiments. $SR(\bullet)$ denotes the spatial reduction module to reduce the spatial scale of $\bar{z}^l$, which reduces the memory and computational cost. Similar to the conventional Transformer (Dosovitskiy et al., 2021; Liu et al., 2021), the attention operation can be computed as follows:

$$Attention(Q,K,V) = Soft\max\left(\frac{QK^T}{\sqrt{d}} + B\right)V \qquad (2)$$

where Q,K,V represent the query, key, and value matrices; the other parameters are in accordance with Swin Transformer.

Compared with the previous MSA in vision Transformers, the MW-MSA controls the computation area in multi-window as a unit. It reduces the complexity and computational cost, enhancing the ability to detect multi-scale features. MW-Swin Transformer block can serve as a plug-and-play block to replace the raw Transformer block in Swin Transformer, with only minor modifications to the vanilla structure.

3.2.3 Architecture variants

We named the base model WheatFormer-B, which is a trade-off between efficiency and accuracy. Considering higher

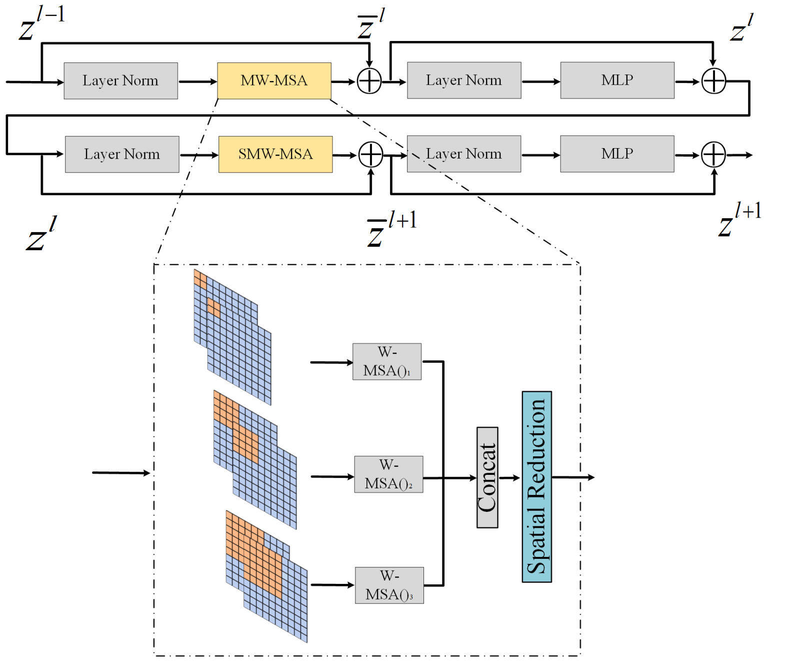


FIGURE 3
MW-Swin Transformer block.

efficiency needs in some cases, we have introduced a small version named WheatFormer-S. On the other hand, when accuracy needs to be considered more, we have introduced a large version named WheatFormer-L. The architectures of our base model and variants are listed in Table 2.

3.3 Wheat detector

3.3.1 One-stage object detector

FCOS is a one-stage anchor-free object detection algorithm (Tian et al., 2019) with higher accuracy and faster speed compared with the representative model Faster R-CNN (Ren et al., 2017) and other two-stage detectors. FCOS mainly consists of three parts: a feature extraction backbone, a feature pyramid network (FPN), and a detection head. The backbone extracts multi-level features of the input image. Then, low-level spatial information and high-level semantic information are fed into FPN, generating multi-scale feature maps. In previous research, low-level information can obtain more detailed texture information, which leads to more efficient detection. High-level information gets more semantic information and is more suitable for classification. FCOS is a pixel-based detector, which means that each pixel on the feature map is used for regression. First, each pixel map back to the original input image, and a pixel considers a positive sample if its location falls within any ground-truth box with the correct class label. Otherwise, it is a negative sample. As for regression, FCOS uses a vector $t^{\star}=(l^{\star},t^{\star},r^{\star},b^{\star})$, where $l^{\star},t^{\star},r^{\star},b^{\star}$ denote the distances from the location (x,y) to the four sides of the bounding box, as shown in Figure 4. The target regression process can be formulated as follows:

$$\begin{aligned} l^{\star} &= x - {x_0}^{(i)} \\ t^{\star} &= y - {y_0}^{(i)} \\ r^{\star} &= {x_1}^{(i)} - x \\ b^{\star} &= {y_1}^{(i)} - y \end{aligned} \tag{3}$$

where $({x_0}^{(i)},{y_0}^{(i)})$ and represent coordinates of the left-top and right-bottom corners of the bounding box.

3.3.2 WIoU loss

The training loss function of the proposed WheatFormer mainly obtains three branch loss functions:

$$L_{WheatFormer} = \frac{1}{N_{pos}} L_{cls} + \frac{\lambda_1}{N_{pos}} L_{\text{center}-ness} + \frac{\lambda_2}{N_{pos}} L_{\text{reg}} \tag{4}$$

where L_{cls} and $L_{\text{center}-ness}$ represent the classification and center-ness loss function which are designed in FCOS. N_{pos} denotes the number of positive pixels. λ_1 and λ_2 are balance weights to adjust the proportions of three branch loss functions. The parameters follow the settings in Tian et al. (2019). FCOS uses IoU loss to calculate the regression loss, which can be formulated as follows:

$$L_{\text{reg}} = \sum\nolimits_{x,y \in (R^p \cup R^n)} (1 - IoU(\text{Pr}^{x,y}, Gt^{x,y})) \tag{5}$$

where R^p represents the positive sample region and R^n denotes the negative sample region. $Gt^{i,j}$ indicates the ground truth localization of the pixel (x,y) , while $Pr^{i,j}$ denotes the predicted target of (x,y) .

The IoU loss regresses all bound variables as a whole for joint regression and directly enforces the maximum overlap between the prediction bounding box and the ground truth. The IoU loss leads to faster convergence and more accurate localization compared with the *ln−norm* loss used in previous studies. However, the IoU loss cannot provide moving gradients for non-overlapping cases, *i.e.*, IoU loss is only valid when the bounding boxes overlap. Based on previous researches and the IoU loss, we consider three important geometric metrics, which are the overlap region, Euclidean distance, and aspect ratio of bounding boxes. In summary, we add a penalty term to the IoU loss, named WIoU loss. The new loss function directly minimizes the Euclidean distance between the predicted box and the ground truth. At the same time, we take into account the effect of the consistency of aspect ratios. The WIoU loss function is defined as follows:

$$\begin{gathered} L_{\text{reg}} = \sum\nolimits_{x,y \in (R^p \cup R^n)} (1 - IoU(\text{Pr}^{x,y}, Gt^{x,y}) + \psi \left\| \text{Pr}^{x,y}, Gt^{x,y} \right\|_2) \\ \psi = \tfrac{4}{\pi^2} (\arctan \tfrac{{w_{Gt}}^{x,y}}{{h_{Gt}}^{x,y}} - \arctan \tfrac{{w_{\text{Pr}}}^{x,y}}{{h_{\text{Pr}}}^{x,y}})^2 \end{gathered} \tag{6}$$

where ψ measures the consistency of the aspect ratio and plays the role of regularization for the distance between the predicted bounding box and the target bounding box. w_{Gt} and h_{Gt} represent the width and height of the ground truth. w_{Pr} and h_{Pr} represent the width and height of the predicted bounding box. The optimization of WIoU loss is the same as the IoU loss.

TABLE 2 Detailed settings of WheatFormer variants.

Models	C_1,C_2,C_3,C_4	N_1,N_2,N_3,N_4	#Head	#Expansion	#Params (MB)
WheatFormer-S	[96, 192, 384, 768]	[2, 2, 2, 2]	32	α=4	42.4
WheatFormer-B	[96, 192, 384, 768]	[2, 2, 6, 2]	32	α=4	60.1
WheatFormer-L	[96, 192, 384, 768]	[2, 2, 18, 2]	32	α=4	100.6

C_i , channel number of the hidden layers in each stage; N_i , layer numbers in each stage; #Head, query dimension of each head; #Expansion, expansion layer of each multilayer perceptron; #Params, amount of model parameters.

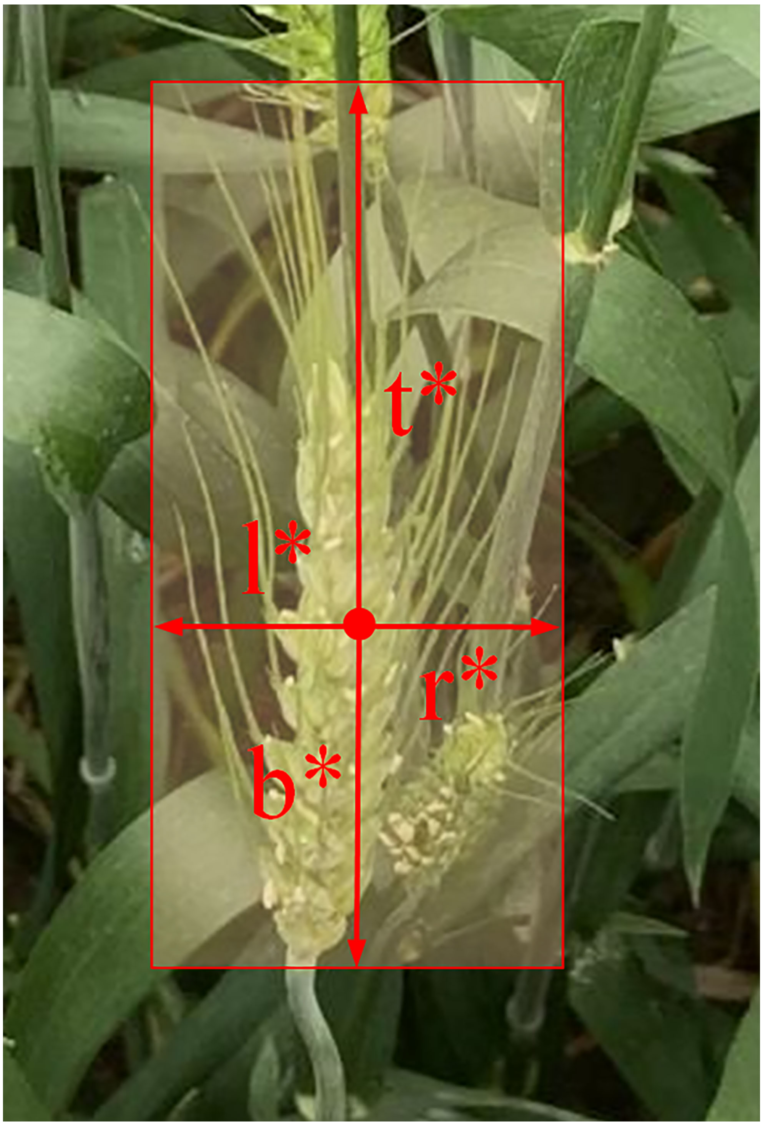


FIGURE 4
Regression method of FCOS. *l**, *t**, *r**, and *b** represent the distances from the pixel to the left, top, right, and bottom, respectively, of the bounding box.

4 Experiments and discussion *AP*

4.1 Experimental settings

All the experiments were performed using the Pytorch deep learning frame, and the operation system was Ubuntu 18.04 with CUDA10.1. We use a piece of NVIDIA TITAN RTX GPU, Intel Core i9-9900k CPU with 128GB RAM. Furthermore, we train our model with the AdamW (Loshchilov and Hutter, 2017) optimizer for 24 epochs. The initial learning rate is $1e{-4}$, and the weight decay is 0.05. The settings of comparison networks follow the original settings.

4.2 Evaluation metrics

In our experiments, we use the evaluation metrics as the metric definition of the COCO dataset. Average precision (AP) is the area surrounded by the precision-recall curve. The definition of AP is defined as Formula 7. AP@50 (AP_{50}) means the value when IoU is equal to 0.5, $AP@75$ (AP_{75}) is the AP value when the IoU equals 0.75, and the mean AP (mAP) is the threshold of the IoU from 0.5 to 0.95 ($AP@[0.5:0.05:0.95]$) with a step size of 0.05.

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ AP &= \int_0^1 precision(recall)d(recall) \end{aligned} \tag{7}$$

where TP (true positive), FP (false positive), and FN (false negative) represent the number of correctly detected wheat spikes, false detected wheat spikes, and missing detected wheat spikes. At the same time, we use AP_s , AP_m , AP_l defined in the COCO dataset in our experiments, which represent the detection accuracy for different target sizes. Considering that the wheat spike in the dataset occupies a larger proportion of the image, we only apply AP_m (for medium targets) and AP_l (for large targets) as the evaluation metric. In the field of object detection, AP metric is widely adopted for evaluating the comprehensive detection performance of the model.

4.3 Model performance

The experiments in this section aim to demonstrate the effectiveness of the proposed method in terms of detection performance. We compared seven state-of-the-art algorithms, including Faster R-CNN (Madec et al., 2019), Mask R-CNN (He et al., 2020), FCOS (Tian et al., 2019), ATSS (Zhang et al., 2020b), SSD (Fu et al., 2017), Centernet (Zhou et al., 2019), and YOLOv3 (Redmon and Farhadi, 2018). Faster R-CNN and Mask R-CNN are two-stage networks, and the rest are one-stage networks. The experimental results are listed in Table 3, and we can find that the proposed WheatFormer outperforms the other models. To be specific, compared with the two-stage CNN-based models, WheatFormer achieves about 10–20% higher in AP_{50} and 8–15% improvement in. Compared with the one-stage CNN models, our model increases the AP_{50} and mAP by 1.2–11.5 and 2.2–9.5%, respectively. In terms of Swin Transformer-based models, the detection performance is generally better than the CNN-based models. The FCOS-based Swin Transformer achieves a mAP of 0.452, while our model increases mAP by 0.7% and AP_{50} by 3.2%. The Mask R-CNN based on Swin Transformer achieves the AP_{50} of 0.914, which is comparable to that of WheatFormer, but our model gets a higher mAP of 3.3%. Considering the model parameters, our model achieves a larger size than most CNN models but is similar to Swin Transformer-based models. We show some comparison examples in Figure 5 and the detection results of WheatFormer in Figure 6. Figure 5 shows that Faster R-CNN has too many overlapping prediction boxes, and YOLOv3 obtains too many missing boxes. At the same time, WheatFormer obtains a higher accuracy than the comparison models in classification. In Figure 6, we can find that WheatFormer has excellent detection performance at different shooting angles, different light conditions, different wheat growth periods, different wheat distribution densities, and different wheat spikes sizes. WheatFormer can accurately identify most wheat spikes even at high density and high occlusion. This intuitively illustrates the excellent performance of WheatFormer.

4.4 Ablation experiments

As mentioned, the major drawbacks of CNN models are the consistently produced local receptive fields, which are unsuitable

TABLE 3 Detection results on WSD-2022.

Method	Backbone	mAP	AP_{50}	AP_{75}	AP_m	AP_l	#Params (MB)
Faster R-CNN	ResNet50	0.301	0.709	0.215	0.284	0.339	39.4
Mask R-CNN		0.345	0.774	0.237	0.311	0.382	41.9
Faster R-CNN	ResNet101	0.304	0.750	0.208	0.306	0.352	57.6
Mask R-CNN		0.366	0.812	0.246	0.331	0.394	60.1
FCOS	ResNet50	0.368	0.825	0.250	0.355	0.409	30.6
ATSS		0.364	0.803	0.255	0.357	0.402	30.6
SSD	SSDVGG	0.428	0.890	0.362	0.382	0.488	22.7
CenterNet	ResNet18	0.414	0.876	0.318	0.345	0.487	13.8
YOLOv3	DarkNet53	0.437	0.906	0.381	0.387	0.497	58.7
Faster R-CNN	Swin Transformer	0.397	0.881	0.276	0.352	0.450	65.6
Mask R-CNN		0.426	0.914	0.318	0.379	0.473	68.1
FCOS		0.452	0.886	0.402	0.415	0.523	43.8
WheatFormer	MW-Swin Transformer	**0.459**	**0.918**	**0.384**	**0.415**	**0.533**	**60.1**

Faster R-CNN and Mask R-CNN are the representative models of two stages. FCOS, ATSS, SSD, CenterNet, and YOLOv3 are the representative models of one stage.

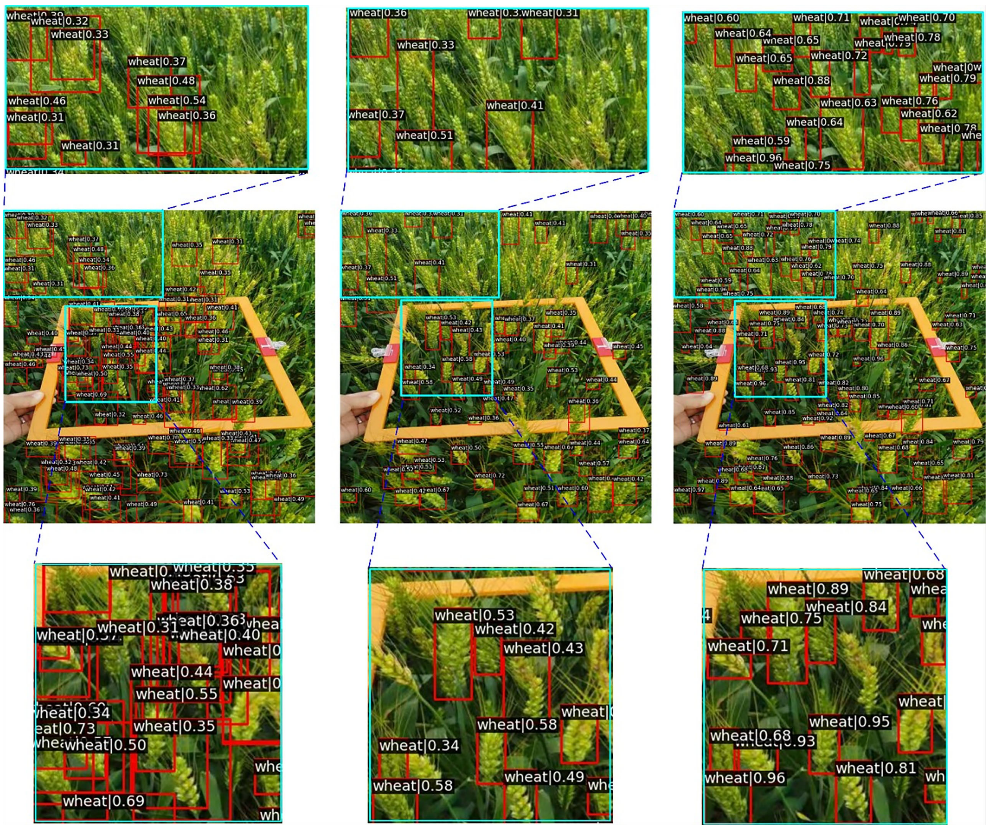


FIGURE 5
Visualization of the comparative models. The left column represents the result of Faster R-CNN, the middle column represents the result of YOLOv3, and the right column represents the result of WheatFormer.

for detecting objects in complex backgrounds. There are relatively few studies on Transformers-based backbone applied to wheat spike detection. We conduct ablation experiments to represent the effectiveness of our proposed methods.

4.4.1 Effect of the MW-Swin Transformer

In this part, we describe the effectiveness of the proposed MW-Swin Transformer. The results are listed in Table 4, which contains three backbones: the CNN backbone, the Swin Transformer backbone, and the MW-Swin Transformer backbone. Obviously, the Swin Transformer backbone-based models greatly improve the detection performance of the state-of-the-art algorithms. For a detailed representative comparison of different backbones, we show the precision–recall curve of WheatFormer in Figure 7. Specifically, compared with the CNN backbone and the Swin Transformer backbone, the WheatFormer boosts the Loc, Sim, Oth, and BG to 0.964, 0.964, 0.964, and 0.990. It obtains 9.1% improvements on mAP and 9.3% improvements on AP_{50} after replacing the backbone with MW-Swin Transformer. This indicates that the proposed Transformer can effectively increase the detection ability of the detectors.

4.4.2 Effect of the WIoU loss

The loss function plays an important role in the deep learning training process. To further validate the performance of the proposed WioU loss, we conduct experiments comparing IoU, GioU, and CioU (Zheng et al., 2020). We present the comparison results in Table 5. We can find that GioU, CioU, and WioU make further detection improvements than the original IoU loss for most cases—for instance, the WheatFormer with WioU loss obtains 0.452 mAP , which is 2.9% higher than the IoU-based model, 1% higher than the GioU-based model, and 2.4% higher than the CioU-based model. Therefore, we can conclude that the WheatFormer can obtain better detection performance when trained with WioU loss.

4.4.3 Performance of the variant models

As mentioned, we constructed three different variants of WheatFormer, and the detection results are shown in Table 6.

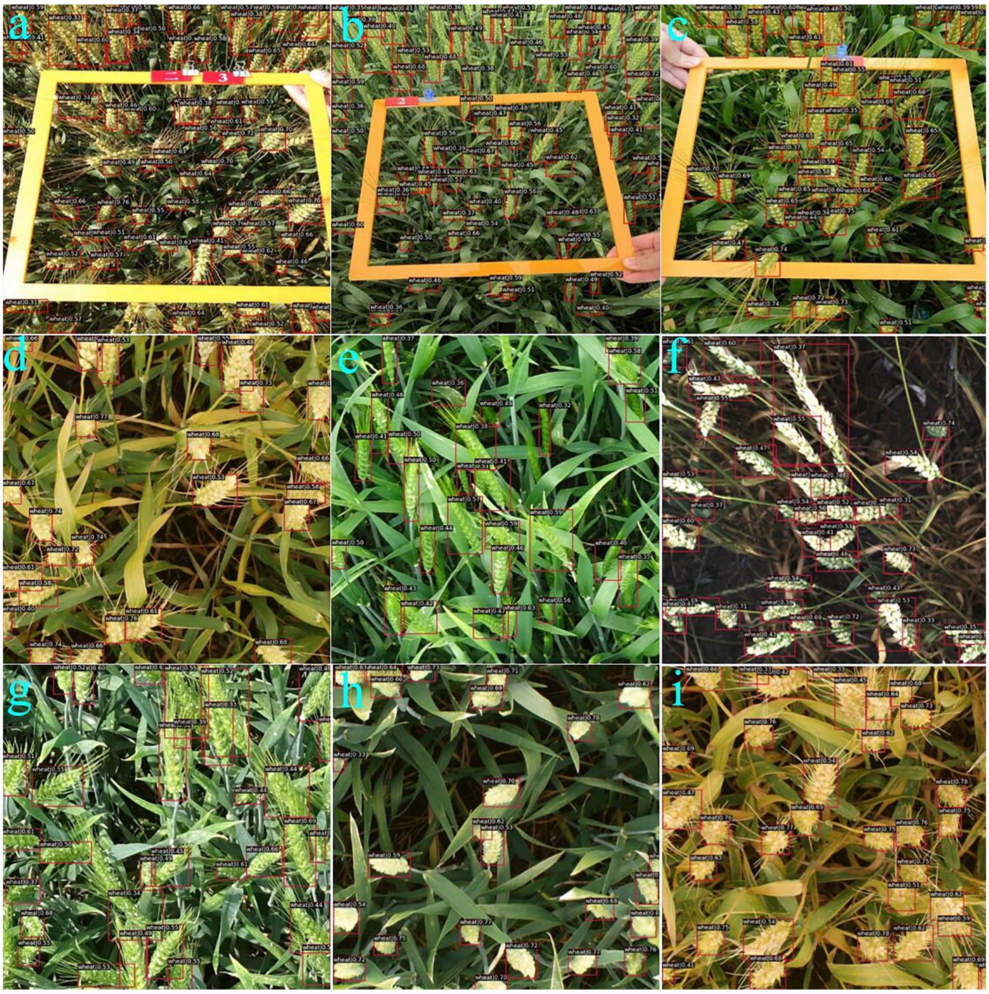


FIGURE 6
Visualization of detected results by the WheatFormer. **(A)** Early maturity, 65 spikes per image, direct sunlight, and wheat ear group with 80° viewing angle of photographing, **(B)** filling stage, 75 spikes per image, diffuse light conditions, and wheat ear group with 45° viewing angle of photographing, **(C)** filling stage, 45 spikes per image, diffuse light conditions, and wheat ear group with 45° viewing angle of photographing, **(D)** early maturity, 25 spikes per image, diffuse light conditions, and wheat ear group with 90° viewing angle of photographing, **(E)** poplar blossom, 23 spikes per image, direct sunlight, and wheat ear group with 45° viewing angle of photographing, **(F)** the milk stage of maturity, 30 spikes per image, direct sunlight, and wheat ear group with 90° viewing angle of photographing, **(G)** poplar blossom, 27 spikes per image, direct sunlight, and wheat ear group with 30° viewing angle of photographing, **(H)** the milk stage of maturity, 22 spikes per image, diffuse light conditions, and wheat ear group with 90° viewing angle of photographing, and **(I)** the milk stage of maturity, 30 spikes per image, diffuse light conditions, and wheat ear group with 90° viewing angle of photographing.

WheatFormer-S obtains 42.4 MB parameters, similar to the Swin Transformer-based FCOS (43.8 MB), while WheatFormer achieves 0.438 at *mAP* (1.4% lower than SSD) and 0.908 at AP_{50} (2.2% higher than Swin Transformer-based FCOS). WheatFormer-B obtains 60.1 MB parameters, the same as Mask R-CNN. Nevertheless, our model achieves 0.459 at *mAP* (9.3% higher than Mask R-CNN) and 0.918 at AP_{50} (10.6% higher than Mask R-CNN), which significantly surpasses the detection ability of Mask R-CNN. The large version obtains parameters of 100.6 MB, showing a better performance than the previous versions.

4.5 Limitations and future work

In this work, we conduct extensive experiments to evaluate the effectiveness of the proposed methods. The experimental results prove that the proposed methods can greatly improve the

TABLE 4 Comparison of different backbones.

Method	CNN backbone	Swin Transformer	MW-Swin Transformer		*AP*$_{50}$	*AP*$_{75}$
Faster R-CNN	✓			0.301	0.709	0.215
		✓		0.397 (9.6%↑)	0.881 (17.2%↑)	0.276 (6.1%↑)
			✓	0.417 (2%↑)	0.893 (1.2%↑)	0.315 (1.2%↑)
Mask R-CNN	✓			0.345 *mAP*	0.774	0.237
		✓		0.426 (8.1%↑)	0.914 (14%↑)	0.318 (8.1%↑)
			✓	0.433 (0.7%↑)	0.909 (0.5%↓)	0.344 (2.6%↑)
Centernet	✓			0.414	0.876	0.318
		✓		0.436 (2.2%↑)	0.913 (3.7%↑)	0.372 (5.4%↑)
			✓	0.448 (1.2%↑)	0.912 (0.1%↑)	0.365 (0.7%↓)
WheatFormer	✓			0. 368	0.825	0. 250
		✓		0. 452 (8.4%↑)	0. 886 (6.1%↑)	0. 402 (15.2↑)
			✓	**0. 459** (0.7%↑)	**0. 918** (3.2%↑)	**0. 384** (1.8%↓)

Bold values are the results of our experimental method.
The symbols "↑" means the increase values compared to the previous method, "↓" means the decrease values compared to the previous method, and "✓" means the method used in the model.

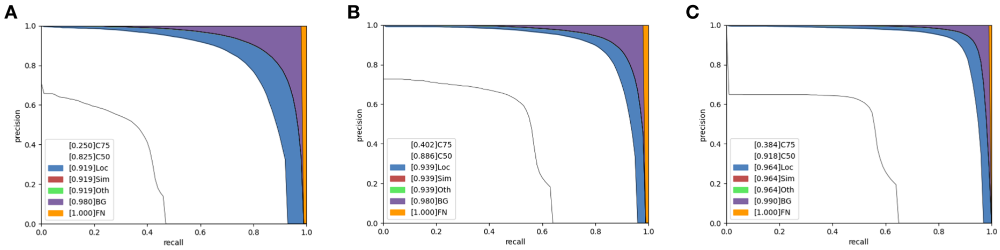


FIGURE 7
Precision–recall (PR) curves of WheatFormer with different backbones. **(A)** WheatFormer with convolutional neural network backbone. **(B)** WheatFormer with Swin Transformer backbone. **(C)** WheatFormer with MW-Swin Transformer backbone. C75: PR at threshold equals 0.75; C50: PR at threshold equals 0.50; Loc: PR at threshold equals 0.1, and location errors ignored without duplicate detections; Sim: PR after supercategory false positives are removed; Oth: PR after all class confusions are removed; BG: PR after all background false positive are removed; FN: PR after all remaining errors are removed.

TABLE 5 Results of WheatFormer with different IoU loss functions.

Method	IoU	GioU	CioU	WioU	*mAP*	*AP*$_{50}$	*AP*$_{75}$
WheatFormer	✓				0.423	0.894	0.322
		✓			0.442	0.896	0.374
			✓		0.428	0.900	0.326
				✓	**0.459**	**0.918**	**0.384**

Bold values are the results of our experimental method.
The symbols "✓" means the method used in the model.

TABLE 6 Comparison of variant models.

Method	*mAP*	*AP*$_{50}$	*AP*$_{75}$	*AP*$_{m}$	*AP*$_{l}$	#Params (M)
WheatFormer-S	0.438	0.908	0.366	0.402	0.516	42.4
WheatFormer-B	0.459	0.918	0.384	0.415	0.533	60.1
WheatFormer-L	0.466	0.927	0.400	0.422	0.524	100.6

detection performance of wheat spike detection. Although WheatFormer has shown to be effective in wheat spike detection tasks, there are still some limitations. It is worth noting that the experiment is only perfomed on the WSD-2022 dataset with a limited number of images. Moreover, our method attempts to improve the detection ability of the spike detector, while the parameters of our base model are relatively large. In future research, we will focus on solving the above-mentioned problems. Firstly, we will collect more wheat spike images containing more regions and more growth cycles to validate our methods. Secondly, we will continue to design more lightweight models to improve the capabilities for practical applications.

5 Conclusions

In this paper, we explore a Transformer-based network for wheat spike detection within a newly constructed dataset. We are the first to introduce the Transformer for wheat spike detection. To extract global and long-range semantic information, we design the MW-Swin Transformer as the backbone, and we propose the WioU loss function to improve positioning accuracy. Finally, we created a wheat spike dataset named WSD-2022 to verify the effectiveness of our model. The extensive experiments show that the method proposed in this study can obtain an encouraging detection performance compared with those state-of-the-art algorithms. We hope that this research will provide novel insights into the development of more advanced detection methods in the agricultural field.

Data availability statement

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

Author contributions

QZ: conceptualization, methodology, software, investigation, formal analysis, and writing—original draft. ZH: conceptualization, methodology, software, investigation, formal analysis, and writing—original draft. SZ: visualization and investigation. LJ, LW and RW: conceptualization, funding acquisition, resources, supervision, and writing—review and editing. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Key R&D Program of China (2019YFE0125700) and the Natural Science Foundation of Anhui Province (2208085MC57).

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

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *ArXiv abs* 2004, 10934. doi: 10.48550/arXiv.2004.10934

Bosilj, P., Aptoula, E., Duckett, T., and Cielniak, G. (2020). Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *J. Field Robotics* 37, 7–19. doi: 10.1002/rob.21869

Cai, Z., and Vasconcelos, N. (2018). "Cascade r-CNN: Delving into high quality object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6154–6162.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-End object detection with transformers," in *Computer vision – ECCV 2020*. Ed. A. Vedaldi, et al (Cham: Springer International Publishing), 213–229.

Catherine, F., Klaus, F., Mayer, X., Rogers, J., and Eversole, K. (2014). SLICING THE WHEAT GENOME. *Science* 345, 285–285. doi: 10.1126/science.1257983

David, E., Serouart, M., Smith, D., Madec, S., Velumani, K., Liu, S., et al. (2021) Global wheat head dataset 2021: more diversity to improve the benchmarking of wheat head localization methods. arXiv preprint arXiv:2105.07660.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv abs* 2010, 11929. doi: 10.48550/arXiv.2010.11929

Fang, Y., Qiu, X., Guo, T., Wang, Y., and Gui, L. (2020). An automatic method for counting wheat tiller number in the field with terrestrial LiDAR. *Plant Methods* 16, 132. doi: 10.1186/s13007-020-00672-8

Fernandez-Gallego, J., Buchaillot, M., Gutiérrez, N. A., Nieto-Taladriz, M., Araus, J., and Kefauver, S. (2019). Automatic wheat ear counting using thermal imagery. *Remote Sens.* 11 (7), 751. doi: 10.3390/rs11070751

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. (2017). DSSD : Deconvolutional single shot detector. *ArXiv abs*, 1701.06659. doi: 10.48550/arXiv.1701.06659

Girshick, R. (2015). "Fast r-CNN," in *IEEE International Conference on Computer Vision (ICCV)*. 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.

Gong, B., Ergu, D., Cai, Y., and Ma, B. (2020). Real-time detection for wheat head applying deep neural network. *Sensors* 21 (1), 191. doi: 10.3390/s21010191

Hasan, M. M., Chopin, J. P., Laga, H., and Miklavcic, S. J. (2018). Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods* 14, 100. doi: 10.1186/s13007-018-0366-8

He, K. M., Gkioxari, G., Dollár, P., and Girshick, R (2017). "Mask r-cnn," In *Proceedings of the IEEE international conference on computer vision*, (pp. 2961–2969).

He, K. M., Gkioxari, G., Dollar, P., and Girshick, R. (2020). "Mask r-CNN," in *Ieee Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42. 386–397. doi: 10.1109/tpami.2018.2844175

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*. pp. 4700–4708.

Jin, X., Liu, S., Baret, F., Hemerlé, M., and Comar, A. (2017). Estimates of plant density of wheat crops at emergence from very low altitude UAV imagery. *Remote Sens. Environ.* 198, 105–114. doi: 10.1016/j.rse.2017.06.007

Khoroshevsky, F., Khoroshevsky, S., and Bar-Hillel, A. (2021). Parts-per-Object count in agricultural images: Solving phenotyping problems *via* a single deep neural network. *Remote Sens.* 13, 2496. doi: 10.3390/rs13132496

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: Hierarchical vision transformer using shifted windows," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.

Loshchilov, I., and Hutter, F. (2017). Fixing weight decay regularization in Adam. *ArXiv abs*, 1711.05101. doi: 10.48550/arXiv.1711.05101

Madec, S., Jin, X., Lu, H., De Solan, B., Liu, S., Duyme, F., et al. (2019). Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* 264, 225–234. doi: 10.1016/j.agrformet.2018.10.013

Misra, T., Arora, A., Marwaha, S., Chinnusamy, V., Rao, A. R., Jain, R., et al. (2020). SpikeSegNet-a deep learning approach utilizing encoder-decoder network with hourglass for spike segmentation and counting in the wheat plant from visual imaging. *Plant Methods* 16 (1), 1–20. doi: 10.1186/s13007-020-00582-9

Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., and Lin, D. (2019). "Libra R-CNN: Towards balanced learning for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 821–830.

Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P., and French, A. P. (2017). "Deep learning for multi-task plant phenotyping," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2055–2063.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788.

Redmon, J., and Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv abs* 1804.02767. doi: 10.48550/arXiv.1804.02767

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

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. (2019). "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 658–666.

Sadeghi-Tehran, P., Virlet, N., Ampe, E. M., Reyns, P., and Hawkesford, M. J. (2019). DeepCount: In-field automatic quantification of wheat spikes using simple linear iterative clustering and deep convolutional neural networks. *Front. Plant Sci.* 10. doi: 10.3389/fpls.2019.01176

Tian, Z., Shen, C., Chen, H., and He, T. (2019). "FCOS: Fully convolutional one-stage object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*. 9626–9635.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). *Attention is all you need, proceedings of the 31st international conference on neural information processing systems* (Long Beach, California, USA: Curran Associates Inc.), 6000–6010.

Wang, D., Fu, Y., Yang, G., Yang, X., Liang, D., Zhou, C., et al. (2019). Combined use of FCN and Harris corner detection for counting wheat ears in field conditions. *IEEE Access* 7, 178930–178941. doi: 10.1109/ACCESS.2019.2958831

Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., et al. (2021). "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *IEEE/CVF International Conference on Computer Vision (ICCV)*. 548–558.

Wan, Z., Zhang, J., Chen, D., and Liao, J. (2021). "High-fidelity pluralistic image completion with transformers," in *18th IEEE/CVF International Conference on Computer Vision (ICCV 2021), IEEE.*

Xizhou, Z., Weijie, S., Lewei, L., Bin, L., Xiaogang, W., and Jifeng, D. (2020). Deformable DETR: Deformable transformers for end-to-End object detection. *arXiv preprint* arXiv:2010.04159 doi: 10.48550/arXiv.2010.04159

Yang, B., Gao, Z., Gao, Y., and Zhu, Y. (2021). Rapid detection and counting of wheat ears in the field using YOLOv4 with attention module. *Agronomy* 11 (6), 1202. doi: 10.3390/agronomy11061202

Yu, J., Jiang, Y., Wang, Z., Cao, Z., and Huang, T. (2016). *UnitBox: An advanced object detection network, proceedings of the 24th ACM international conference on multimedia* (Amsterdam, The Netherlands: Association for Computing Machinery), 516–520.

Zhang, S., Chi, C., Yao, Y., Lei, Z., and Li, S. Z. (2020b). "Bridging the gap between anchor-based and anchor-free detection *via* adaptive training sample selection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9756–9765.

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 (Basel)* 20 (5), 1520. doi: 10.3390/s20051520

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2021). "Rethinking semantic segmentation from a sequence-to-Sequence perspective with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6877–6886.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 12993–13000).

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *ArXiv abs*, 1904.07850. doi: 10.48550/arXiv.1904.07850

frontiers | Frontiers in Plant Science

TYPE Methods
PUBLISHED 23 November 2022
DOI 10.3389/fpls.2022.1010981

OPEN ACCESS

EDITED BY
Peng Chen,
Anhui University, China

REVIEWED BY
Shichao Jin,
Nanjing Agricultural University, China
Muhammad Shoaib Farooq,
University of Management and Technology, Pakistan

*CORRESPONDENCE
Sook Yoon
syoon@mokpo.ac.kr
Dong Sun Park
dspark@jbnu.ac.kr

This article was submitted to Sustainable and Intelligent Phytoprotection,
a section of the journal Frontiers in Plant Science

SPECIALTY SECTION
RECEIVED 03 August 2022
ACCEPTED 20 October 2022
PUBLISHED 23 November 2022

CITATION
Xu M, Yoon S, Jeong Y and Park DS (2022) Transfer learning for versatile plant disease recognition with limited data.
Front. Plant Sci. 13:1010981.
doi: 10.3389/fpls.2022.1010981

Transfer learning for versatile plant disease recognition with limited data

Mingle Xu[1,2], Sook Yoon[3]*, Yongchae Jeong[1] and Dong Sun Park[1,2]*

[1]Department of Electronics Engineering, Jeonbuk National University, Jeonbuk, South Korea,
[2]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonbuk, South Korea,
[3]Department of Computer Engineering, Mokpo National University, Jeonnam, South Korea

Deep learning has witnessed a significant improvement in recent years to recognize plant diseases by observing their corresponding images. To have a decent performance, current deep learning models tend to require a large-scale dataset. However, collecting a dataset is expensive and time-consuming. Hence, the limited data is one of the main challenges to getting the desired recognition accuracy. Although transfer learning is heavily discussed and verified as an effective and efficient method to mitigate the challenge, most proposed methods focus on one or two specific datasets. In this paper, we propose a novel transfer learning strategy to have a high performance for *versatile plant disease recognition*, on multiple plant disease datasets. Our transfer learning strategy differs from the current popular one due to the following factors. First, PlantCLEF2022, a large-scale dataset related to plants with 2,885,052 images and 80,000 classes, is utilized to pre-train a model. Second, we adopt a vision transformer (ViT) model, instead of a convolution neural network. Third, the ViT model undergoes transfer learning twice to save computations. Fourth, the model is first pre-trained in ImageNet with a self-supervised loss function and with a supervised loss function in PlantCLEF2022. We apply our method to 12 plant disease datasets and the experimental results suggest that our method surpasses the popular one by a clear margin for different dataset settings. Specifically, our proposed method achieves a mean testing accuracy of 86.29over the 12 datasets in a 20-shot case, 12.76 higher than the current state-of-the-art method's accuracy of 73.53. Furthermore, our method outperforms other methods in one plant growth stage prediction and the one weed recognition dataset. To encourage the community and related applications, we have made public our codes and pre-trained model[1].

KEYWORDS

plant disease recognition, transfer learning, vision transformer, self-supervised learning, few-shot learning, PlantCLEF2022

[1] https://github.com/xml94/MAE_plant_disease

1 Introduction

Keeping plants healthy is one of the essential challenges to having an expected and high yield. Traditionally, experts have to go to farms to check if plants are infected with diseases but deep learning enables the check to take place automatically based on their images. Because of the decent performance of deep learning, plant disease recognition has witnessed a significant improvement in recent years (Abade et al., 2021; Liu et al., 2021; Ngugi et al., 2021). To obtain a comparable recognition performance, a large-scale dataset is entailed to train a deep learning-based model. However, collecting images for plant disease is expensive and time-consuming. Besides, few images are normally available at the beginning of a plant disease recognition project when sanity checking should be executed before devoting more resources. Therefore, *limited dataset*, a situation where a few labeled images are accessible for some classes in the training process is one of the main issues in the literature (Fan et al., 2022). To facilitate this issue, many algorithms and strategies are proposed, such as data augmentation (Mohanty et al., 2016; Xu et al., 2022b; Olaniyi et al., 2022), transfer learning (Mohanty et al., 2016; Too et al., 2019; Chen J. et al., 2020; Xing and Lee, 2022; Zhao et al., 2022), few-shot learning (Afifi et al., 2020; Egusquiza et al., 2022), and semi-supervised learning (Li and Chao, 2021).

Although the challenge of a limited dataset is considered in many works, most of them merely focus on one or few specific datasets, such as the PlantVillage dataset (Mohanty et al., 2016; Too et al., 2019; Li and Chao, 2021), AI Challenger dataset (Zhao et al., 2022), tomato dataset (Xu et al., 2022b), wheat and rice dataset (Sethy et al., 2020; Rahman et al., 2020), cucumber (Wang et al., 2022), and apple leaf disease dataset (Fan et al., 2022). A basic question in this situation is whether a useful method for one dataset is helpful for other datasets. Further, there is a fundamental desire to find a robust method for most plant disease recognition applications. On the other hand, improving the application performance with a limited dataset is desired. For example, can we get a comparable result with only 20 training images for each class (20-shot)? To address these two issues, we propose a novel transfer learning strategy to achieve high performance for different limited datasets and various types of plants and diseases.

Via obtaining a good feature space, transfer learning aims to learn something beneficial for a target task with a target dataset from a source task with a source dataset (Pan and Yang, 2009). In plant disease recognition, a deep learning-based model is generally pre-trained in the source dataset and then fine-tuned in the labeled target dataset. As shown in Figure 1, it is understood that three key factors essentially lead to a positive transfer learning performance, a *desired source dataset*, *powerful model*, and suitable *loss function* to pre-train the model (Wu et al., 2018; Kornblith et al., 2019; Kolesnikov et al., 2020; Tripuraneni et al., 2020; He et al., 2022). However, the three factors have been undeveloped in plant disease recognition.

First, it is beneficial to have a *plant-related* dataset with a high number of images and classes (*large scale*), as well as *wide image variation*. For example, a plant-related source dataset could be better than the widely used ImageNet (Deng et al., 2009) for plant disease recognition, which has been verified (Kim et al., 2021; Zhao et al., 2022). Hence, finding a suitable source dataset is essential for plant disease recognition. Following this idea, PlantCLEF2022, a plant-related dataset with 2,885,052 images and 80,000 classes, was adopted for our paper.

Second, a model with higher performance in ImageNet or a source dataset may have a better performance in the target dataset with a transfer learning strategy (Kornblith et al., 2019). Convolution neural networks (CNN) (Krizhevsky et al., 2012; He et al., 2016) achieved the best accuracy for the ImageNet validation dataset. Simultaneously, the attention mechanism has been leveraged to boost the performance of plant disease recognition (Yang et al., 2020; Qian et al., 2022; Zhao et al., 2022). In recent years, Vision Transformer (ViT) (Dosovitskiy et al., 2020), a general model of attention mechanism, has become a hot topic in the computer vision community and outperforms CNN-based models. For example, MAE (He et al., 2022) scores 85.9 inaccuracy for the ViT-L model which is higher than Resnet50 and ResNet152 with scores of 79.26 and 80.62, respectively. Therefore, for plant recognition, ViT-based models with a transfer learning strategy are promising but still underdeveloped (Wang et al., 2022).

Third, the supervised loss function inevitably pushes the model to learn source task-related features that may not be helpful for the target task (Wu et al., 2018). In contrast, the self-

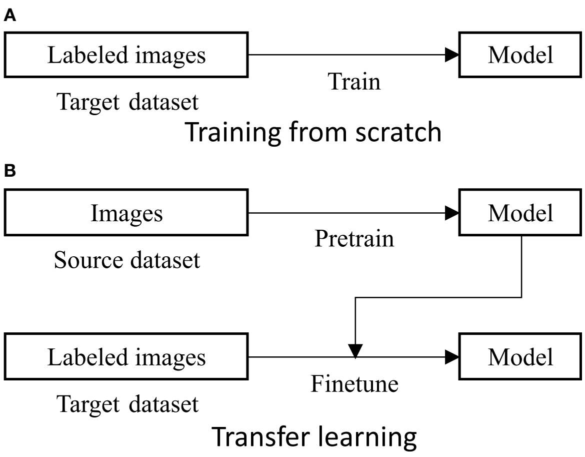


FIGURE 1
Training from scratch **(A)** and transfer learning **(B)**. Three key factors in transfer learning are the source dataset, the model, and the loss function to pre-train the model. These have all been undeveloped in plant disease recognition.

supervised loss function eases the issue by introducing a pretext task, such as contrast loss (Wu et al., 2018) and reconstruction loss (He et al., 2022). Thus, a ViT mode pre-trained in the PlantCLEF2022 dataset with a self-supervised loss function is assumed to be better than the current popular transfer learning strategy that is pre-trained on a CNN-based model in the ImageNet dataset with a supervised loss function (Mohanty et al., 2016; Yang et al., 2020; Abbas et al., 2021; Fan et al., 2022; Yadav et al., 2022).

Besides, the transfer learning strategy is slightly problematic when considering computing devices and the large-scale PlantCLEF2022 dataset. To be more specific, training a ViT model 800 epochs in PlantCLEF2022 as MAE (He et al., 2022) requires more than five months with four RTX 3090 GPUs. To reduce the computing cost, we utilize a dual transfer learning strategy, where a public ViT model pre-trained in ImageNet with a self-supervised loss function is trained in the PlantCLEF2022 dataset with a supervised loss function. In this way, we only spend about 15 days training the model in PlantCLEF2022. We emphasize that our dual transfer learning is different from (Azizi et al., 2021; Zhao et al., 2022) due to the following facts, aiming to reduce the cost of pre-training a model, large-scale PlantCLEF2022 dataset, and employing a ViT-based model.

To summarize, our paper will make the following contributions:

- We propose a novel transfer learning to achieve versatile plant disease recognition with a plant-related source dataset PlantCLEF2022, ViT model, and self-supervised learning to pre-train the model.
- We utilize dual transfer learning to save computation costs, considering the large-scale PlantCLEF2022 dataset.
- We validate our method in 12 plant disease datasets and our method surpasses the current widely used strategy by a large margin. Specifically, we score an average testing accuracy of 86.29 in a 20-shot case, 12.76 higher than the widely used strategy.
- Our transfer learning strategy also outperforms other methods in one plant growth stage prediction and one plant weed recognition, which suggests that our strategy contributes beyond plant disease recognition.

2 Material and method

2.1 Plant disease datasets

To validate the generalization of transfer learning and deep learning, we executed our method in fourteen public datasets, thirteen related to plant disease recognition. To be more specific, we used PlantVillage (Hughes et al., 2015), PlantDocCls (Singh et al., 2020), Cassava (Ramcharan et al., 2017), Apple2020 (Thapa et al., 2020), Apple2021 (Thapa et al., 2021), Rice1426 (Rahman et al., 2020), Rice5932 (Sethy et al., 2020), TaiwanTomato[2], IVADLTomato and IVADLRose[3], CitrusLeaf (Rauf et al., 2019), CGIARWheat[4], and PDD271* (Liu et al., 2021). More details of the datasets are shown in Table 1 while three random images for each class are displayed here[5].

The datasets are considered from several viewpoints. Figure 2 gives a glance at some images in the datasets. First is the *number of images and the number of classes*. Generally, the more classes and fewer images, the more difficult the recognition task. PDD271 covers 271 classes, including fruit trees, vegetables, and field crops, but unfortunately, it is not public. Only ten samples for each class are available and therefore, we adopted it as a few-shot learning task. In contrast, most of the public datasets only involved one type of plant, such as rice (Rahman et al., 2020; Sethy et al., 2020) or apple (Thapa et al., 2020; Thapa et al., 2021). Besides, the number distribution of classes may cause class-imbalance trouble, in which the trained model may have higher performance for the class with a dominant number of images in the training stage. Second, *the conditions the images were taken in* matters since controlling the conditions reduces the variation in the collected images, such as background and illuminations. A previous work (Barbedo, 2019) proves that controlling the conditions or masking the background out can improve recognition performance. Third, the *organs* of plants in images are also important. The main organs in the datasets are leaves, but also include some fruits, stems, and whole plants. Interestingly, different leaves of plants have heterogeneous shapes that may result in various performances with the same model. For example, the leaves of cassava are far different from their counterparts in apple and tomato plants. Especially, some images in PDD271 are captured with part of a leaf, not the whole leaf as in PlantVillage. Fourth, the *scale* of the images is also essential to the performance. The scale is related to the distance between the camera and the plant when taking pictures. For example, the leaves in PlantVillage and Apple2020 have a similar scale while the images in Rice1426 are on different scales. Fifth, *image size*,i.e. height and width, may incur challenges for recognition tasks as the disease phenomenon may not be clear enough in small-size images. To summarize, we emphasize that image variations (Xu et al., 2022a) in the dataset have an influence on training models and their corresponding

[2] https://data.mendeley.com/datasets/ngdgg79rzb/1

[3] https://github.com/IVADL/tomato-disease-detector

[4] https://www.kaggle.com/datasets/shadabhussain/cgiar-computer-vision-for-crop-disease?resource=download

[5] https://github.com/xml94/MAE_plant_disease/blob/main/visualize_dataset/dataset.md

TABLE 1 Information of the used plant disease recognition datasets.

Dataset	Images	Classes	Highlights
PlantVillage	54,305	38	Covers 14 types of plants. Each image is taken in controlled conditions and only includes one leaf in the center. Some diseases are spilt into two cases according to their severities, early and late. Each class has more than 273 images. All images are the same height and width, 256*256.
PlantDocCls	2,576	27	Includes 13 plants. The images are collected from the Internet with diverse heights and widths and most of the images are taken in real field conditions. The original training and testing dataset include 2,340 and 236 images, respectively.
Cassava	21,397	5	The images are taken in real field conditions and thus have wide variations, such as background, illumination, and leaf scales. All images have the same height and width, 800*600.
Apple2020	3,642	4	Taken in real field conditions. One leaf may include more than one type of disease and those images are labeled as one class. All images are the same size, 2048*1365.
Apple2021	18,632	6	An updated version of Apple2020 but with 2 more classes. All images are the same size, 4000*2672.
Rice1426	1,426	9	Images are taken in both real filed and controlled conditions. The images are not just related to leaves, but also other organs, stems, and grains. Images are in 224*224 resolution.
Rice5932	5,932	4	Only includes rice leaf images with different scales. All images are resized to 300*300.
TaiwanTomato	622	5	One image may include one or multiple leaves taken in either controlled conditions or real field conditions. There are 495 and 127 images in the original training and testing dataset, respectively. All images are resized to 227*227.
IVADLTomato	3,021	9	The original dataset includes more images in an unbalanced way. We limited the number for each class to less than 520. The original images have a large height and width, and we resized the images to 520*520 to save disk space.
IVADLRose	3,132	6	Similar to IVADLTomato, we limited the number for each class and resized the images.
CitrusLeaf	609	5	Images are taken in controlled conditions and resized to 256*256. We only used the leaf parts from the original Citrus dataset.
CGIARWheat	876	3	Includes leaves, stems, and whole plants. Images are taken from different viewpoints with diverse distances and different image sizes.
PDD271*	2,710	271	Covers fruit trees, vegetables, and field crops, with huge image variations. Ten images for each class are available as samples.

performance, and thus, recognizing the image variations is significant to understanding the dataset.

2.2 PlantCLEF2022 dataset

PlantCLEF2022[6] was originally a challenge to identify the plant species based on their images. The trusted training dataset, PlantCLEF2022, annotated by human experts with 2,885,052 images and 80,000 classes, is leveraged and used as the default PlantCLEF2022 dataset in this paper. Each class in the dataset is limited to no more than 100 images and has 36.1 images on average. As shown in Figure 3, the images cover plant habitat (environment or background) and organs such as the leaf, fruit, bark, or stem. Essentially, plants can be recognized based on multiple pieces of visual evidence, instead of only one piece of evidence (Xu et al., 2022c). Besides, the images belonging to one class embrace huge variations. As displayed in Figure 4, the variations include background, illumination, color, scale, and image size.

Why PlantCLEF2022? We recognize that three characteristics make PlantCLEF2022 beneficial to plant disease recognition with transfer learning strategy, i.e., *plant-related, large-scale, and wide variations*. First, it is accepted that a large-scale related source dataset contributes to the target task. As the PlantCLEF2022 dataset is plant-related and on a large scale, even when compared to ImageNet (Deng et al., 2009), it can be beneficial to plant disease recognition and related tasks, such as growth stage prediction. Second, the PlantCLEF2022 dataset has wide variations as mentioned before, by which we can learn a better feature space when using it to pre-train a model. Arguably, the variations in PlantCLEF2022 are much stronger than all of the plant disease datasets introduced in Section 2.1. We have noticed that finding this kind of dataset for plant disease cognition tasks is one of the main interests in recent years. In the beginning, ImageNet made a significant contribution as a source dataset. Recently, the AI Challenger dataset, a little bit bigger than PlantVillage but with small variations as most of the images are taken in controlled conditions, is considered as a source dataset (Zhao et al., 2022). Although it is plant-related, the AI Challenger dataset is far behind when compared to PlantCLEF2022 because of its number of images and classes and poor image variations.

2.3 Dual transfer learning

To achieve versatile plant disease recognition with a limited dataset, we believe that, under the transfer learning paradigm, a large-scale related dataset, PlantCLEF2022, and a powerful model are beneficial. Hence, we designed a dual transfer learning model, taking the computation load and device into

[6] https://www.aicrowd.com/challenges/lifeclef-2022-plant

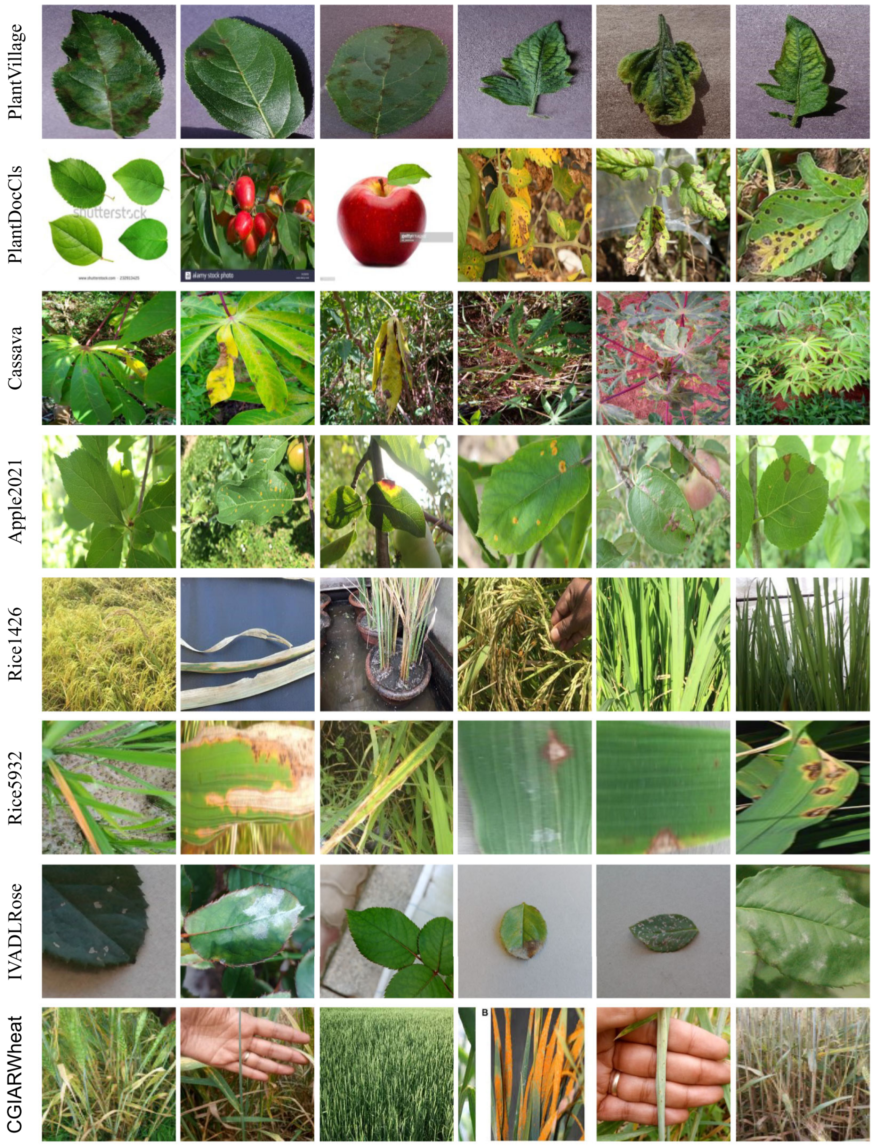


FIGURE 2
Image examples from different datasets. We recognize that there are image variations [40], such as background, the shape of leaves, illumination, and scale.

consideration. As shown in Figures 5A, C, our transfer learning consists of three steps with transfer learning occurring twice.

In the first step, a vision transformer (ViT) model is pre-trained with the ImageNet (Deng et al., 2009) in a self-supervised manner, reconstruction loss. We emphasize here that we directly adopted the pre-trained model from masked autoencoder (MAE) (He et al., 2022), instead of training the model ourselves. Simultaneously, we argue that superior pre-trained models are essential for better plant disease recognition, even if the models have the same architecture. The experiments in the following section prove that the original pre-trained ViT model (Dosovitskiy et al., 2020) performs worse than MAE (He et al., 2022). As shown in Figure 6, MAE is a composite of an encoder and a decoder that are optimized by a reconstruction loss, $\mathcal{L}_{recon}=||input, target||_2$ where *input* is the original image and *target* denotes the reconstructed image. During the training process, the original image *input* is split into several patches that are randomly blocked. The encoder aims to extract necessary information from the blocked image and the decoder is required to fill the blocked patches. As the optimization does not require labels, it falls under self-supervised learning.

The decoder in MAE is discarded and the encoder is utilized in the second step, followed by a linear layer and a

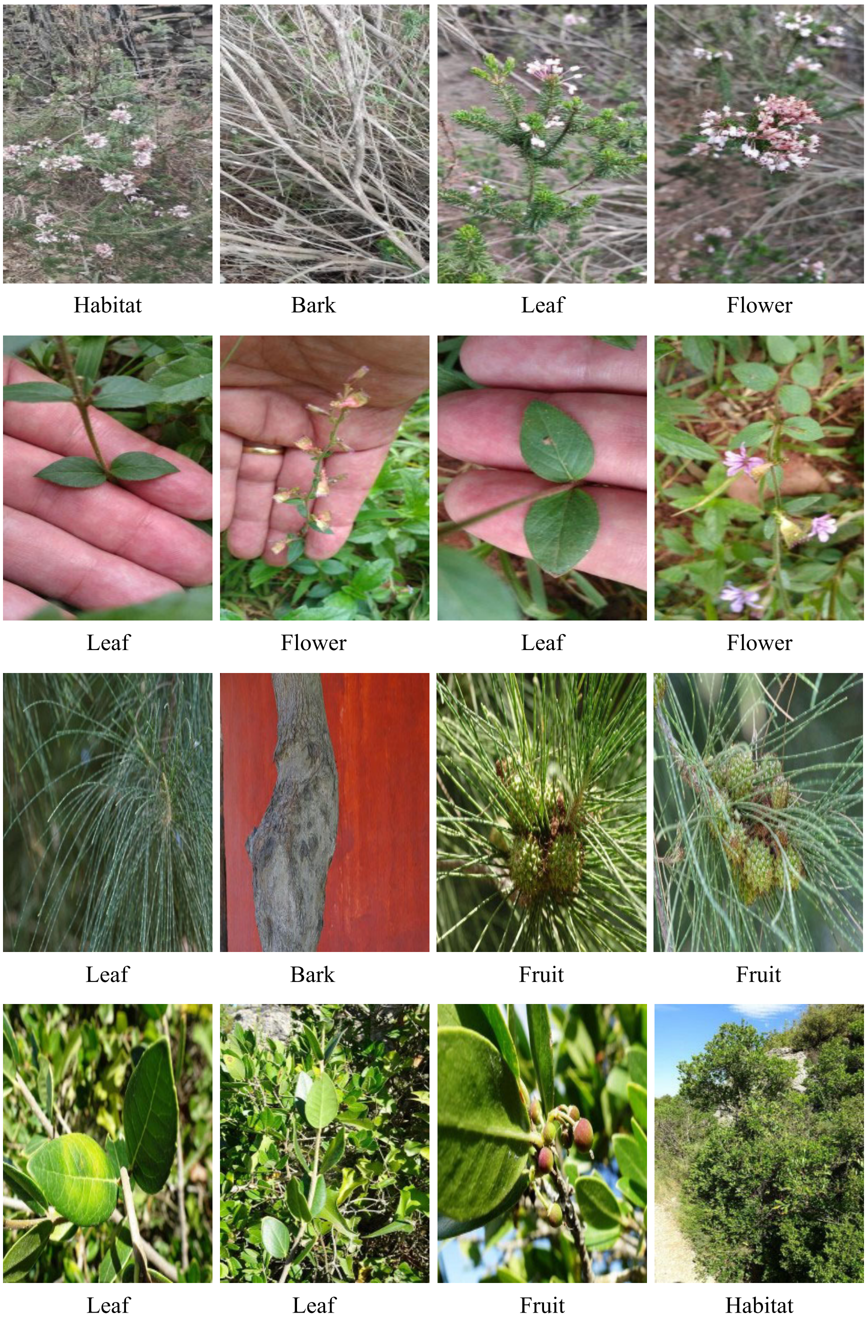


FIGURE 3
Different interests or organs in PlantCLEF2022 testing dataset.

softmax operation to do classification. The encoder and the added linear layer are fine-tuned in the PlantCLEF2022 dataset, optimized by the cross entropy loss, $\mathcal{L}_{ce}=-log(p(y_j))$ where j is the ground truth index and $p(y)$ is the output of softmax operation. Different from the first step, the input is not split into patches and blocked. The main characteristic of the second step is the PlantCLEF2022 dataset, related to the plant disease recognition dataset. We highlight that the second step is outlined and trained in our previous paper (Xu et al., 2022c) for the PlantCLEF2022 challenge and thus is not outlined and trained in this paper.

In the third step, the added linear layer in the second step is replaced by a new linear layer. To be clear, the encoder and the new linear layer in this step are fine-

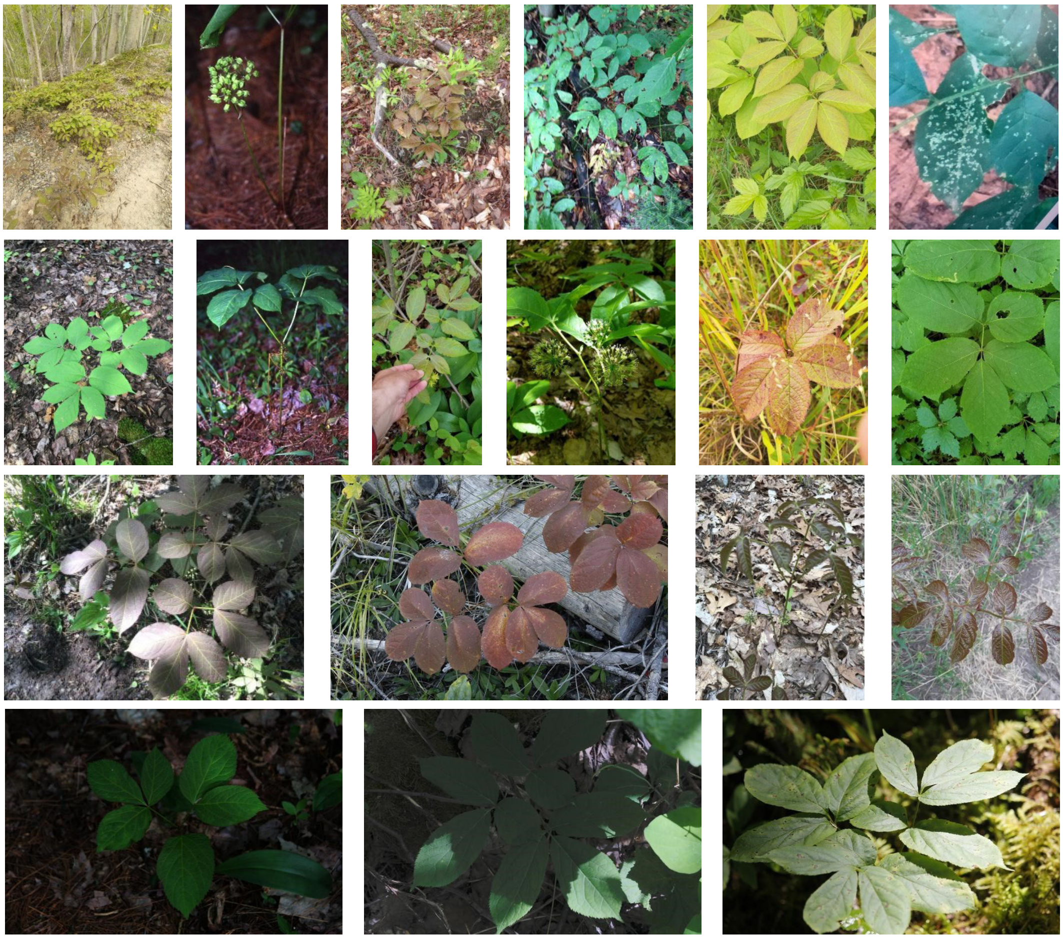


FIGURE 4
Images of Aralia Nudicaulis L. species from PlantCLEF2022 dataset. The images from the same plant species are heterogeneous in the background, illumination, color, scale, etc.

tuned in a specific plant disease recognition dataset. The cross-entropy loss is again utilized to optimize the whole network. As mentioned before, the first and second steps are executed in other papers and thus only the third step is required for this paper. We have termed our strategy dual transfer learning since the model is trained with two other datasets and transferred twice.

We believe that the first step is not mandatory for better performance in versatile plant disease recognition but contributes to the reduction of the training time for the whole system. As shown in Figure 5B, we can pre-train a model in the PlantCLEF2022 dataset and then fine-tune it for the plant disease dataset. Unfortunately, this setting may entail a long training epoch in PlantCLEF2022 to have a better performance, such as 800 epochs in MAE (He et al., 2022). In contrast, we only train 100 epochs for the second step and hence can save time. Besides, by training an MAE model in a self-supervised way, one decoder is trained at the same time which needs more time for one epoch. Therefore, our dual transfer learning reduces training time *via* utilizing the public model from MAE (He et al., 2022).

3 Experiment

3.1 Experimental settings

Dataset. For each original dataset in Table 1, we split them into training, validation, and testing datasets. The training dataset is leveraged to train the models while the validation one is only used to choose the best-trained model from different epochs. Then, the best model is evaluated in the testing dataset. If there is a testing dataset with annotations in the original dataset, we directly used the original testing dataset. Otherwise, the whole original dataset is split into training, testing, and validation datasets in different percentages or an exact number

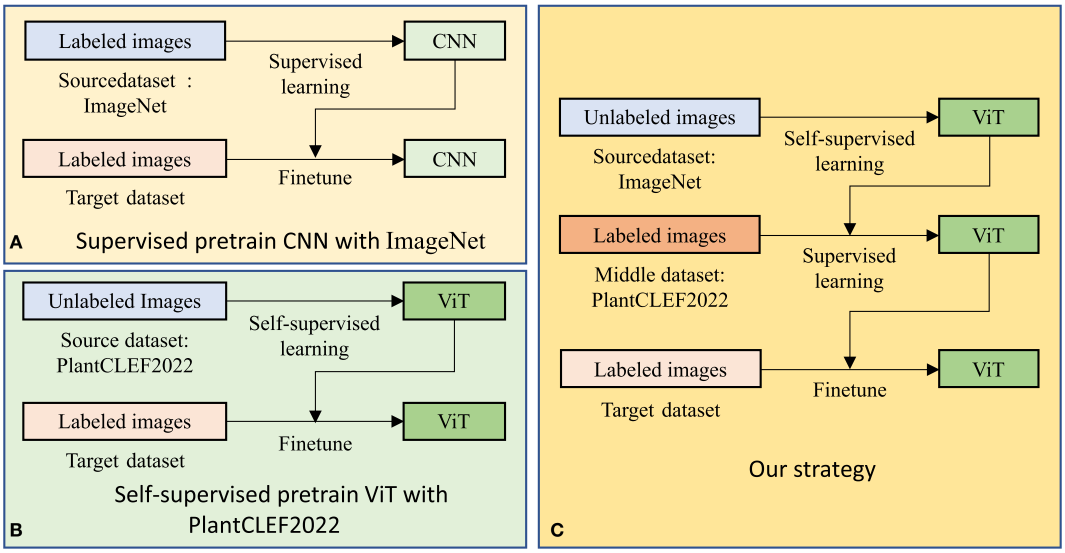


FIGURE 5
Transfer learning strategies for plant disease recognition. Our strategy differs from the current popular transfer learning strategy **(A)** in the source dataset, model, and loss function. Furthermore, we adopt dual transfer learning **(C)** to save computation time by utilizing the public pre-trained model, compared to **(B)**.

of images. To be more specific, the original testing datasets in PlantDocCls and TaiwanTomato are directly used while a new testing dataset is made for other datasets.

For each plant disease dataset, we consider two training cases, generic and few-shot cases. Different percentages of the training dataset are utilized in the generic case, such as 20% and 40%, while only several images for each class are taken to train the model in the few-shot case. To summarize, we set eight dataset modes, as shown in Table 2, four percentages as training in generic cases and 4 types of few-shot cases. Except for ratio80, 20% is taken for the validation and testing datasets for all experiments. The validation and testing datasets are the same for the generic and few-shot cases. Furthermore, the dataset splitting was randomly executed once only, by which the images of each dataset mode are fixed for all compared models or strategies. Although the percentage of validation and testing datasets is the same for most of the dataset modes, the images are different because of a different random process.

Comparison methods. To validate our method, we designed several comparisons with different strategies or models. To choose the compared methods, we held to the following features: with transfer learning or without transfer learning, CNN-based or ViT-based, supervised or self-supervised, and trained with PlantCLEF2022 or not. Simultaneously, we do not want to pre-train the models because of our lack of GPUs and the almost 3 million images in PlantCLEF2022. Based on these two ideas, the compared methods are described below and more interesting methods are listed in Table 3 with their corresponding characteristics.

- RN50. A ResNet50 model is trained from scratch with the target datasets shown in Table 1.
- RN50-IN. A ResNet50 model is pre-trained with the ImageNet (IN) dataset in a supervised way and then fine-tuned in the target datasets.
- MoCo-v2. A MoCo-v2 model is pre-trained with the ImageNet dataset in a self-supervised way and then fine-tuned in the target datasets.
- ViT. A ViT-large (Dosovitskiy et al., 2020) model is trained from scratch with the target datasets.
- ViT-IN. A ViT-large model is pre-trained with the Imagenet dataset in a supervised way and then fine-tuned in the target datasets.
- MAE. A ViT-large model is pre-trained with the ImageNet dataset in a self-supervised way. Specifically, MAE (He et al., 2022) uses reconstruction loss to learn better performance with a high occlusion.
- Our model. We fine-tuned a ViT model from MAE with the PlantCLEF2022 dataset and then fine-tuned it again with the target datasets.

We noticed that there were several other possible strategies. For instance, it is interesting to directly pre-train a ViT model with only the PlantCLEF2022 dataset in a self-supervised manner, no ImageNet, shown as Case 8 in Table 3. Further, pre-training an RN50 model with the PlantCLEF2022 dataset in a self-supervised manner is also encouraged to distinguish the impact of convolution neural networks (CNNs) and vision transformers (ViTs), shown as Case 3 in Table 3.

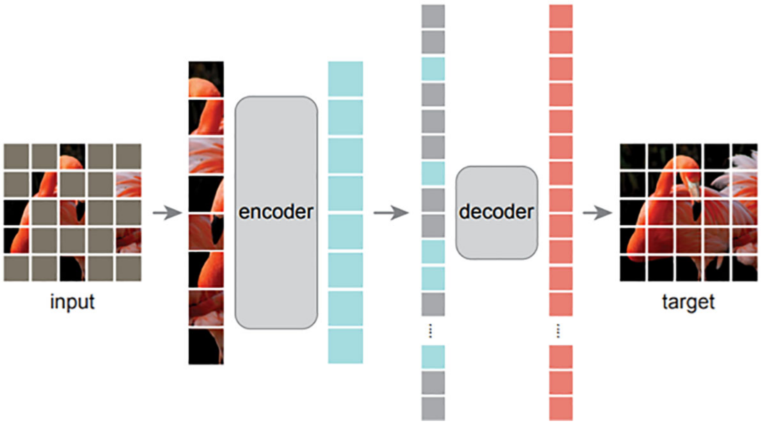


FIGURE 6
The high-level architecture of MAE [13]. With MAE, an image is split into patches that are then randomly blocked. The unblocked patches are fed to an encoder, followed by a decoder to reconstruct the whole input image. After the unsupervised pre-training, the decoder is discarded and only the encoder is utilized in the downstream task. The input is not blocked and a specific classifier is added after the encoder when fine-tuning the model in a target task.

Simultaneously, fine-tuning a MoCo-v2 model in the PlantCLEF2022 dataset is also inspired to see the difference between CNN and ViT, shown as Case 5 in Table 3, even if we expect a lower performance because MoCo-v2 has a lower accuracy in ImageNet than MAE. However, training these models is too expensive. It is estimated that pre-training a ViT-large model as MAE costs more than *five* months with our current computation devices, four RTX 3090 GPUs. Therefore, these possible strategies are left for future studies.

Implementation details. As mentioned in Section 2.3, we have used the pre-trained ViT-L model from our previous paper (Xu et al., 2022c). Hence, we only focus on the last fine-tuning process in this paper, i.e. fine-tuning the ViT-L model in the plant disease recognition dataset. The ViT-L model has 24 transformer blocks with a hidden size of 1024, an MLP size of 4096, and 16 heads for each multi-head attention layer. The ViT-L model has approximately 307 million trainable parameters in total.

For a fair comparison, all models or transfer learning strategies were executed with the same settings with most of them following the fine-tuning schemes in MAE (He et al., 2022). In detail, the basic learning rate lr_b was 0.001, and the actual learning $lr_a = lr_b$ * *batch*/256 where *batch* was the batch size for different training dataset modes. The model was warmed up in 5 epochs with the learning rate increasing linearly from the first epoch to the set learning rate. Furthermore, 0.05 weight decay and 0.65 layer decay were utilized. Mixup (Zhang et al., 2017) and CutMix (Yun et al., 2019) were adopted as data augmentation methods.

The main change from MAE experimental setting was the batch size. Considering the number of images in each dataset, in the generic case, the batch size was 64 for CGIARWheat, Strawberry2021, CitrusLeaf, and TaiwanTomato, while it was 128 for other datasets. In terms of the few-shot case, the number of classes was one factor to set as the batch size should not be larger than the number of classes in the 1-shot case. Specifically, the batch size was 4 for most of the datasets, except for CGIARWheat with 2, IVADLTomato with 8, PlantDocCls with 16, PlantVillage with 32, and Rice1426 with 8. Besides, the generic case was trained with four GPUs while the few-shot cases were trained with only one GPU. To evaluate during thetraining process, the models were trained for 50 epochs and validated after every 5 epochs in the validation dataset, including the first epoch. The best models were tested in the testing datasets.

Evaluation metric. *Accuracy*, a common evaluation metric for image classification (Dosovitskiy et al., 2020; Xu et al., 2022b; He et al., 2022) was leveraged to assess different methods in a specific dataset. Since we aim to achieve versatile plant disease recognition performance, the *mean accuracy*, *mAcc*, over all datasets was utilized and computed as follows:

$$mAcc = \frac{1}{M}\sum_{i=1}^{M} Acc_i, \tag{1}$$

where Acc_i is the testing accuracy in the i-th dataset and N is the total number of datasets. To assess the generality, testing accuracy and mean testing accuracy was employed, instead of validation accuracy and mean validation accuracy as used in MAE (He et al., 2022). In general, high testing accuracy and mean testing accuracy were desired.

TABLE 2 The settings in different dataset modes for the original dataset without labeled testing dataset.

Dataset case	Dataset mode	Training	Validation	Testing
Generic case	Ratio20	20%	20%	20%
	Ratio40	40%	20%	20%
	Ratio60	60%	20%	20%
	Ratio80	80%	10%	10%
Few-shot case	1-shot	1	20%	20%
	5-shot	5	20%	20%
	10-shot	10	20%	20%
	20-shot	20	20%	20%

The splitting was random once only, by which the images of each dataset mode are fixed for all compared models or transfer learning strategies. Although the percentage of validation and testing dataset was the same for most of the dataset modes, the images are different because of a different random process.

TABLE 3 The characteristics of the compared methods.

Case	Name	Model	ImageNet	PlantCLEF2022
1	RN50	CNN	N/A	N/A
2	RN50-IN	CNN	Supervised	N/A
3	-	CNN	N/A	Self-supervised
4	MoCo-v2	CNN	Self-supervised	N/A
5	-	CNN	Self-supervised	Supervised
6	ViT	ViT	N/A	N/A
7	ViT-IN	ViT	Supervised	N/A
8	-	ViT	N/A	Self-supervised
9	MAE	ViT	Self-supervised	N/A
10	Ours	ViT	Self-supervised	Supervised

N/A denotes not available or not used. We evaluated the compared methods from these viewpoints: no pre-training process because of our lack of GPUs, and showing the impacts of the basic model (CNN orViT), supervised or self-supervised, plant-related dataset (ImageNet or PlantCLEF2022), and dual transfer learning strategy. The named methods are compared in our paper while the other methods are encouraged and left for future studies considering the availability of GPUs.

3.2 Experimental results

3.2.1 Main result

As our main objective was achieving versatile plant disease recognition with a limited dataset, we first compared our method to other strategies. Table 4 displays the mean testing accuracy of different methods over the 12 plant disease datasets mentioned in Table 1 and Figure 7 illustrates the tendency of mean testing accuracy of various methods in few-shot case and generic case respectively. The testing accuracy, the curve of validation loss, and the accuracy for each dataset can be found in the Supplementary Material. As shown in Table 4, the experimental results suggested that our method surpasses other methods by a clear margin across all dataset modes. Specifically, our method achieves 86.29 *mAcc* in a 20-shot case where only 20 images per class are utilized to train the models, compared to the second-best method, RN50-IN. We observed that the gap between our method and other methods becomes less when the number of training images increases. For example, the gap between our method and the second-best method,

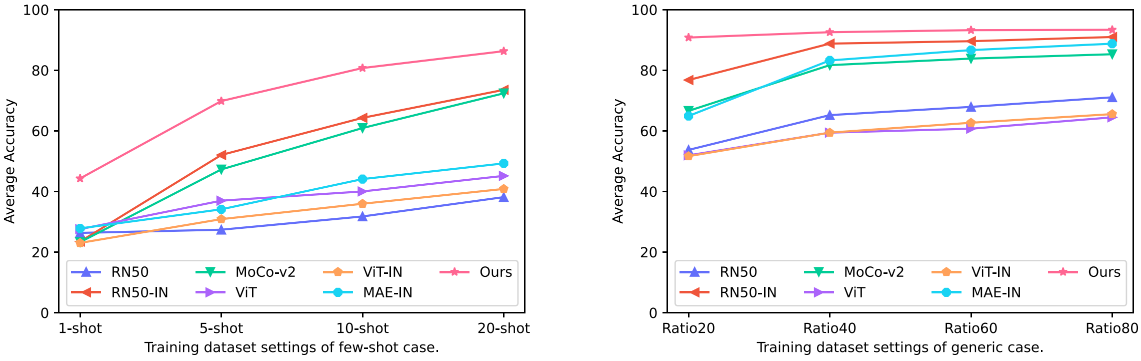


FIGURE 7
Curves of average testing accuracy *mAcc* of different methods in various training dataset modes over the 12 plant disease datasets.

TABLE 4 The mean testing accuracy *mAcc* of different training methods over the 12 datasets for plant disease recognition detailed in Table 1. .

	1-shot	5-shot	10-shot	20-shot	Ratio20	Ratio40	Ratio60	Ratio80
RN50	26.33	27.38	31.75	38.13	53.71	65.19	67.91	71.07
RN50-IN	23.46	52.03	64.28	73.53	76.77	88.78	89.58	90.97
MoCo-v2	23.28	47.27	60.93	72.38	66.58	81.68	83.84	85.28
ViT	27.56	36.96	40.01	45.14	51.93	59.40	60.71	64.46
ViT-IN	23.02	30.87	35.94	40.83	51.64	59.42	62.67	65.53
MAE	27.81	34.11	44.08	49.26	64.90	83.23	86.65	88.76
Ours	**44.28**	**69.83**	**80.73**	**86.29**	**90.79**	**92.55**	**93.23**	**93.34**

The best average accuracy for each dataset mode is in boldface.

RN50-IN, in Ratio20 is 14.02 and becomes 2.37 in Ratio80, which suggests that a limited training dataset is one main obstacle for current methods.

In terms of the impact of transfer learning, the CNN-based method, RN50-IN, has the second-best mean testing accuracy, much higher than its counterpart, RN50 training from scratch, in the target dataset. However, ViT-IN shows its inferiority for a limited training dataset while more training images lead to a minor increase. We postulate that ViT is harder to train than the original ViT-IN, as suggested in the original paper (Dosovitskiy et al., 2020). In contrast, CNN has been regularly developed in the last decade, and thus the optimizing problem has been largely mitigated. A similar phenomenon exists in the loss function to train the models. For example, MoCo-v2 (Chen X. et al., 2020) scores 71.1 top-1 in accuracy in ImageNet while RN50 (He et al., 2016) obtains 77.15. On the contrary, MAE (He et al., 2022) achieves a 85.9 top-1 accuracy score. A comparison between ViT, ViT-IN, and MAE suggests that the self-supervised loss function contributes to the improvement of the ViT-based model in all training dataset modes.

Our method is based on MAE and is pre-trained one more time in the PlantCLEF2022 dataset. Excitingly, our method obtained 35.42, 36.65, and 37.03 higher accuracy scores than MAE in 5-shot, 10-shot, and 20-shot, respectively. The soar of the mean testing accuracy of our method compared to MAE proves that PlantCLEF2022 is essentially beneficial for achieving versatile plant disease recognition with a limited dataset. Our method not only achieved the best performance but also converged faster than other methods. For example, the validation loss was minimized to a low value within 5 epochs for the Ratio40 case. Please refer to Figures S1 and S2 in the Supplementary Material.

Finally, 10 images for each class are available in PDD271* (Liu et al., 2021) and we used them as a few-shot learning task. Our method achieved a testing accuracy of 81.9 with only 1,355 images for both training and testing, compared to the original accuracy of 85.4 with 154,701 and 21,889 images for training and testing (Liu et al., 2021).

3.2.2 Beyond plant disease

Beyond achieving versatile plant disease recognition, we believe that our transfer learning strategy is also beneficial for other types of plant-related work. We performed two types of experiments over two datasets. The Strawberry2021[7] dataset, designed to predict plant growth stages, such as the young leaves and flowering stages, includes 557 images and 4 classes. The CottonWeedID15 (Chen et al., 2022) dataset requires the model to distinguish 15 types of weed in a cotton field, with 5,187 images in total.

The mean testing accuracy is displayed in Table 5 while the details can be found in the Supplementary Material. It is interesting that our method scored a mean testing accuracy of 97.60 in a 5-shot case where only 5 images of each label were utilized to train the network. The current popular strategy obtains similar results but in the Ratio40 case, with approximately 121 images per class. The experimental results suggest that our method can also contribute to plant-related applications beyond plant disease recognition with few training samples.

3.2.3 Discussion

Limited data is one main challenge in achieving high performance in the computer vision field (Xu et al., 2022a) and plant disease recognition (Lu et al., 2022; Xu et al., 2022b). Through our experimental results, we argue that the required amount of training dataset is partly dependent on the model or pre-trained model. As shown in Table 4, the mean testing accuracy of RN50-IN was 83.23 in the Ratio40 case and gains 12.76 from the Ratio20 case, while our method only had a 1.76 increase. Through this analysis, we believe that our method mitigates the requirement of a large dataset for plant disease recognition.

Furthermore, we emphasized that more training data tends to contribute to high performance but the gains become lower when a decent performance is obtained. For example, 20 percent more data only resulted in an increase of 0.11 in mean testing accuracy score in the Ratio60 case with our strategy. Therefore, recognizing the limitation of increasing data is also essential for practical applications. Sometimes, we may have to resort to alternative ways to have higher performance, instead of just increasing the training dataset.

Future work. First, we emphasize here that we are not aiming to achieve the best performance with our method in

[7] https://aistudio.baidu.com/aistudio/datasetdetail/98233

TABLE 5 The mean testing accuracy of different training methods over Strawberry2021 and CottonWeedID15.

	1-shot	5-shot	10-shot	20-shot	Ratio20	Ratio40	Ratio60	Ratio80
RN50	20.50	21.75	26.45	35.95	39.90	68.90	66.90	78.25
RN50-IN	45.55	75.95	87.90	87.15	60.85	98.00	98.35	98.55
MoCo-v2	45.65	70.25	84.65	86.05	66.90	96.45	96.20	97.50
ViT	32.70	39.90	44.30	51.45	56.25	65.65	75.40	80.90
ViT-IN	27.20	33.35	43.10	45.25	55.05	68.30	75.50	82.35
MAE	17.45	41.45	59.50	59.20	85.20	97.80	98.35	98.75
Ours	**73.90**	**97.60**	**97.55**	**97.85**	**99.80**	**99.35**	**98.80**	**99.70**

The best average accuracy for each dataset mode shows in boldface.

this paper. Instead, we propose a versatile plant disease recognition method with a limited training dataset. Therefore, we encourage our method to be used as a baseline for future works, although we did obtain superior performance in plant disease recognition. For example, is the PlantCLEF2022 dataset beneficial for a CNN-based network? In this way, we can pre-train the RN50 model and then fine-tune it in the target dataset. Moreover, it is interesting to analyze the reason why the same model and strategy behave differently in different datasets. For example, our method achieved a score of 97.4 in testing accuracy in the 20-shot case in the PlantVillage dataset as shown in Table S1 while scoring only 63.8 in the IVADLTomato dataset as shown in Table S9. Furthermore, we only validated our method in plant disease recognition, and encourage deploying our method to perform object detection and segmentation (Xu et al., 2022b). We also highlight combining our transfer learning with other unsupervised or self-supervised learning in the future. For instance, using a few labeled images to train a model and then leveraging the trained model to generate pseudo labels for unlabeled images (Li and Chao, 2021) and reduce annotation cost. Our preliminary results in Strawberry2021 and CottonWeedID15 suggest that our transfer learning strategy is not just promising for plant disease but also plant stage recognition and weed identification. We encourage more plant-related applications to deploy our method as a baseline.

4 Conclusion

We proposed a simple but nontrivial transfer learning strategy to achieve versatile plant disease recognition with limited data. Our method strikingly outperforms current strategies, not only on 12 plant disease recognition datasets but also in one plant growth stage prediction and one weed detection dataset. One main characteristic of our method is the use of PlantCLEF2022, a plant-related dataset including 2,885,052 images and 80,000 classes with huge image variations, which enables our transfer learning to be beneficial for versatile plant disease recognition tasks. Considering the large-scale dataset, our method employs a vision transformer (ViT) model because of its higher performance than the widely used convolution neural network. To reduce the computation cost, dual transfer learning is leveraged as the ViT model is first pre-trained with ImageNet in a self-supervised manner because the ImageNet dataset is different to the plant disease dataset. The model is then fine-tuned with PlantCLEF2022 in a supervised manner. We believe that our transfer learning strategy contributes to the field and to fuel the community, our codes and the pre-trained model are publicly available.

Data availability statement

Publicly available datasets were analyzed in this study. Their download links can be found here: https://github.com/xml94/MAE_plant_disease.

Author contributions

MX: conceptualization, methodology, software, writing - original draft, writing - review and editing. SY: supervision and writing - review and editing. YJ: writing - review and editing. DP: supervision, project administration, funding acquisition, writing - review and editing. All authors contributed to the article and approved the submitted version.

Funding

This research is partly supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No.2019R1A6A1A09031717), supported by the National Research Foundation of Korea(NRF) grant funded by the Korean government (MSIT). (NRF-2021R1A2C1012174), and supported by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, and Forestry (IPET) and the Korea Smart Farm R&D Foundation (KosFarm) through the Smart Farm Innovation Technology Development Program, funded by the Ministry of Agriculture, Food and

Rural Affairs (MAFRA) and the Ministry of Science and ICT (MSIT), Rural Development Administration (RDA) (No. 421005-04).

Acknowledgments

We appreciated the valuable suggestions from the reviewers to make the paper clear and easier to follow.

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.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.1010981/full#supplementary-material

References

Abade, A., Ferreira, P. A., and de Barros Vidal, F. (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

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

Afifi, A., Alhumam, A., and Abdelwahab, A. (2020). Convolutional neural network for automatic identification of plant diseases with limited data. *Plants* 10, 28. doi: 10.3390/plants10010028

Azizi, S., Mustafa, B., Ryan, F., Beaver, Z., Freyberg, J., Deaton, J., et al. (2021). "Big self-supervised models advance medical image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, Montreal: IEEE. 3478–3488.

Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* 180, 96–107. doi: 10.1016/j.biosystemseng.2019.02.002

Chen, J., Chen, J., 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

Chen, X., Fan, H., Girshick, R., and He, K. (2020). Improved baselines with momentum contrastive learning. *arXiv. preprint. arXiv:2003.04297*.

Chen, D., Lu, Y., Li, Z., and Young, S. (2022). Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. *Comput. Electron. Agric.* 198, 107091. doi: 10.1016/j.compag.2022.107091

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 Beach: Ieee), 248–255.

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," in *International conference on learning representations*.

Egusquiza, I., Picon, A., Irusta, U., Bereciartua-Perez, A., Eggers, T., Klukas, C., et al. (2022). Analysis of few-shot techniques for fungal plant disease classification and evaluation of clustering capabilities over real datasets. *Front. Plant Sci.* 295. doi: 10.3389/fpls.2022.813237

Fan, X., Luo, P., Mu, Y., Zhou, R., Tjahjadi, T., and Ren, Y. (2022). Leaf image based plant disease identification using transfer learning and feature fusion. *Comput. Electron. Agric.* 196, 106892. doi: 10.1016/j.compag.2022.106892

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, New Orleans: IEEE. 16000–16009.

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*, Caesars Palace: IEEE. 770–778.

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*.

Kim, B., Han, Y.-K., Park, J.-H., and Lee, J. (2021). Improved vision-based detection of strawberry diseases using a deep neural network. *Front. Plant Sci.* 11, 559172. doi: 10.3389/fpls.2020.559172

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., et al. (2020). "Big transfer (bit): General visual representation learning," in *European Conference on computer vision* (Springer), 491–507.

Kornblith, S., Shlens, J., and Le, Q. V. (2019). "Do better imagenet models transfer better?," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach: IEEE. 2661–2671.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, Lake Tahoe vol. 25. Eds. F. Pereira, C. Burges, L. Bottou and K. Weinberger (Curran Associates, Inc).

Li, Y., and Chao, X. (2021). Semi-supervised few-shot learning approach for plant diseases recognition. *Plant Methods* 17, 1–10. doi: 10.1186/s13007-021-00770-1

Liu, X., Min, W., Mei, S., Wang, L., and Jiang, S. (2021). Plant disease recognition: A large-scale benchmark dataset and a visual region and loss reweighting approach. *IEEE Trans. Image. Process.* 30, 2003–2015. doi: 10.1109/TIP.2021.3049334

Lu, Y., Chen, D., Olaniyi, E., and Huang, Y. (2022). Generative adversarial networks (gans) for image augmentation in agriculture: A systematic review. *Comput. Electron. Agric.* 200, 107208. doi: 10.1016/j.compag.2022.107208

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

Ngugi, L. C., Abelwahab, M., and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition–a review. *Inf. Process. Agric.* 8, 27–51. doi: 10.1016/j.inpa.2020.04.004

Olaniyi, E., Chen, D., Lu, Y., and Huang, Y. (2022). Generative adversarial networks for image augmentation in agriculture: a systematic review. *arXiv. preprint. arXiv:2204.04707*.

Pan, S. J., and Yang, Q. (2009). A survey on transfer learning. *IEEE Trans. Knowledge. Data Eng.* 22, 1345–1359. doi: 10.1109/TKDE.2009.191

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.* 1154. doi: 10.3389/fpls.2022.864486

Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., et al. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* 194, 112–120. doi: 10.1016/j.biosystemseng.2020.03.020

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

Rauf, H. T., Saleem, B. A., Lali, M. I. U., Khan, M. A., Sharif, M., and Bukhari, S. A. C. (2019). A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data Brief* 26, 104340. doi: 10.1016/j.dib.2019.104340

Sethy, P. K., Barpanda, N. K., Rath, A. K., and Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* 175, 105527. doi: 10.1016/j.compag.2020.105527

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020). "Plantdoc: a dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad: ACM (Association for Computing Machinery). 249–253.

Thapa, R., Wang, Q., Snavely, N., Belongie, S., and Khan, A. (2021). The plant pathology 2021 challenge dataset to classify foliar disease of apples. doi: 10.1002/aps3.11390

Thapa, R., Zhang, K., Snavely, N., Belongie, S., and Khan, A. (2020). The plant pathology challenge 2020 data set to classify foliar disease of apples. *Appl. Plant Sci.* 8, e11390. doi: 10.1002/aps3.11390

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

Tripuraneni, N., Jordan, M., and Jin, C. (2020). On the theory of transfer learning: The importance of task diversity. *Adv. Neural Inf. Process. Syst.* 33, 7852–7862. doi: 10.5555/3495724.3496382

Wang, F., Rao, Y., Luo, Q., Jin, X., Jiang, Z., Zhang, W., et al. (2022). Practical cucumber leaf disease recognition using improved swin transformer and small sample size. *Comput. Electron. Agric.* 199, 107163. doi: 10.1016/j.compag.2022.107163

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City: ACM (Association for Computing Machinery). 3733–3742.

Xing, S., and Lee, H. J. (2022). Crop pests and diseases recognition using danet with tldp. *Comput. Electron. Agric.* 199, 107144. doi: 10.1016/j.compag.2022.107144

Xu, M., Yoon, S., Fuentes, A., and Park, D. S. (2022a). A comprehensive survey of image augmentation techniques for deep learning. *arXiv. preprint. arXiv:2205.01491*. Bologna

Xu, M., Yoon, S., Fuentes, A., Yang, J., and Park, D. S. (2022b). Style-consistent image translation: A novel data augmentation paradigm to improve plant disease recognition. *Front. Plant Sci.* 12, 773142–773142. doi: 10.3389/fpls.2021.773142

Xu, M., Yoon, S., Jeong, Y., Lee, J., and Park, D. S. (2022c). "Transfer learning with self-supervised vision transformer for large-scale plant identification," in *International conference of the cross-language evaluation forum for European languages* (Springer), 2253–2261.

Yadav, A., Thakur, U., Saxena, R., Pal, V., Bhateja, V., and Lin, J. C.-W. (2022). Afd-net: Apple foliar disease multi classification using deep learning on plant pathology dataset. *Plant Soil*, 477, 1–17. doi: 10.1007/s11104-022-05407-3

Yang, G., He, Y., Yang, Y., and Xu, B. (2020). Fine-grained image classification for crop disease based on attention mechanism. *Front. Plant Sci.* 11, 600854. doi: 10.3389/fpls.2020.600854

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, Long Beach: IEEE. 6023–6032.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization. *arXiv. preprint. arXiv:1710.09412*.

Zhao, X., Li, K., Li, Y., Ma, J., and Zhang, L. (2022). Identification method of vegetable diseases based on transfer learning and attention mechanism. *Comput. Electron. Agric.* 193, 106703. doi: 10.1016/j.compag.2022.106703

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 13 December 2022
DOI 10.3389/fpls.2022.1011499

Research on the identification and detection of field pests in the complex background based on the rotation detection algorithm

Wei Zhang[1,2], Xulu Xia[1], Guotao Zhou[3], Jianming Du[2], Tianjiao Chen[2], Zhengyong Zhang[2]* and Xiangyang Ma[4]*

[1]Institute of Physical Science and Information Technology, Anhui University, HeFei, China, [2]Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, China, [3]Technology Research and Development Center, Henan Yunfei Technology Development Co. LTD, Henan, China, [4]Harvesting and Processing Department, Liaoning Provincial Instiitute of Agricultural Mechanization, Shengyang, China

Check for updates

OPEN ACCESS

EDITED BY
Po Yang,
The University of Sheffield,
United Kingdom

REVIEWED BY
Jun Liu,
Weifang University of Science and Technology, China
Sf Dong,
University of Science and Technology of China, China

*CORRESPONDENCE
Zhengyong Zhang
zyzhang@iim.ac.cn
Xiangyang Ma
799419036@qq.com

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection,
a section of the journal
Frontiers in Plant Science

RECEIVED 04 August 2022
ACCEPTED 15 November 2022
PUBLISHED 13 December 2022

CITATION
Zhang W, Xia X, Zhou G, Du J, Chen T, Zhang Z and Ma X (2022) Research on the identification and detection of field pests in the complex background based on the rotation detection algorithm. *Front. Plant Sci.* 13:1011499.
doi: 10.3389/fpls.2022.1011499

As a large agricultural and population country, China's annual demand for food is significant. The crop yield will be affected by various natural disasters every year, and one of the most important factors affecting crops is the impact of insect pests. The key to solving the problem is to detect, identify and provide feedback in time at the initial stage of the pest. In this paper, according to the pest picture data obtained through the pest detection lamp in the complex natural background and the marking categories of agricultural experts, the pest data set pest rotation detection (PRD21) in different natural environments is constructed. A comparative study of image recognition is carried out through different target detection algorithms. The final experiment proves that the best algorithm for rotation detection improves mean Average Precision by 18.5% compared to the best algorithm for horizontal detection, reaching 78.5%. Regarding Recall, the best rotation detection algorithm runs 94.7%, which is 7.4% higher than horizontal detection. In terms of detection speed, the rotation detection time of a picture is only 0.163s, and the model size is 66.54MB, which can be embedded in mobile devices for fast detection. This experiment proves that rotation detection has a good effect on pests' detection and recognition rate, which can bring new application value and ideas, provide new methods for plant protection, and improve grain yield.

KEYWORDS

image recognition, object detection, rotation detection, pest detection, plant protection

1 Introduction

As the most populous country in the world, China's annual food demand is the most critical social and livelihood issue. In recent years, urbanization has been getting faster and faster with the rapid development of China's economy. The immediate problem with it is the reduction of the available agricultural area. In order to ensure that China's annual grain output can be maintained at 650 billion kg above, it is necessary to improve the efficiency of grain cultivation on limited land. Food production is related to many factors, such as climate, temperature, and humidity (Dayan, 1988). Among them, the most severe threat to food every year is the impact of pests and diseases (Guru-Pirasanna-Pandi et al., 2018). According to the Food and Agriculture Organization of the United Nations statistics, global food production will decrease by 10-16% annually due to the impact of pests and diseases. In China, surveys show that about 40 million tons of food are lost yearly (CCTV News,). The key to solving the problem of grain production is promptly predicting the early formation of pests and scientific control. Therefore, the most critical link is accurately identifying and detecting different pests.

In recent years, traditional machine learning technology has undergone revolutionary changes with the improvement of the computing power of graphics cards and the rapid development of computer software and hardware resources. More and more experts and scholars use their computing power in image recognition. Object detection is a branch of image recognition based on deep learning-based CNN algorithms. At present, CNN has made incredible breakthroughs in theoretical and practical experiments. Current object detection algorithms are divided into two stages and one stage. The main difference is that the second stage forms a series of target candidate boxes and classifies the samples according to the convolutional network; the first stage converts the regression box prediction into a regression problem and then performs regression and sample classification at the same time. The two-stage mainstream target detection algorithms are represented by RCNN (Girshick et al., 2014), Fast RCNN (Girshick, 2015), Faster RCNN (Ren et al., 2017), Cascade RCNN (Cai and Vasconcelos, 2018), and Mask RCNN (He et al., 2017). The mainstream detection algorithms in the first stage are represented by YOLO (Redmon et al., 2016; Redmon and Farhadi, 2017; Redmon and Farhadi, 2018; Bochkovskiy et al., 2020; Ge et al., 2021) series, SSD (Liu et al., 2016), and RetinaNet (Lin et al., 2017).

The development of rotating object detection with horizontal detection has also received more and more attention from researchers. Rotation detection algorithms are represented by R3Det (Yang et al., 2021), ReDet (Han et al., 2021), S2A-Net (Han et al., 2022) and so on. In real environments, most detection objects often appear irregularly, such as text scene recognition in real life (Liao et al., 2018) and ship detection in remote sensing image ports (Fu et al., 2018; Yang et al., 2018; Li et al., 2018). Under these conditions, achieving satisfactory results in horizontal detection is difficult. Based on horizontal detection, rotation detection adds object Angle prediction, which makes the application of rotation detection more extensive. This method can adapt to any Angle and shape transformation of object detection and has good robustness to object localization and classification detection. For example, Ma et al. (2022) used R3Det detection and identification for coastal intensive marine cages. The experimental results showed that the mean Average Precision (mAP) in circular and square cages reached 92.65% and 98.06%, respectively. Peng et al. (2021) applied the rotation detection algorithm to detect insulators in the power grid. The experiments show that R3Det can better determine the position of insulators and reduce economic losses.

Pests live in complex and changeable natural conditions with many species, and the growth patterns of different pests are pretty different. At the same time, some pests are tiny in size and have certain similarities in appearance, color, and other characteristics, making detection and identification difficult. Traditional crop pest detection relies on many experts' on-site observation, identification, and detection. On the one hand, such detection is time-consuming and labor-intensive. On the other hand, the crops have been seriously affected because many pests can be observed manually, and the best control period is missed. In recent years, the rapid development of target detection algorithms and supporting software and hardware in the field of deep neural network learning has brought the possibility of quick identification and detection of pests, which has extensively promoted the application and development of intelligent plant protection and precision agriculture. Many domestic and foreign scholars conduct computer vision research by processing pest images. For example, M.A. Ebrahimi et al. (2017) proposed to use a machine learning Support Vector Machines(SVM) classifier to detect crops and use SVM to use differential kernel functions to classify and detect greenhouse pests. Li et al. (2021) improved the TPest-RCNN network structure based on the Faster RCNN network. Its backbone uses the VGG16 network for feature learning and uses bilinear interpolation on the candidate coordinates instead of the ROIPool method to generate more accurate values. Finally, classification and coordinate regression correction predictions are performed. Experiments show that whiteflies' mAP reaches 95% under greenhouse conditions. Cho et al. (2007) collected three pests under greenhouse conditions and proposed using Prewitt for edge detection and counting. Solis-Sánchez et al. (Solis-Sánchez et al., 2011) an improved loss identification algorithm was used to detect six pests under greenhouse conditions.

However, most of the above detection methods mainly classify and identify a single pest image under greenhouse conditions, which has certain limitations in the actual natural environment. The current horizontal target detection network needs more pest training samples to obtain a better recognition rate when training multi-category pests. For example, Liu et al.

(2019) An improved convolutional neural network (CNN) and PestNet algorithm with a modular channel attention mechanism were proposed to evaluate 16 pests on 80k datasets MPD2018. The experiment proved that the result of mAP reached 75.46%. The improved convolution network and YOLOv4 network proposed by Tang et al. (2017) integrate attention mechanism and crosses-stage feature fusion to improve feature extraction and fusion capabilities. Experimental results on 28k data and 24 types of pests show that mAP and Recall achieved 71.6% and 83.5%, respectively. Wang et al. (2020) collected data on field pests to obtain 25k pictures with 24 categories and used different level detection algorithms to conduct comparative experiments. Finally, the mAP of YOLOv3 reached 59.37%. The level detection method in the above experiments is used for multi-category experimental research under large-scale data. It can be seen from the above that the horizontal detection method needs extensive data when detecting pests, which takes up many computer resources, and the final detection effect map is only about 75%, which can not reach the practical application value.

In this paper, a multi-target pest rotation detection method is proposed. Rotation detection is often used to detect objects with considerable lengths and widths and dense objects, such as ships in remote sensing ports (Fu et al., 2018; Li et al., 2018; Yang et al., 2018). Under the same circumstances, different pests or the same type of pests in motion obtained by the filming equipment will also be affected by different angles, and pests easily pile up densely. Therefore, it is difficult for the horizontal target detection algorithm to achieve a good recognition effect on small and dense targets. As shown in Figure 1, the target detection under shade environment level in training will be part of the other characteristics of objects of study, the recognition of samples have larger interference. The rotation detection algorithm can better fit the pest to the samples under the dense shadow, and the performance of the pest can achieve the effect of identifying different poses. This paper will compare the detection differences between different target detection algorithms and rotation detection in different situations to provide a reference for more agricultural pest detection in the future. The main research work of this paper is as follows: (1) Using a variety of horizontal and rotation detection algorithms to detect, identify, compare and analyze field pests. (2) It is concluded that the rotation detection algorithm is generally better than the horizontal detection algorithm in pest detection. The best representative algorithm of rotation detection is selected; (3) In this experiment, a pest rotation detection dataset (PRD21) of 21 pests under the horizontal frame and the rotating frame is constructed, and the difficulty of data detection is classified. It is hoped that the experiment will provide new ideas for accurately identifying pests and diseases and intelligent plant protection, which is conducive to the early and timely detection and prevention of pests and diseases and minimizes economic losses.

2 Materials and methods

2.1 Introduction to agricultural pests dataset-PRD21

This experiment ultimately needs to be detected in the natural environment, so the experiment's data are obtained through the detection and insect detection and reporting trapping equipment to get pest images under natural conditions. As shown in Figure 2A, the insect situation monitoring and reporting light device is placed in the actual natural environment to trap pests for 24 hours and automatically set to collect and take photos of pests through the camera in the

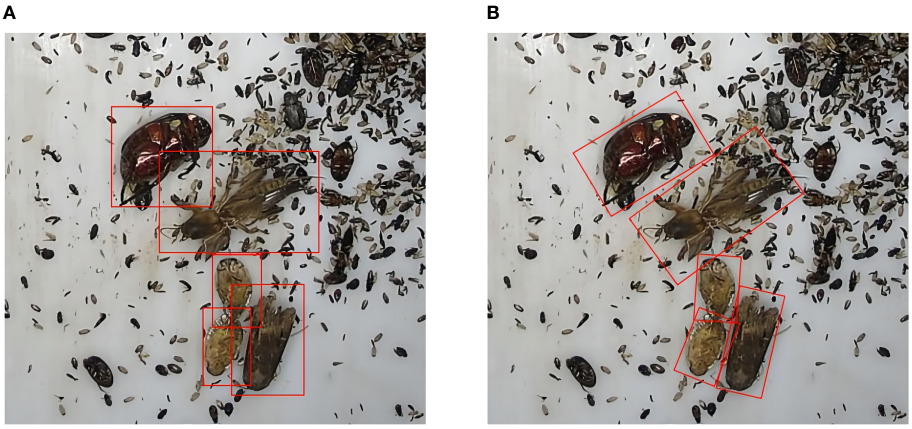


FIGURE 1
The training samples of horizontal algorithm and rotation detection algorithm are different. **(A)** is the horizontal frame More disturbed by other backgrounds, **(B)** is a rotating frame, which can better fit pest samples.

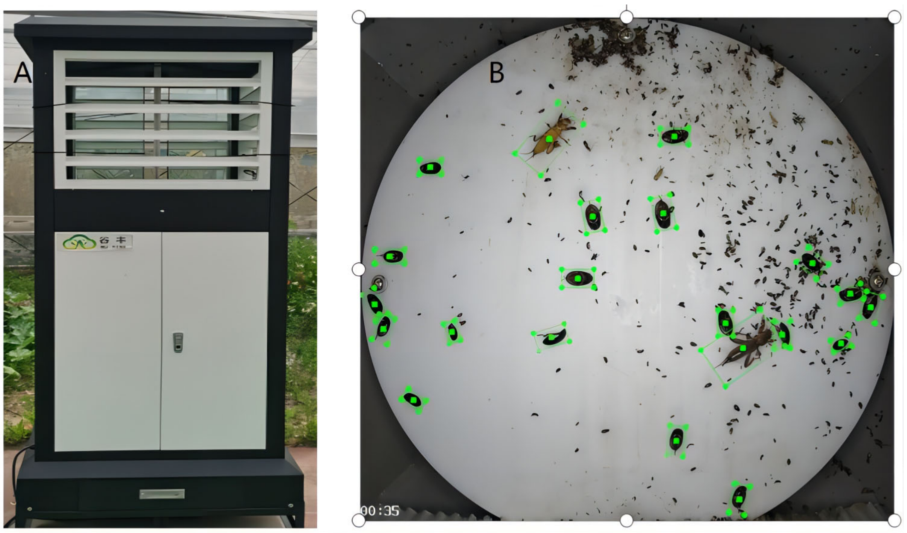


FIGURE 2
(A) is the detection and warning light device for collecting pests. **(B)** shows the collected pest samples.

machine every once in a while and upload them to the background database in time. Figure 2B shows the collected pest data samples for a certain period.

A total of 2398 pieces of valuable data were obtained in this dataset, and the image format was unified in JPG format with a resolution of 3840*2160 pixels. According to the pest classification of the Ministry of Agriculture of China and the number of data samples collected in the data set, it is divided into 21 types of pests (Wang et al., 2020). These data are processed into computer-trainable Pascal VOC (Everingham et al., 2010) type data, wherein agricultural experts and lableImg label software generate the training data set for level detection. The rotation detection data is generated by roLabelImg software. Finally, the datasets are divided into 1942 training sets, 216 validation sets, and 240 test sets according to the ratio of 8:1:1.The detected dataset is called Pest Rotate Detection(PRD21).

This paper aims to verify the generalization of the effect of rotation detection in different application scenarios. It is divided by the pest occlusion situation shown in Figure 3 shows the mutual shielding degree of pests in different environments. Figure 4 is the name of the specific separated different data sets, namely simple with no occlusion(SNO), simple with occlusion(SO), interference with no occlusion (INO), and interference with occlusion(IO). As shown in Table 1, the collected pest species, the pest area, and the relative size of the horizontal frame and the rotating frame are calculated according to Formula (1) and (2). Finally, Formula (3) calculates the severity of occlusion between pests.

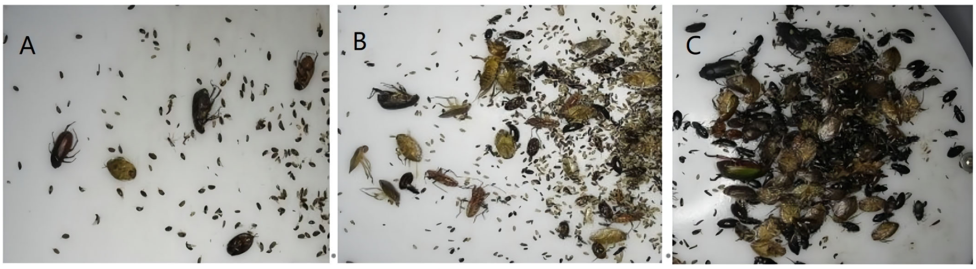


FIGURE 3
This figure shows the collection of different types of data. **(A)** refers to the occlusion of pests, **(B)** refers to the partial occlusion among pests, and **(C)** refers to the data type with serious occlusion.

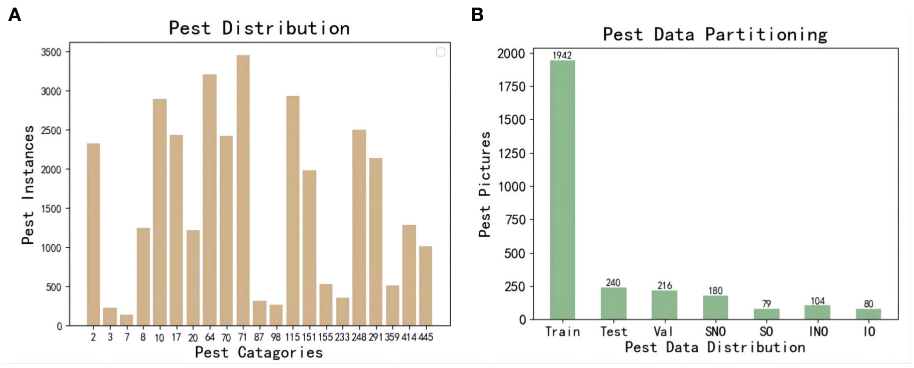


FIGURE 4
The number of pest instances and data set division. **(A)** is the number of instances in the data set, and **(B)** is the division of the training set.

$$HoReScale = \frac{1}{M}\sum_{M}^{1}(X_{\max} - X_{\min})*(Y_{\max} - Y_{\min})/C*100\,\% \quad (1)$$

$$\mathrm{RoReScale} = \frac{1}{M}\sum_{M}^{1}(\mathrm{w}*\mathrm{h})/\mathrm{C}*100\,\% \quad (2)$$

$$\alpha = area(GTBox_A \cap GTBox_B)/area(GTBox_A \cup GTBox_B) \quad (3)$$

Formula 1 is the area and relative proportion of the horizontal frame, and Formula 2 is the area and relative proportion of the rotating frame. C is the image's original size, and M is the total number of instances of a specific class. X_i is the horizontal relative position value of the corresponding pest, and Y_i is the vertical value of the corresponding pest. w and h are the width and height of corresponding pest coordinates. The function area() represents the area of the two pest objects,s A and B, ∩ where the two pest objects intersect and ∪ where the two pest objects are combined. α is the scaling factor, and its value is between 0 and 0.2. When α>0.1, it was considered that the two pests had severe shading; when α<0.1, it was supposed to be slightly shading. GTBox is the area of a single pest.

2.2 The algorithm model used is introduced

This experiment uses the horizontal box target detection one-stage algorithms RetinaNet, YOLOX, YOLOv5, YOLOv6, and two-stage algorithms Faster RCNN and Cascade RCNN for comparison experiments. Rotation detection includes ReDet, R3Det, Rotated Faster RCNN, and S2ANet as comparison algorithm models.

2.2.1 Introduction to algorithm models related to horizontal object detection

2.2.1.1 Faster RCNN introduction

This algorithm is an improved and optimized classic CNN convolution network algorithm. First, use the convolution layers for feature extraction to obtain feature maps and generate region proposals through Region Proposal Networks. The region of interest in Roi Pooling is extracted through feature maps and proposals, and the accurate location and category of the detection target are finally determined through the fully connected layer and bounding box regression.

2.2.1.2 Cascade RCNN introduction

This algorithm further optimizes the threshold setting in Faster RCNN, cascades multiple regressors and detectors with different thresholds, and continuously improves the threshold multi-cascade network structure iteratively. Ultimately, the accuracy of detecting target locations is maximized.

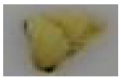
2.2.1.3 YOLOX introduction

As a single-stage target detection algorithm of the You Only Look Once(YOLO) series, positioning and classification are performed simultaneously. The generation method of anchor free is adopted to reduce the amount of calculation. The network structure mainly includes four parts, 1) Input: input image and perform data enhancement. 2) Backbone network (CSPDarknet53 (Wang et al., 2020)): Mainly used for feature extraction. 3) Neck: This layer uses Feature Pyramid Network (FPN) (Lin et al., 2017) and Path Aggregation Network(PAN) (Liu et al., 2018) as feature fusion. 4) Head: This layer predicts classification and location results.

2.2.1.4 YOLOv5 introduction

The network structure of the algorithm can be divided into four parts, the Input layer, the Backbone network, the Neck network, and the Prediction layer. The backbone network consists of Focus, CSP, and Spatial Pyramid Pooling module layers (Zhang et al., 2022). The Neck layer uses the residual network to improve the feature fusion ability. In the prediction layer, the loss of the regression box is calculated by GIoU Loss (Rezatofighi et al., 2019), and three different scale predictions are

TABLE 1 The species of pests and the proportion of relevant sizes.

Index	Pest name	Portrait	Ho Relative scale (%)	Ro Relative scale (%)	Index	Pest name	Portrait	Ho Relative scale (%)	Ro Relative scale (%)
2	Noctuidae		0.206	0.199	98	AnomalaexoletaFald		0.131	0.141
3	Athetis Lepigone		0.194	0.174	115	Diving Beetle		0.133	0.142
7	Spodoptera Litura		0.14	0.141	151	Cricket		0.236	0.219
8	Mole crickets		0.697	0.72	155	Sphaerodema Rustica Fabricius		0.155	0.157
10	Snout Moths		0.117	0.107	233	Spotted Red Bug		0.132	0.16
17	Helicoverpa Armigera		0.209	0.2	248	Marumba Gaschkewitschii		0.998	0.981
20	Oriental Armyworm		0.196	0.193	291	Carabidae		0.084	0.084
64	Holotrichia Parallela		0.192	0.194	359	Cockchafer		0.13	0.134
70	Anomala corpulenta Motschulsky		0.264	0.284	414	Turtle Shell		0.092	0.084
71	Coleopters		0.099	0.097	445	Metaboluo Impressifros Fairmaire		0.136	0.161
87	Tiger Beetle		0.085	0.083					

obtained, divided into 80×80, 40×40, and 20×20. The BCELogitsLoss function calculated Objectness-loss and Classification-loss. Finally, the best prediction results are selected according to three dimensions.

2.2.1.5 YOLOv6 introduction

As the latest algorithm of the YOLO series, many algorithm improvements have been made. Initially, the anchor-free method was used to generate the prediction frame and the same data enhancement as YOLOv5. The backbone network uses EfficientRep to replace the previous CSPDarknet for feature extraction. Neck built Rep-PAN based on Rep and PAN for feature fusion. The Head layer is decoupled in the same way as YOLOX, which separates the efficient structure of regression and category classification. The label assignment selection uses simOTA (Ge et al., 2021) to equalize the positive and negative samples. Finally, a new regression loss SIOU (Gevorgyan, 2022) is introduced to reduce the degree of freedom of regression to accelerate network convergence and further improve the accuracy of regression. From the above, we can be found that YOLOv6 combines the advantages of YOLOv5 and YOLOX.

2.2.1.6 RetinaNet introduction

As a one-stage target detection algorithm, the network structure is backbone using (vgg, resnet) for feature extraction, and then through Feature Pyramid Networks(FPN) to enhance the feature map of target area information for features of different scales, and finally predict the target frame in two FCN layers location and category. The main innovation of this structure is that Focal Loss is added to the one-stage detector to optimize the sample category imbalance problem, and anchor boxes are used to generate prediction boxes.

2.2.2 Introduction to algorithm models related to rotating target detection

2.2.2.1 ReDet introduction

When the traditional convolution network detects objects in any direction, it usually enhances the rotation data in the training samples, so the detection effect is poor, and more inclined models are required. The ReDet algorithm uses the equivariant rotation network combined with the detector to obtain the rotation features, uses the rotation invariant RiRoi Align space and the angle dimension to extract the features, and finally predicts the output.

2.2.2.2 S2ANet introduction

Due to the rotation detection network's rotation characteristics, sometimes the generated anchor box has a high degree of confidence, but there is still a significant dislocation in the instance fitting. To optimize this problem, S2A-Net adopts RetinaNet (Lin et al., 2017) as the backbone, plus FPN and component Feature Alignment Module (FAM) (Wang et al., 2019) and Oriented Detection Module (ODM) (Xie et al., 2021) modules for region selection and feature extraction fusion.

2.2.2.3 R3Det introduction

This experiment uses the R3Det rotation detection algorithm as a research method to compare other horizontal detection and rotation detection. The network structure is shown in Figure 5. The algorithm designed a refined one-stage accurate and a fast detector that combined the anchor points of the horizontal target detection algorithm and the anchor points of the rotation detection algorithm. The final effect significantly improved the adaptability of pest recognition in different scenes. Firstly, horizontal detection anchors are used to generate more candidate regions. Secondly, rotating anchors are used to optimize the dense target scene further. In the middle, the feature refinement module (FRM) (Yang et al., 2021) is used to refine and accurately process the predicted target locations. In order to achieve feature alignment, the algorithm uses Range non-maximum Suppression(RNMS) (Yang et al., 2021) instead of traditional non-maximum Suppression(NMS) (Neubeck and Van Gool, 2006). This part of the improvement method sets different filtering thresholds according to the number of samples and appearance characteristics of different pest categories. In terms of the loss function, the algorithm uses the approximate SkewIoU loss function, which can be pushed to calculate the multi-objective and multi-task rotation box. Further, it optimizes the problem of difficult identification of small objects and sample imbalance. The relevant calculation formulas are shown in the following (4-6).

$$SkewIoU = \frac{area(c1 \cap c2)}{area(c1 \cup c2)} \tag{4}$$

$$L_{loss} = \frac{\lambda_1}{S}\sum_{s=1}^{S} obj_s \frac{L_{reg}(v'_n, v_n)}{\left|L_{reg}(v'_n, v_n)\right|}\left|f(SkewIoU)\right| + \frac{\lambda_2}{S}\sum_{s=1}^{S} L_{cls}(p_s, t_s) \tag{5}$$

$$L_{reg}(v', v) = L_{smooth-l1}(v'_\theta, v_\theta) - IoU(v'_{\{x,y,w,h\}}, v_{\{x,y,w,h\}}) \tag{6}$$

Where S is the number of anchor boxes when the parameter obj is 1, it means the foreground, and when it is 0, it means the background. v' and v represent the ground-truth box's prediction vector and target vector. p_n is the probability distribution of various types, and t_n is the corresponding target label. SkewIoU is the overlapping area of the predicted and ground-truth boxes. λ is the sum of different weights and is 1. Finally, f(SkewIoU) and L_{reg} are combined as the regression gradient function.

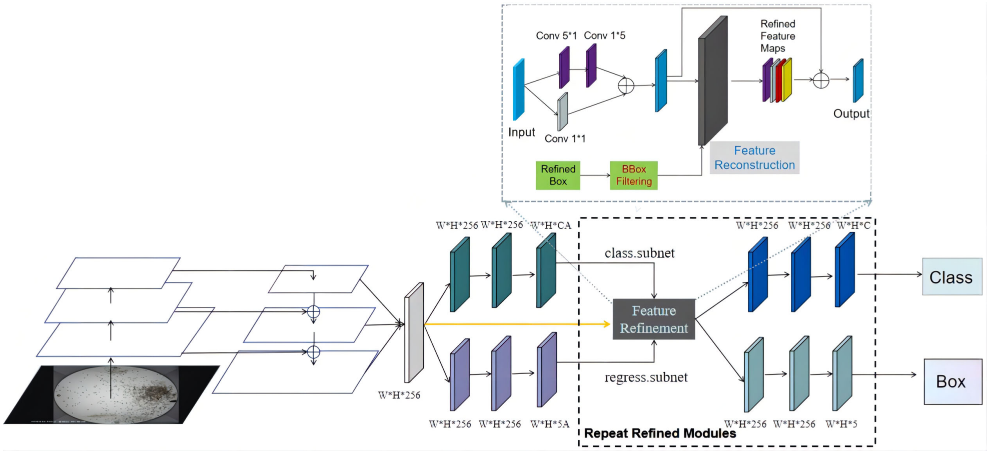


FIGURE 5
The network structure diagram of the rotation detection algorithm used in this experiment.

2.3 Evaluation indicators

The evaluation criteria used in this experiment are single-class Average Precision (AP), single-class Recall, all-class average precision mAP, all-class average recall rate mean Average Recall (mR), model parameters, and detection time comparisons analysis. The relevant calculation formula is shown in the following (7-10).

$$P = \frac{TP}{TP + FP} \times 100\,\% \tag{7}$$

$$R = \frac{TP}{TP + FN} \times 100\,\% \tag{8}$$

$$AP = \int_0^1 P(R)dR \tag{9}$$

$$mAP = \frac{1}{M}\sum_{k=1}^{M} AP(k) \times 100\,\% \tag{10}$$

Where TP and FN are the numbers of positive and negative samples predicted to be positive, FP is the number of negative samples predicted to be positive, and M is the total number of classes in the data. P is precision, R recalls, and AP is precision for a single class.

3 Experimental

3.1 Experimental environment

The operating platform of this experiment is the Ubuntu20.04.4 system. The CPU is Intel Core i9-9900K, the frequency is 3.6GHz, and the running memory is 16G. The graphics card is NVIDIA TITAN RTX, and the GPU memory is 24G. The CUDA version is 10.2, and the CUDNN accelerated version is 7.6.5. PyCharm Professional Edition, Python 3.7.11 interpreter, MMCV version 1.4.0, and Pytorch 1.10 deep learning framework are used.

3.2 Experimental procedure

In the experiment, under the same training set, the number of iterations epoch is 36, the batch size is 4, the learning rate is 0.01, and the value is dynamically optimized during the training process. Momentum is 0.9, and weight decay is set to 0.0005. SGD is a parameter optimizer to train and validate different classification test datasets.

3.2.1 Comprehensive comparison between rotation detection and horizontal detection algorithms

In this experiment, the most representative horizontal detection algorithms and rotation detection algorithms are selected as comparisons. Some of them have the same backbone network structure and are adjusted to Resnet101, and the input image size is scaled to (1800, 1200) during training. During the test, experimental verification was carried out in 5 different scenarios, and the experimental results are shown in Table 2.

It can be seen from the experimental results that the YOLO series algorithm is better than other detection algorithms in mAP. The best level detection algorithm is the YOLOv5 model, which is 6.4%, 7.7%, 3%, and 13.9% higher than Faster RCNN, Cascade RCNN, YOLOX, and RetinaNet at mAP0.5. Regarding recall rate, YOLOv5 and YOLOv6 in the YOLO series are far lower than other detection algorithms, only YOLOX can reach more than 82%, and the algorithm with the highest recall rate for horizontal detection is RetinaNet, which reaches 87.3%. The experiments show that both the one-stage and two-stage target detection algorithms have advantages and disadvantages. Compared with the rotation detection algorithm, the best one-stage algorithm is far lower than the RoFaster RCNN, R3Det, and S2ANet algorithms. RoFaster RCNN is 5.7% and 3.5% higher than Faster RCNN in mAP and Recall under the same conditions. On the same Backbone, R3Det is 24.9%, 26.2%, and 32.4% higher than Faster, Cascade, and RetinaNet algorithms.

3.2.2 Influence of backbone network and image input size

As seen above, rotation detection has initially demonstrated its advantages. In practice, many factors affect the final result of different algorithms. For example, the backbone network and the input image size play a crucial role in the feature extraction of the target object. This paper conducts comparative research experiments on these two effects in different scenarios. The same backbone network is still set to Resnet101, the YOLOv5 and YOLOv6 use CSPDarknet and EfficientRep as the backbone network, respectively, and the image input size during training and testing is adjusted to (1000, 600). The experimental results are shown in Table 3.

Through the comparison of experimental results, it is found that each algorithm has a certain degree of reduction when the input size is reduced. When the size is reduced, YOLOv5 and YOLOv6 mAP drop by 4.2% and 1%, respectively, under Test240. Other horizontal detection Faster RCNN and Cascade RCNN algorithms reduce mAP by 4.4% and 3.6% and Recall by 9.1% and 13.4%, respectively. The rotation detection algorithm declines further; the minor reduction is 2.1% of RoFaster RCNN, and the most significant drop is 7.9% of R3Det. Experimental results show that the image size change substantially impacts the final result. Except for the ReDet algorithm, other rotation detection algorithms are still better than the horizontal detection algorithm model. To verify the influence of the backbone network of the algorithm, continue to join the experiment. Keep the training image input size as (1800,1200) while setting the backbone adjustment depth to Resnet50. The experimental results are shown in Table 4 below.

TABLE 2 Comprehensive model comparison results.

Algorithm model	Backbone	Test240%		SNO180%		SO79%		INO104%		IO80%	
	Indicators	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**
two-stage											
Faster RCNN	Resnet101	53.6	85.3	56.1	84.1	62.0	82.5	56.8	84.3	44.7	68.2
Cascade RCNN	Resnet101	52.3	83.2	54.9	83.4	58.2	77.2	53.6	77.6	45.9	70.2
one-stage											
RetinaNet	Resnet101	46.1	87.3	50.6	94.3	48.0	80.3	47.6	88.1	34.0	72.3
YOLOX	CSPDarknet	57.0	82.0	61.3	82.6	65.5	78.2	53.4	72.2	47.8	75.6
YOLOv5	CSPDarknet	60.0	63.0	62.5	57.3	65.9	67.4	62.2	61.8	53.4	57.0
YOLOv6	EfficientRep	58.2	54.7	60.5	54.3	66.4	51.4	64.6	53.0	50.8	52.3
rotation detection											
RoFaster RCNN	Resnet101	59.3	88.8	58.6	81.2	69.1	81.4	65.3	86.8	53.6	82.0
ReDet	Resnet101	54.4	87.9	54.6	87.2	60.3	77.7	51.6	83.4	43.4	73.5
S2ANet	Resnet101	60.2	**94.7**	60.0	95.7	69.0	**92.2**	63.8	90.6	54.2	**93.4**
R3Det	Resnet101	**78.5**	93.6	**85.1**	**99.1**	**82.6**	89.7	**79.0**	**91.9**	**70.3**	85.0

The bold numbers in the table indicate the highest values of he experimental results.

We can be seen from the results that when the image training size is (1800, 1200) and the backbone network depth is reduced to Resnet50, the horizontal detection and rotation detection algorithms have a slight reduction. Among them, the algorithm with the most negligible reduction is 0.8% of R3Det, and the highest is only 1.7%. The highest reduction of the horizontal detection algorithm above the recall rate is 7.6% of Cascade RCNN, and the rotation detection algorithm has almost no change. However, experiments show that when the data size is large, the network training model has less influence on the depth of feature extraction.

3.2.3 Analysis of recall and mAP of different algorithms in different types of datasets

This experiment selects four algorithms with the best detection effect for comparison. The horizontal one-stage detection algorithm is YOLOX, the second-stage detection algorithm is Faster RCNN, and the rotation detection algorithm is R3Det and S2ANet. Take Test240 data as the test set for the model. The comparison of mAP and mean Average Recall(mRecall) is shown in Figure 6.

Figure 6 shows that at mAP, S2ANet is higher than other level detection algorithms for various pests under different environmental conditions, and the mAP is only lower than 1.3% on SNO. The detection effect of R3Det in different test sets, mAP reached 78.5%, 85.1%, 82.6%, 79%, and 70.3%, respectively; this shows that R3Det is more efficient and flexible in the detection of dense target pests through the refinement module and the feature reconstruction module.

In the mRecall comparison, although the mAP of YOLOX is higher than that of Faster RCNN, the recall rate is lower than that of Faster RCNN. The two algorithmic models of rotation detection outperformed the horizontal detection algorithm. Rotation detection achieves the highest Recall of more than 95% on the SNO simple data set. The Recall calculated by R3Det is above 86% on all types of data sets, which shows that the horizontal anchor frame and the rotation frame used by R3Det are combined to improve the recall rate. At the same time, the

TABLE 3 Comparison of detection results when the input image is 1000*600.

Algorithm model	Backbone	Test240%		SNO180%		SO79%		INO104%		IO80%	
	Indicators	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**
Faster RCNN	Resnet101	49.2	76.2	50.7	76.8	55.9	72.1	52.1	78.2	42.5	74.3
Cascade RCNN	Resnet101	48.7	69.8	51.0	74.1	54.8	70.0	49.3	65.2	41.2	67.3
YOLOv5	CSPDarknet	55.8	59.7	62.3	61.1	65.4	67.4	59.4	53.5	50.1	57.0
YOLOv6	EfficientRep	57.2	51.9	60.4	58.7	64.8	54.3	58.1	54.1	47.6	54.3
RoFaster RCNN	Resnet101	57.2	83.0	55.0	87.1	67.7	82.2	63.9	84.7	49.9	80.8
ReDet	Resnet101	44.8	76.0	51.3	85.3	53.6	74.6	46.1	74.4	37.5	66.0
S2ANet	Resnet101	56.5	**93.6**	57.6	95.2	64.9	**89.2**	60.2	90.2	47.4	**90.6**
R3Det	Resnet101	**70.6**	91.9	**78.6**	**97.7**	**74.8**	85.8	**77.9**	**91.4**	**55.1**	76.8

The bold numbers in the table indicate the highest values of the experimental results.

TABLE 4 Comparison of detection results when the input picture is 1800*1200.

Algorithm model	Backbone	Test240%		SNO180%		SO79%		INO104%		IO80%	
	Indicators	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**	**mAP**	**mR**
Faster RCNN	Resnet50	52.5	82.5	53.8	74.5	59.3	77.9	57.7	67.7	45.2	66.1
Cascade RCNN	Resnet50	53.1	75.6	56.2	86.3	58.6	71.4	53.2	84.4	43.3	77.1
RoFaster RCNN	Resnet50	58.3	85.6	59.0	87.2	68.1	87.4	64.1	86.3	52.9	83.2
ReDet	Resnet50	52.7	87.0	55.0	86.3	59.9	78.6	53.3	85.7	41.6	69.7
S2ANet	Resnet50	58.8	**95.6**	59.2	97.2	68.4	**91.6**	63.5	**92.2**	51.7	**94.7**
R3Det	Resnet50	**77.7**	94.0	**76.1**	**98.2**	**72.9**	87.8	**74.0**	91.5	**59.0**	79.1

The bold numbers in the table indicate the highest values of the experimental results.

approximate SkewIoU loss function is used to achieve more accurate rotation. Finally, the results show that the recall rate can be significantly improved, which has good results under austere conditions and overcomes the problem of dense scenes.

In summary, whether a one-stage or two-stage target detection algorithm, the detection effect is not as good as rotation detection in various environments. In contrast, other rotation algorithms, such as S2ANet and RoFaster RCNN, have an excellent recognition ratio. In particular, the R3Det algorithm still performs well in environments with severe occlusion and more complex backgrounds, which shows that the rotation algorithm has good results in remote sensing data and a reasonable recognition rate in pest detection in different fields in the field and generalization rate.

3.2.4 Analysis of a single type of pest

The total categories of the data set in this experiment are 21 categories. The growth shape and other characteristics of different pest types have specific differences, and some attributes of some categories are similar. In order to provide a theoretical reference for identifying more varieties of pests in the future, this paper analyzes the influence of characteristics of different pests. Figure 7 shows the aspect ratio and relative size of a single category of pests. The algorithm model is trained with horizontal detection and rotation detection. The single-category AP50 of different algorithms is calculated, and the results are shown in Table 5.

It can be seen from Table 5 that under the same data conditions, the aspect ratio of the rotating frame is larger than the scale of the horizontal structure, and the relative proportion of the rotating frame is lower than that of the horizontal frame. In general, the area occupied by pests is small. It shows that the detection and recognition of tiny pests are complex, and the training samples of the rotating frame can better fit the target object. The interference of other environmental factors on the models during training in different scenarios is also reduced. Therefore, the rotation detection algorithm can still achieve good results under more complex or denser conditions.

The table shows the single-class experimental results for the selected model comparisons. It can be concluded from this table that when the aspect ratio of pests is greater than 2, only one pest is the 151st pest, and the mAP of this pest is 90%. When the ratio is [1.75, 2), the mAP of the 8th class of pests is 89.7%. When the ratio was [1.65, 1.75], including the 87th, 20th, and 115th types

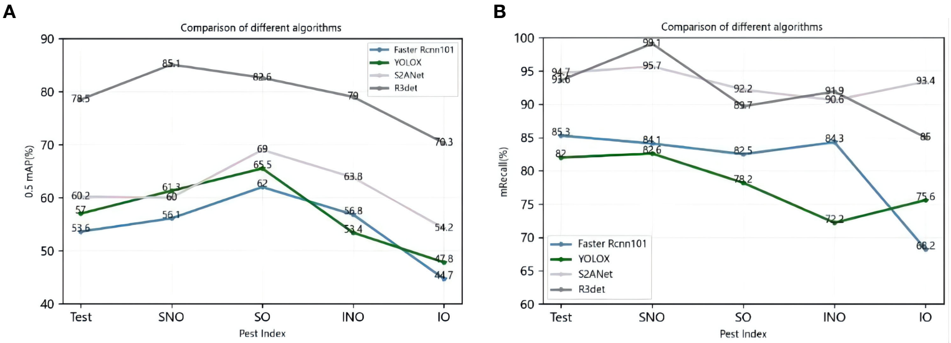


FIGURE 6
Left panel **(A)** shows the mAP of the four algorithms on the Test datasets, and correct panel **(B)** shows the Recall of the corresponding algorithms and datasets.

of pests, the mAP was 79.7%, 89.6%, and 83%, respectively. When the ratio was [1.55, 1.65), there were 6 species of pests; the highest was 86% of class 233, and the lowest was 62.8% of class 291. There are also six classes where the ratio is [1.50, 1.55), where the best detections are 97.9% for class 445 and 96.3% for class 359. When the ratio was lower than 1.5, there were four classes, 2, 248, 3, and 64, with mAP of 76.5%, 81.8%, 55.6%, and 73%, respectively.

After analysis, there were 15 types of detected pests with aspect ratios between [1.5, 1.75], accounting for 71.4% of the total detected pest species. The R3det rotation detection algorithm is generally more effective than other horizontal detection algorithms in detecting these categories. When it is lower than 1.5, the rotation detection still performs well. Experiments show that the rotation algorithm detection not only has a good effect on detecting pests at a high aspect ratio but also has a reasonable recognition rate when the ratio is low. For example, in comparing 21 categories of total pests, R3det is the highest in 19 pests, second only to Cascade RCNN in the 248th category of pests, but still achieves an mAP of 81.8%. The analysis results further demonstrate that the R3det model can detect most pests.

3.2.5 Comparative analysis of detection speed and parameter quantity

Regarding recognition rate, the rotation detection algorithm has shown better results than the horizontal detection. However, timely detection of changes before and after pests and diseases and making correct judgments are the key to agricultural control. Therefore, the detection time is also an important indicator. On the other hand, different detection algorithm models finally need to be transplanted to specific hardware devices for mobile deployment. However, due to the limited resources of various hardware devices, they cannot carry large capacities; Therefore, the model's size is also one of the essential considerations when choosing a suitable algorithm. Finally, as shown in Table 6, we

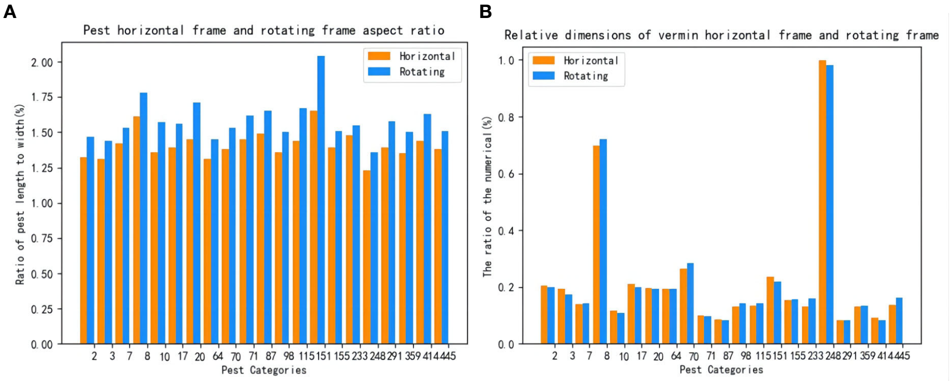


FIGURE 7
Left panel **(A)** shows the aspect ratio data of A single class of pests, and right panel **(B)** shows the proportion of the relative original size of A single type of pest.

TABLE 5 AP50 for a single class.

Method	2	3	7	8	10	17	20	64	70	71	87	98	115	151	155	233	248	291	359	414	445
Faster RCNN	32.4	37.3	32.9	74.3	28.8	59.4	61.1	54.8	81.6	60.8	54.7	60.1	47.4	68.0	58.7	26.4	88.1	17.7	53.3	42.4	85.3
Cascade RCNN	30.0	34.0	15.3	77.0	24.8	44.8	61.8	63.1	77.7	61.4	50.2	44.9	40.9	68.1	53.3	55.5	**92.7**	18.5	68.6	41.1	75.5
RetinaNet	28.2	29.1	6.6	73.4	29.6	48.2	47.8	53.2	76.1	58.2	51.7	47.1	30.4	74.0	54.1	41.2	59.5	14.7	58.0	27.1	59.6
YOLOX	47.5	37.9	**56.8**	77.2	40.4	49.0	54.5	56.5	76.0	63.7	26.9	63.0	44.8	59.5	57.2	69.1	70.0	23.6	74.8	60.3	87.4
YOLOv5	32.3	26.5	33.0	71.1	35.4	49.2	52.1	52.6	80.7	64.4	50.0	71.6	35.8	63.6	58.1	71.9	81.9	21.4	72.6	55.5	86.3
RoFaster RCNN	27.9	46.3	36.4	80.2	36.7	64.4	56.9	63.6	78.4	66.3	62.5	62.3	52.5	75.4	66.3	46.8	85.5	25.0	74.3	54.4	82.8
ReDet	27.0	43.5	14.1	80.4	28.7	53.2	52.2	55.2	75.4	60.2	53.7	51.7	42.6	68.3	57.1	46.9	85.7	20.5	62.0	50.5	76.7
S2ANet	28.0	35.2	29.8	85.7	39.6	65.1	58.1	61.4	83.8	71.2	58.2	61.6	57.0	76.5	70.9	53.1	90.7	23.4	70.7	62.6	80.6
R3Det	**76.5**	**55.6**	14.7	**89.7**	**77.8**	**86.0**	**89.6**	**73.0**	**87.1**	**85.7**	**79.7**	**88.3**	**83.0**	**90.0**	**90.4**	**72.7**	81.8	**62.8**	**96.3**	**69.7**	**97.9**

The bold numbers in the table indicate the highest values of the experimental results.

compared the model parameters and detection time of different models under different backbone network depth conditions and when the image input size changes during training.

It can be seen from the experimental results that on the same backbone network, the rotation detection algorithm is slightly lower than the horizontal detection algorithm in the detection speed of a single image. The maximum time of the rotation detection algorithm for a single image is only 0.163s, which can meet the requirements of practical detection applications. Similarly, in terms of the number of algorithm models, the parameters of RoFaster, ReDet, and S2ANet algorithms are all lower than those of the horizontal detection algorithm. The performance of R3Det is slightly higher than that of the horizontal detection algorithm, but the amount of parameters is only 66.54MB. The practice has proved that the algorithm can be flexibly applied to the embedded mobile deployment of pest-monitoring lights.

3.2.6 Pest detection visualization comparison

Through the above comparative studies in different aspects, it is found that rotation detection algorithms such as R3det have better detection results. In this experiment, to verify the detection effect in the actual scene, the Faster RCNN and Cascade RCNN with the best horizontal detection effect were selected, and the rotation detection was compared with R3det and S2ANet as the representative algorithms. The threshold was set to 0.5, and the test data included small targets, dense and occlusion type 3, the detection effect is shown in Figure 8, and Figure 9 compares the decreasing trend of the loss of different algorithms.

It can be seen from the comparison effect that R3det can detect all pests in small target detection. The detection results of Faster RCNN and S2ANet are the same. Meanwhile, Cascade RCNN has the worst detection performance, only detecting a few pests. In dense scenarios, the horizontal detection algorithm can only detect a few pests, which is far from meeting the actual needs. The rotation detection algorithm shows its superior detection ability in a dense environment. And the detection capability is much higher than horizontal detection, and more pests can be detected in this environment. In practical situations, pests are prone to occlusion when they appear in piles. The horizontal detection algorithm is prone to be disturbed by other target features during training and has a seriously missed detection rate. In this case, rotation detection can better fit the pest samples under different postures and accurately identify the blocked pests. Among them, the R3det algorithm can account for both small targets and occluded pests in the case of occlusion.

4 Discussion and conclusion

Detecting agricultural pests has always been a complex problem for many experts and scholars. Insect pests will not only eventually reduce crop yield but also may impact the ecological balance of a specific area. Therefore, accurate identification and detection of pests in complex scenarios is the key to the environmental protection of crops. Traditional reliance on agricultural experts for on-site inspection and testing is inefficient and time-sensitive, often missing the optimal period of protection. In the current research on deep learning object detection, it is found that horizontal detection has a certain effect on the simple background of a single pest. However, the product is difficult to meet the actual requirements in complex multi-category environments. In this paper, the rotation detection algorithm is firstly proposed to be applied to the pest detection field of the constructed pest datasets PRD21, and good detection results have been achieved, which provides a new solution for pest detection in the early stage of agriculture. Among them, the R3Det algorithm uses its refinement module to improve the recognition rate and approximate SkewIoU loss to improve the recall rate. Finally, the detection comparison in the actual environment proves its superiority and strong adaptability. The overall experimental conclusions are as follows:

1) This paper uses rotation and horizontal detection algorithms to research pest detection and identification.

TABLE 6 Comparison of detection speed and parameter amount of the same backbone network algorithm.

Algorithmmodel	Backbone Resnet 50 1800*1200 (1800*1200)			Backbone Resnet 101 1000*600 (1000*600)			Backbone Resnet 101 1800*1200 (1800*1200)		
	FPS	Single graph detection time/s	GFLOPs/ MB	FPS	Single graph detection time/s	GFLOPs/ MB	FPS	Single graph detection time/s	GFLOPs/ MB
Faster RCNN	13.5	**0.074**	41.23	25.3	**0.040**	60.22	10.1	**0.100**	60.22
Cascade RCNN	12.0	0.083	68.99	21.1	0.047	87.98	9.2	0.109	87.98
RoFaster RCNN	12.5	0.08	41.14	22.1	0.045	60.14	9.5	0.105	60.14
ReDet	8.8	0.114	40.23	16.8	0.060	58.22	6.7	0.149	58.22
S2ANet	10.8	0.092	**38.63**	20.8	0.048	**57.62**	8.4	0.119	**57.62**
R3Det	7.2	0.138	47.54	14.0	0.071	66.54	6.1	0.163	66.54

The bold numbers in the table indicate the highest values of the experimental results.

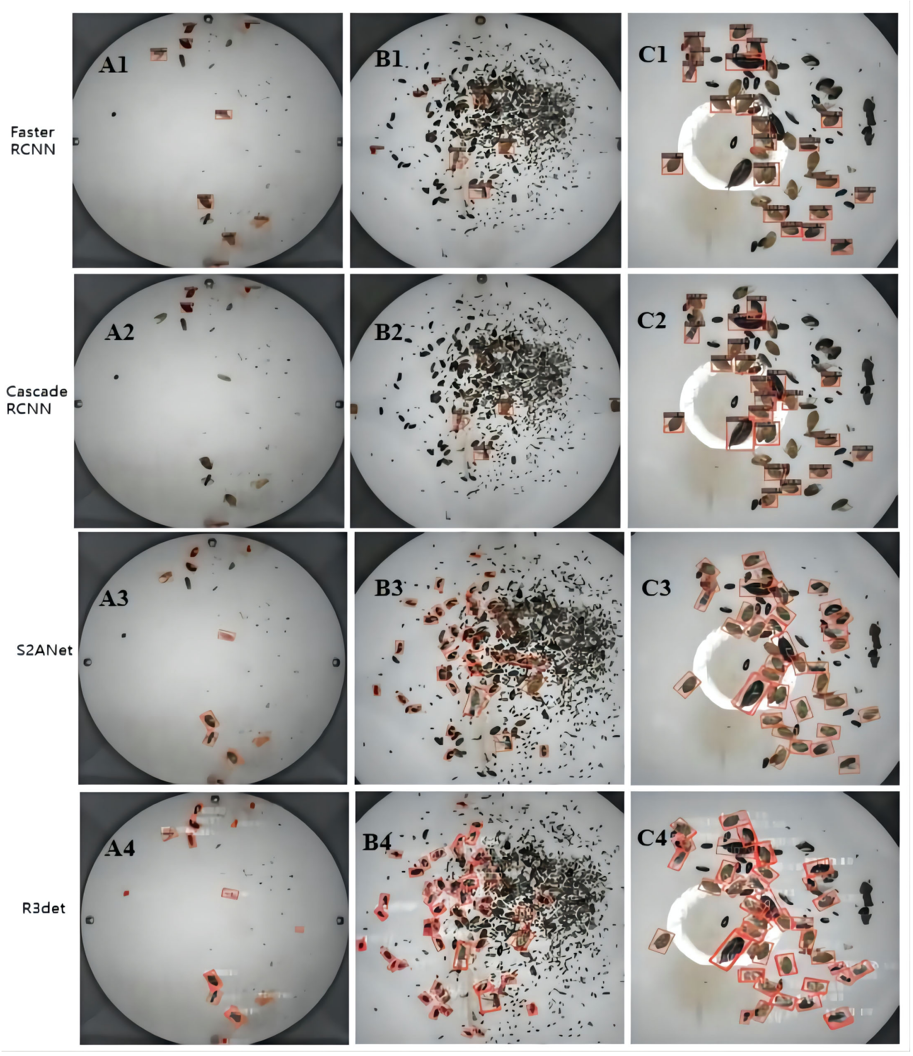


FIGURE 8
Comparison between horizontal algorithm and rotation algorithm. The algorithm model for comparison is Faster RCNN, Cascade RCNN, S2ANet and R3Det. Test figure **(A)** represents small-target pest detection, **(B)** represents intensive pest target detection, and **(C)** represents interpest occlusion type detection.

Under different natural image detection environments, rotation detection reflects the advantages of good generalization and strong adaptability. The R3det algorithm can still achieve a recognition rate of more than 70% under more occlusion and serious background interference, and the Recall also reaches 86.0%. It achieves 85.1%, 82.6 and 79% under the other classification test data sets, SNO, SO, and INO.

2) In single-class detection, the performance of rotation detection is the highest in 19 of the 21 categories. The highest category is the 445th category, which reaches 99.7%, and the other category achieves 81.1%. The detection effect shows that the rotation algorithm has good robustness to multi-category targets in addition to the influence of environmental factors.

3)Since pests may increase over time over large areas, it is necessary to detect and identify pests in the exact location within a short period. Through experiments, it has been found that the detection time of a single image of the rotation detection algorithm is less than 0.17s, which can realize rapid identification and detection.

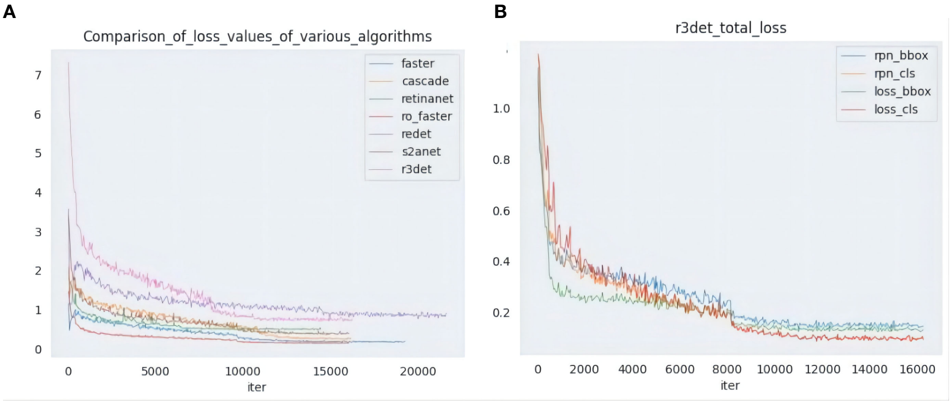


FIGURE 9
Left panel **(A)** shows the Loss comparison of multiple algorithms, and right panel **(B)** shows multiple Loss curves of the R3Det algorithm.

The above experiments prove that rotation detection has practical application value on pests. However, at the same time, there are some deficiencies. For example, the detection effect of category 7 pests is low, and there is still room for improvement when the environment is the most complex. In the future, we will further collect samples of various pests in different environments and add specific pest categories to expand the training sample database of pests in other regions. In addition, the algorithm is optimized, improved, and innovated. Ultimately, it provides a new research method for intelligent plant protection and detecting crop diseases and insect pests.

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

WZ designed and carried out the experimental design, selected a variety of level detection and rotation detection algorithm models for comparative analysis and research, and wrote the manuscript of the paper. The XX screening data set is annotated with horizontal and rotational labels and article grammar and image modifications. JD, XM, and ZZ proposed the overall framework design for this paper and conducted experimental research to guide it. TC participated in the experimental design and provided constructive comments. JD are the project directors. GZ provides raw pest data samples. All the authors contributed to this article and approved the submitted version.

Funding

This work was supported in part by the project of the Dean’s Fund of Hefei Institute of Physical Science, Chinese Academy of Sciences (YZJJ2022QN32).

Acknowledgments

This paper thanks to the support of software and hardware experimental environment provided by intelligent agriculture valley. We would also like to thank all the authors and commentators cited in this article for their constructive comments and suggestions.

Conflict of interest

Author GZ is employed by Henan Yunfei Technology Development 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

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

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv* 2004, 10934. doi: 10.48550/arXiv.2004.10934

Cai, Z., and Vasconcelos, N. (2018). "Cascade r-cnn: Delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6154–6162. doi: 10.1109/CVPR.2018.00644

CCTV News *Pests and diseases cause 40 million tons of grain loss each year in China*. Available at: http://news.cctv.com/2017/02/11/ARTIPy3JBOkTuecw4r9lP7JF170211.shtml (Accessed 2 November 2017).

Cho, J., Choi, J., Qiao, M., Ji, C. W., and Chon, T. S. (2007). Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis. *International Journal of Mathematics and Computers in Simulation* 1 (1), 46–53. doi: 10.1016/j.ecoinf.2014.09.006

Dayan, M. P. (1988). Survey, identification and pathogenicity of pests and diseases of bamboo in the Philippines. *Sylvatrop* 13, 61–77.

Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., and Jamshidi, B. (2017). "Vision-based pest detection based on SVM classification method.". *Comput. Electron. Agric.* 137, 52–585. doi: 10.1016/j.compag.2017.03.016

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vis.* 88, 303–338. doi: 10.1007/s11263-009-0275-4

Fu, K., Li, Y., Sun, H., Yang, X., Xu, G., Li, Y., et al. (2018). A ship rotation detection model in remote sensing images based on feature fusion pyramid network and deep reinforcement learning. *Remote Sensing.* 10 (12), 1922. doi: 10.3390/rs10121922

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv* 2107, 08430. doi: 10.48550/arXiv.2107.08430

Gevorgyan, Z. (2022). SIoU loss: More powerful learning for bounding box regression. *arXiv* 2205, 12740. doi: 10.48550/arXiv.2205.12740

Girshick, R. (2015). "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. doi: 10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 580–587. doi: 10.1109/CVPR.2014.81

Guru-Pirasanna-Pandi, G., Adak, T., Gowda, B., Patil, N., Annamalai, M., and Jena, M. (2018). Toxicological effect of underutilized plant, cleistanthus collinus leaf extracts against two major stored grain pests, the rice weevil, sitophilus oryzae and red flour beetle, tribolium castaneum.Ecotoxicol. *Environ. Safe.* 154, 92–99. doi: 10.1016/j.ecoenv.2018.02.024

Han, J., Ding, J., Xue, N., and Xia, G. S. (2022). "Align deep features for oriented object detection," in *IEEE transactions on geoscience and remote sensing*. vol 60, 1–11. doi: 10.1109/TGRS.2021.3062048

Han, J., Ding, J., Xue, N., and Xia, G.-S. (2021). "ReDet: A Rotation-equivariant Detector for Aerial Object Detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2785–94. doi: 10.1109/CVPR46437.2021.00281

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. doi: 10.1109/ICCV.2017.322

Liao, M., Zhu, Z., Shi, B., Xia, G. S., and Bai, X. (2018). "Rotation-sensitive regression for oriented scene text detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 5909–5918. doi: 10.1109/CVPR.2018.00619

Li, K., Cheng, G., Bu, S., and You, X. (2018). "Rotation-insensitive and context-augmented object detection in remote sensing images," in *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 56. 2337–2348. doi: 10.1109/TGRS.2017.2778300

Li, W., Wang, D., Li, M., Gao, Y., Wu, J., Yang, X., et al. (2021). "Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse". *Comput. Electron. Agric.* 183, 106048. doi: 10.1016/j.compag.2021.106048

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)*, 936–44. doi: 10.1109/CVPR.2017.106

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal Loss for Dense Object Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol 42 (2), 318–27. doi: 10.1109/TPAMI.2018.2858826

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). "Ssd: Single shot multibox detector," In: B. Leibe, J. Matas, N. Sebe and M. Welling *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science* (Cham: Springer), 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudrman, S., et al. (2019). "PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification," in *IEEE Access*. vol 7, 45301–45312. doi: 10.1109/ACCESS.2019.2909522

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*., 8759–8768. doi: 10.1109/CVPR.2018.00913

Ma, Y., Qu, X., Feng, D., Zhang, P., Huang, H., Zhang, Z., et al. (2022). "Recognition and statistical analysis of coastal marine aquacultural cages based on R3Det single-stage detector: A case study of fujian province, china.". *Ocean Coast. Manage.* 225, 106244. doi: 10.1016/j.ocecoaman.2022.106244

Neubeck, A., and Van Gool, L. (2006). "Efficient non-maximum suppression," in *18th international conference on pattern recognition (ICPR'06)*, vol. 3. (IEEE), 855. doi: 10.1109/ICPR.2006.479

Peng, Y., Lu, X., Quan, W., Zhou, N., Zou, D., and Chen, J. X. (2021). "Adversarial reconstruction for outdoors insulator anomaly detection and recognition in high-speed railway traction substation," in *2021 6th international conference on intelligent computing and signal processing (ICSP)* (IEEE), 1349–1354. doi: 10.1109/ICSP51882.2021.9408830

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*., pp. 779–788. doi: 10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2017). "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6517–6525. doi: 10.1109/CVPR.2017.690

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. (2017). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol 39 (6), 1137–49. doi: doi: 10.1109/TPAMI.2016.2577031

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. (2019). "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*., 658–666. doi: 10.1109/CVPR.2019.00075

Solis-Sánchez, L. O., Castañeda-Miranda, R., García-Escalante, J. J., Torres-Pacheco, I., Guevara-González, R. G., Castañeda-Miranda, C. L., et al. (2011). Scale invariant feature approach for insect monitoring. *Comput. Electron. Agric.* 75 (1), 92–99. doi: 10.1016/j.compag.2010.10.001

Tang, Z., Chen, Z., Qi, F., Zhang, L., and Chen, S. (2017). "Pest-YOLO: Deep Image Mining and Multi-Feature Fusion for Real-Time Agriculture Pest Detection," in *2021 IEEE International Conference on Data Mining (ICDM)*,., 1348–53. doi: doi: 10.1109/ICDM51629.2021.00169

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, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., and Yeh, I. H. (2020). "CSPNet: A new backbone that can enhance learning capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*., 1571–80. doi: 10.1109/CVPRW50498.2020.00203

Wang, G. A., Zhang, T., Cheng, J., Liu, S., Yang, Y., and Hou, Z. (2019). "RGB-Infrared cross-modality person re-identification *via* joint pixel and feature alignment," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*., 3623–3632. doi: 10.1109/ICCV.2019.00372

Xie, X., Cheng, G., Wang, J., Yao, X., and Han, J. (2021). "Oriented R-CNN for Object Detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*., 3500–3509. doi: 10.1109/ICCV48922.2021.00350

Yang, X., Yan, J., Feng, Z., and He, T. (2021). "R3det: Refined single-stage detector with feature refinement for rotating object," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35. doi: 10.1609/aaai.v35i4.16426

Yang, X., Sun, H., Sun, X., Yan, M., Guo, Z., and Fu, K. (2018). "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network,". *IEEE Access* 6, 50839–50849. doi: 10.1109/ACCESS.2018.2869884

Zhang, W., Xia, X., Du, J., Zhang, Z., and Zhang, H. (2022). "Recognition and detection of wolfberry in the natural background based on improved YOLOv5 network," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*. 256–262. doi: 10.1109/CVIDLICCEA56201.2022.9824287

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 26 January 2023
DOI 10.3389/fpls.2022.1033544

Prior knowledge auxiliary for few-shot pest detection in the wild

Xiaodong Wang[1,2], Jianming Du[1*], Chengjun Xie[1], Shilian Wu[3], Xiao Ma[4], Kang Liu[5*], Shifeng Dong[1,2] and Tianjiao Chen[1,2]

[1]Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, China, [2]Science Island Branch, Graduate School of University of Science and Technology of China, Hefei, China, [3]Department of Automation, University of Science and Technology of China, Hefei, China, [4]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, [5]Department of Computer Science, University of Sheffield, Sheffield, United Kingdom

Check for updates

OPEN ACCESS

EDITED BY
Peng Chen,
Anhui University, China

REVIEWED BY
Nisha Pillai,
Mississippi State University, United States
Sashuang Sun,
Zhejiang University, China

*CORRESPONDENCE
Jianming Du
djming@iim.ac.cn
Kang Liu
xinxilk@mail.ustc.edu.cn

SPECIALTY SECTION
This article was submitted to Sustainable and Intelligent Phytoprotection, a section of the journal Frontiers in Plant Science

RECEIVED 31 August 2022
ACCEPTED 15 December 2022
PUBLISHED 26 January 2023

CITATION
Wang X, Du J, Xie C, Wu S, Ma X, Liu K, Dong S and Chen T (2023) Prior knowledge auxiliary for few-shot pest detection in the wild.
Front. Plant Sci. 13:1033544.
doi: 10.3389/fpls.2022.1033544

One of the main techniques in smart plant protection is pest detection using deep learning technology, which is convenient, cost-effective, and responsive. However, existing deep-learning-based methods can detect only over a dozen common types of bulk agricultural pests in structured environments. Also, such methods generally require large-scale well-labeled pest data sets for their base-class training and novel-class fine-tuning, and these significantly hinder the further promotion of deep convolutional neural network approaches in pest detection for economic crops, forestry, and emergent invasive pests. In this paper, a few-shot pest detection network is introduced to detect rarely collected pest species in natural scenarios. Firstly, a prior-knowledge auxiliary architecture for few-shot pest detection in the wild is presented. Secondly, a hierarchical few-shot pest detection data set has been built in the wild in China over the past few years. Thirdly, a pest ontology relation module is proposed to combine insect taxonomy and inter-image similarity information. Several experiments are presented according to a standard few-shot detection protocol, and the presented model achieves comparable performance to several representative few-shot detection algorithms in terms of both mean average precision (mAP) and mean average recall (mAR). The results show the promising effectiveness of the proposed few-shot detection architecture.

KEYWORDS

few-shot detection, hierarchical structure, pest recognition, prior knowledge, cross-relation

1 Introduction

Food issues have long concerned countries around the globe, as they do the Chinese government at all levels. In particular, preventing crop diseases and insect pests is not only crucial for increasing food production but also effective for reducing latent agricultural economic losses and facilitating accurate predictions of future grain yields. Current methods for preventing crop diseases and insect pests are still heavily reliant on manual observations by experienced farmers, and they suffer from a long-term shortage of professional agricultural technicians (He et al., 2012; Parsa et al., 2014). Faced with hundreds of millions of Chinese farming households, having only approximately 550,000 Chinese national agricultural technology extension agencies are far from sufficient (Zhang et al., 2016). Furthermore, (i) a large age gap among agricultural technicians, (ii) a lack of pest-recognition staff in each county-level plant protection station, and (iii) differing field experiences are causing a low cover density of experts specializing in pest identification and a lack of unified pest-identification criteria, thereby leading to the blind application of pesticides and serious environmental pollution (Yu, 2021).

Automatic pest identification originated from combining insect morphology with traditional machine-learning algorithms (Watson et al., 2004; Murakami et al., 2005). However, despite most researchers still placing heavy emphasis on machine-learning-based pest classification, automated pest detection based on deep learning has grown rapidly in recent years. Many researchers have used portable probes with digital cameras (Wang et al., 2021) and stationary light traps (Liu et al., 2020; Dong et al., 2021; Du et al., 2022) to automatically identify over a dozen types of tiny pests by means of artificial intelligence. Pest detection offers more semantic information with which to carry out real-world farming tasks, such as object-detection-based swarm counting (Li et al., 2022) and similar pest detection (Wang et al., 2022), whereas pest classification fails to recognize and locate multiple unknown categories of pests in a single image simultaneously. Therefore, pest detection is much more practical for precise pesticide application and pest control, and it helps agricultural plant protection experts deliver accurate treatments to control and avoid the occurrence of larger-scale pest outbreaks as early as possible.

However, current deep-learning-based methods require sufficient data to build a structural minimization model and to support cross-domain model adaption, while machine-learning-based methods demand complex hand-crafted feature descriptors and controlled laboratory backgrounds (Ngugi et al., 2021). To the best of our knowledge, little attention has been paid to those rarely collected but still harmful insect pest species whose samples are difficult to collect because of geography, season, frequency, and pest mobility (Wang, 2021). Moreover, it is difficult for even many images taken continuously from a single camera angle to fully reflect the semantic information of insects because images that are helpful for distinguishing pest species are often only a few representative images taken from multiple angles, such as of the fronts, sides, backs, and abdomens of pests (Huo and Tan, 2020). Therefore, it would be meaningful to discover a novel class with only a few instances (i.e., 10, 15, or 20 shots) (Wang et al., 2020a; Parnami and Lee, 2022). Until high-performance few-shot conceptual models that can be trained quickly become available, customization to collect big data for different scenarios is a reality that the artificial-intelligence community must face (Zhang et al., 2022). To solve this problem, we may have to start from scratch with data structure, logic causality, various invariants of vision, and compositional concept learning, among other topics, and introduce prior knowledge to auxiliary model training.

On the other hand, introducing few-shot learning technology would make it possible to detect rarely collected pest categories with just a few available samples, which would greatly reduce the cost of manual labeling through a semi-supervised automatic labeling process in which only a small amount of manual verification and calibration would have to be done by agricultural technical experts in the later stages. In addition, it would contribute to establishing a rapid response mechanism for invasive alien pests.

The contributions of this paper are summarized as follows:

1. We introduce a prior-knowledge auxiliary architecture for few-shot pest detection in the wild, which allows us to detect rarely collected pests with extremely few available samples.
2. Based on insect taxonomy, we built a new hierarchical FSIP52 data set for few-shot pest detection in natural scenarios. It could be a valuable supplementary data set for the Intellectual Plant Protection and Pest Control Community when combined with the IP102 data set (Wu et al., 2019).
3. We introduce a pest ontology relation module that is composed of a multi-relation detector and a correlation softmax loss function to incorporate prior knowledge for feature discrimination and representation. These blocks allow us to implement multi-task joint training on our model explicitly and implicitly.

2 Related work

2.1 Pest recognition

For more than a decade now, many researchers have developed various machine-learning-based pest identification methods. Larios et al. (2008) proposed a method for identifying stonefly larvae based on the scale-invariant feature transform, and it achieved a classification accuracy of 82% on four types of stonefly larvae. Wen and Guyer (2012) developed an image-based method for the automated identification and classification of orchard insects using a model that combined global and local features, and it achieved a classification performance of 86.6% on eight species of orchard field insects. Kandalkar et al. (2014) designed a pest identification procedure based on saliency map segmentation and discrete-wavelet-transform feature extraction, and utilized it in classifying pest categories using shallow back-propagation neural networks. These types of algorithms use close-up images of pest specimens in a restricted background to recognize common insects and pests, but they also require a high degree of expertise in hand-crafted feature design and parameter selection for empirical formulas. Currently, deep-learning algorithms based on large-scale data have replaced traditional pest-identification algorithms. By combining low-level and high-level contextual information of images, they have made amazing progress in identifying the pain points of detecting tiny pests and have realized the value of implementing and applying modern pest-identification algorithms. Liu et al. (2020) implemented an approach for large-scale multi-class pest detection in a stationary light trap, which could detect 16 classes with a deep-learning-based automatic multi-class crop-pest monitoring approach using hybrid global and local activated features. Wang et al. (2021) used 76,595 annotations containing ambient temperature, shooting time, and latitude and longitude information to detect *Petrobia latens*, *Mythimna separata*, and *Nilaparvata lugens* (Stål) with a smart phone in a complex field scene. However, existing deep-learning pest-recognition methods are focused mainly on identifying over a dozen of the most common pest species, for which large-scale samples of each species are required, thereby failing to meet the need for rarely collected pests. Meanwhile, pest images in most research (Li and Yang, 2020; Li and Yang, 2021) have been taken in a structured environment, such as a stationary light trap, instead of in sophisticated wild settings that are more suitable for practical applications. Therefore, being able to identify and detect novel pest classes using fewer data would make it possible to help agricultural technicians and amateur entomologists by providing them with a one-on-one expert insect encyclopedia-style service.

2.2 Few-shot learning

In the real world, conventional deep neural networks have always suffered from sample scarcity and the high cost of acquiring labeled data. This challenge indirectly gave rise to few-shot learning, which is generally regarded as the method of training a model to achieve good generalization performance in the target task based on very few training samples. In the fine-tuning stage, there are new classes that have never been seen before, and only a few labeled samples of each class are available; then in the testing process, when faced with new categories, the task can be completed without changing the existing model. Few-shot learning is divided into transductive learning and inductive learning, and all the models discussed herein correspond to inductive learning, in which there are three main methods, namely, meta-learning, metric learning, and transfer learning. Most few-shot classification and detection methods are based on fine-tuning (Fan et al., 2020; Kang et al., 2019; Sun et al., 2021; Wang et al., 2020b; Xiao and Marlet, 2020), and many experiments have shown that fine-tuning offers substantially improved prediction accuracy (Chen et al., 2019; Dhillon et al., 2019; Chen et al., 2020). Dhillon et al. (2019) found that a five-way one-shot fine-tuning increased accuracy by 2%–7%, while a five-way five-shot fine-tuning also increased accuracy by 1.5%–4%. Analogous conclusions have also been drawn in another work (Zhuang et al., 2020). This method is simple but useful, and its accuracy is comparable to that of other sophisticated state-of-the-art (SOTA) meta-learning methods (Li and Yang, 2020). In methods based on fine-tuning, images in the query and support set are mapped to the feature vectors, then the similarities between the query and support images in the feature space are calculated, and the final recognition result is determined by the highest similarity; thus, the model is fine-tuned efficiently even with a limited sample.

2.3 Few-shot pest detection

Research on identifying insect pests and crop diseases based on few-shot learning began in 2019. Li and Yang (2020) implemented metric learning in the few-shot detection of cotton pests and conducted a terminal realization with a field-programmable gate array (FPGA). Li and Yang (2021) provided the Intellectual Plant Protection and Pest Control Community with a task-driven paradigm for meta-learning in agriculture, but it only includes 10 types of close-up pests and plants in low resolution with few-shot classification configuration, which is far from real-world conditions. Yang et al. (2021) used salient-region detection and center neighbor loss to detect insects in complex real-world settings, but the approach focused on only visual features within images and did not introduce prior information to aid detection even with few samples. Yang et al. (2021) also used the iNaturalist open-source data set provided by Google, but this still includes many images whose backgrounds are not natural, generally simple backgrounds such as desktops, cement floors, and specimen trays. Moreover, their samples were collected mainly against strictly controlled laboratory backgrounds or simple natural backgrounds, and they lacked visual external features. In summary, this field is still in its infancy, aiming to identify more novel pests at low data cost.

3 Data preparation

Insecta is the largest class in the animal kingdom, whose number of known species exceeds 850,000, accounting for four fifths of all animals. Within Insecta, nine orders are closely related to agricultural production: Orthoptera, Thysanoptera, Homoptera, Hemiptera, Neuroptera, Lepidoptera, Coleoptera, Hymenoptera, and Diptera. In this paper, all pest species are represented by adults.

Conventionally, almost all image-classification and object-detection tasks are pretrained on the data sets provided by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) or the Microsoft Common Objects in Context (COCO) Detection Data set in order to obtain basic object features and increase the models' generalization ability. Although these prestigious sponsors try their best, their baseline data sets still contain very few images of pests or insects. In this case, we substituted the ImageNet pre-trained data set with the IP102 and few-shot object detection (FSOD) data sets (Fan et al., 2020). IP102 (Wu et al., 2019) is an insect baseline data set that contains 18,974 images with 22,253 annotations for object detection, making it a fairly good replacement for COCO and ImageNet (Krizhevsky et al., 2017).

However, IP102 is collected by web crawlers through common Internet image search engines such as Google, Flickr, and Bing, so it consistently suffers from poor resolution, rough annotation, improper size, and copyright watermarks. As a supplement, our FSIP52 data set contains 1,918 high-quality images that were carefully annotated and manually reviewed by pest-identification experts at the Anhui Academy of Agricultural Sciences and the Yun Fei Company, aiming to improve the signal-to-noise ratio of characteristic information in real-world pest samples with high consistency. It comprises 52 rarely collected adult agricultural and forest fruit-tree pest species with different natural backgrounds in the wild, with only dozens of samples for each pest category on average. Figure 1 gives an intuitive visual demonstration of each category in the FSIP52 data set. The pests in each vignette are in different complex natural settings and vary in size and pose, indicating that the FSIP52 data set is very challenging. After removing the categories of IP102 that overlapped with our FSIP52, we integrated the remaining categories of IP102 as our pre-trained data set. Thus, we are able to fine-tune our model with the FSIP52 split to detect the minority pests fairly.

Few-shot object detection is quite different from general object detection methods. Few-shot learning is the process of method of training a model to achieve good generalization performance in the target task based on very few training samples. Cross-domain

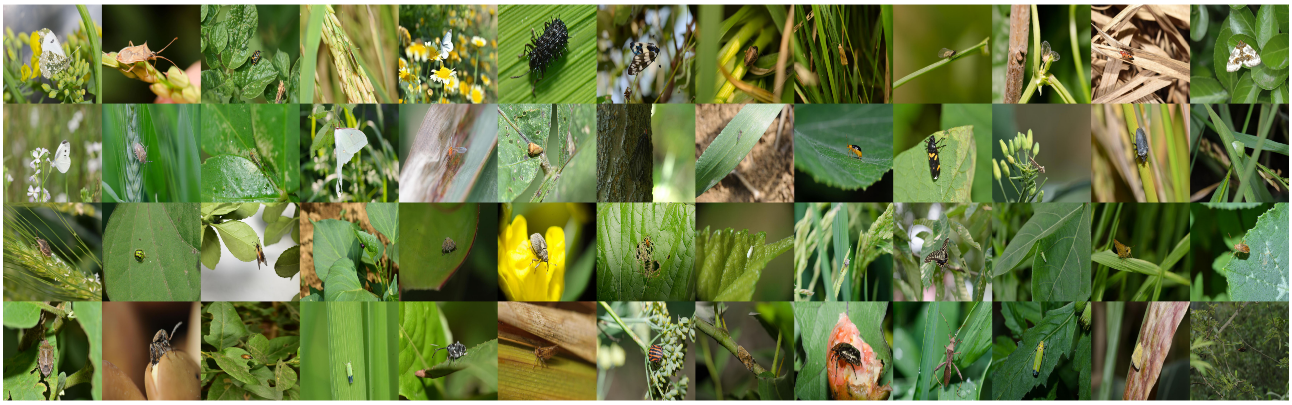


FIGURE 1
Representative demonstration images of each category in the FSIP52 data set.

problems are inevitable, but they can be alleviated by constructing a source data set that is as similar to the target domain as possible. As noted by Sbai et al. (2020), base data set design is crucial for few-shot detection, and typically, it is always more important than the small improvements brought by a complex learning algorithm. Therefore, we carefully designed the base data set size and similarity to test classes and trade off between the numbers of classes and images per class. Furthermore, the degradation of plant–pest cross-domain few-shot classification performance shows the necessity of a scientifically designed data set.

Because pest-victimized crops have complex and changing backgrounds and each pest may harm various crops, it is difficult to encode crop information as effective auxiliary information to guide the model learning. On the other hand, because insect taxonomy reveals inherent connections and provides the respective characteristics of texture and shape of various insect pests, we designed the hierarchical FSIP52 data set based on prior human knowledge and proposed a corresponding hierarchical classifier in our model. FSIP52 is divided explicitly into four super classes and further divided into 52 subclasses. The numbers in brackets after the name of each class of insects indicate the category ID in FSIP52. At the same time, we also find no intersection between our data set and 27 common stationary-light-trap agricultural pest classes that appear in Jiao et al. (2022) and belong to the rarer pest species in the data set. Nevertheless, the FSIP52 data set contains various sizes and poses, and our pre-trained data set and base class data set include three of China's top 10 most harmful, invasive insect species in agro-ecosystems (Wan and Yang 2016), which indicate that ours is a non-trivial practical approach to preventing the invasion of foreign insect pests. For more details, see Figure 2.

4 Proposed methodology

The overall proposed architecture is shown in Figure 3. We designed our framework based on the classic Faster R-CNN framework just like other fine-tuning-based few-shot detection networks. The weight-shared backbone network extracts and shares the features of the support and query images q_s, with $D_b \cap D_n = \varnothing$. Normally, we use ResNet-50 as our backbone network and a multi-input single-output (MISO) feature pyramid network (FPN) (Lin et al., 2017; Chen et al., 2021) to introduce multiple receptive fields, aiming at the target scale imbalance problem of custom data sets. Attention region proposal network (RPN) focuses on a given support set category and filters out the target candidate frames of the other categories. Attention RPN is designed to filter out object proposals in other categories by focusing on the given support category. Support features are pooled equally into a $1 \times 1 \times C$ vector, and a depth-wise cross-correlation calculation is then performed with the query features, the output of which is used as the attention features, which are fed into the RPN to generate recommendations.

For K-shot training, we obtain all the support features through the weight-shared network and use the average feature across all the support images belonging to the same category as its support feature. When testing, when each query image is given, these support features can be used for classification and positioning (equivalently, each test sample is a query image, which is shared by all the support images of the query image). The essence of the association between the support feature and the query feature is to use the given support image and label information to find objects with similar features in the query image and provide their approximate spatial positions. For N-way training, we add an N-1 support set branch extension network structure, where each branch has an independent attention RPN module and a multi-relation detection module.

4.1 Multi-relation detector

The multi-relation detector has three separate blocks: global block, local block, and patch block. Global block is used to learn the depth feature mapping information of global matching. Local block is aimed at learning the channel-by-channel spatial feature inter-correlation between the support set and candidate areas of the query set. Patch block is used to learn the similarity of the deep nonlinear metric between pixel blocks. These three subblocks calculate the similarity for each candidate area of the query set and then compare their fusion with the task threshold.

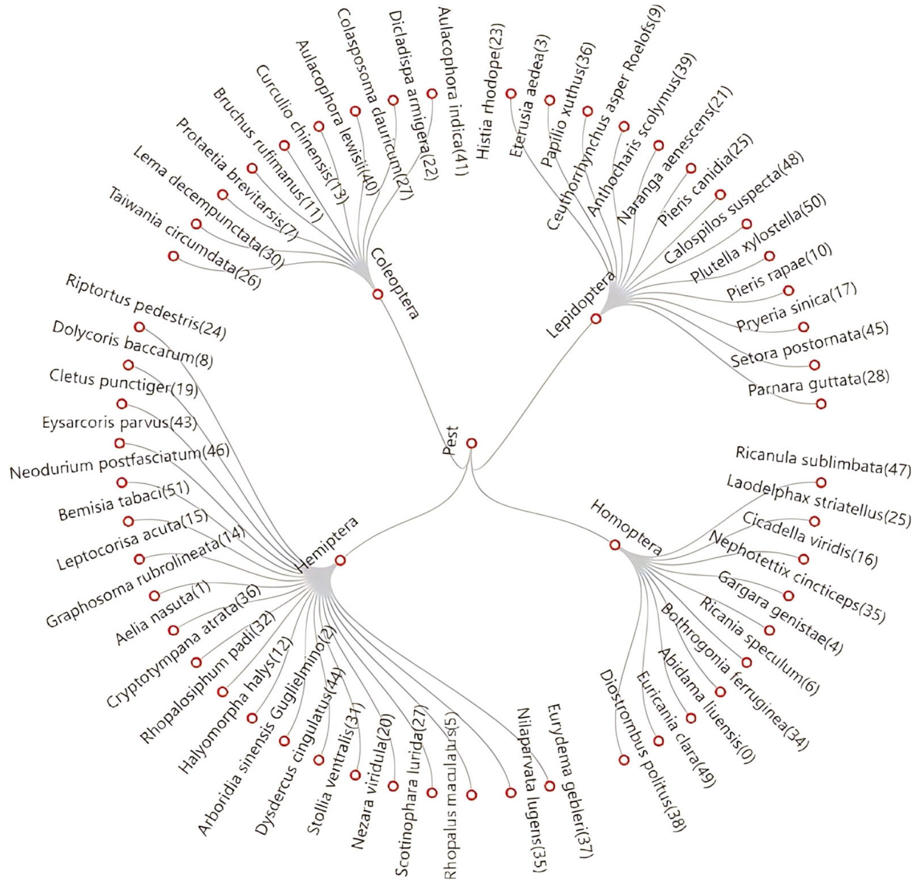


FIGURE 2
The hierarchical taxonomy of FSIP52 is explicitly stratified into four superclasses, namely, Homoptera, Hemiptera, Lepidoptera, and Coleoptera and 52 subclasses that follow the division of the pest class family. The numbers in brackets after the name of each class of insects indicate the category ID in FSIP52.

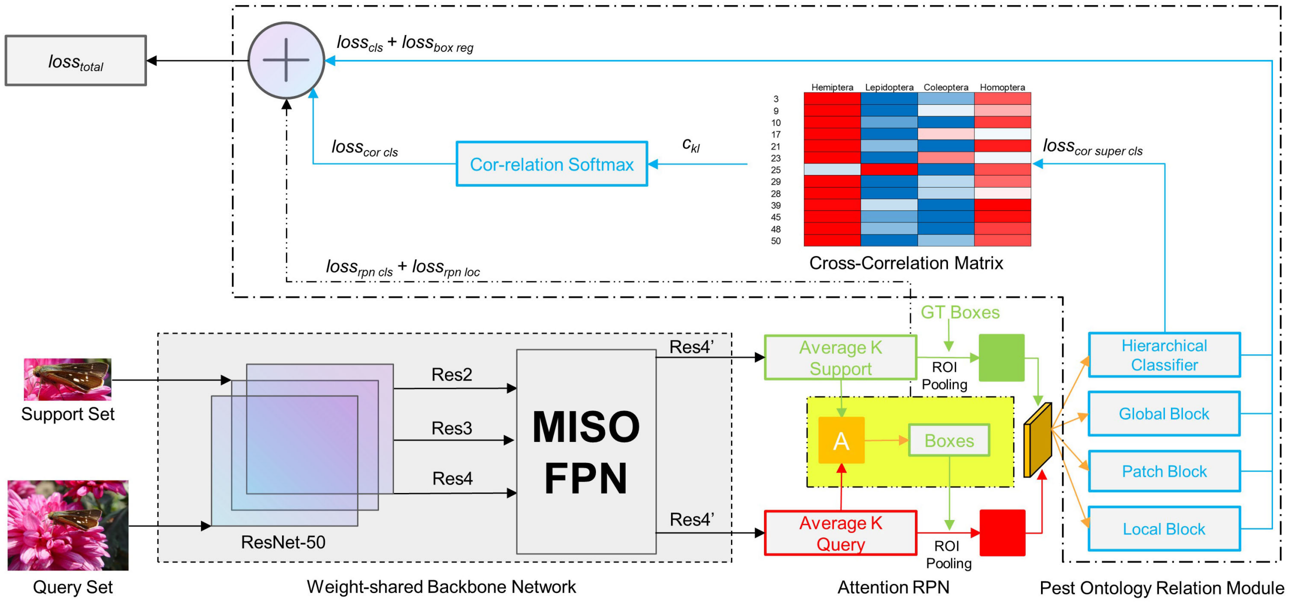


FIGURE 3
Framework of the proposed few-shot pest detection network.

4.2 Hierarchical classifier and cross-correlation matrix

Fan et al. (2020) were unable to make good use of multi-source category information. Rather than using labels directly, samples were re-coded and their categories were predicted by fusing multiple feature similarities and scoring against a preset task threshold. This is essentially a clustering method by means of a specific distance measure. It would work between horses and sheep therein were similar to simple rigid bodies, and the difference between them in terms of external characteristics would still be quite obvious. However, insect pests are typically nonrigid, and insects are diverse and varied, belonging to the arthropod group of invertebrates. This paper expands the aforementioned approach by incorporating pest ontology relation module. By fusing internal and external visual information derived from the image-level pest features and hierarchical insecta information derived from prior human knowledge, multi-category information is encode to directly supervise the model optimization. Therefore, the primary difficulty in detecting pests with few samples lies in the classification of similar pest categories rather than in their localization.

Prior knowledge derived from Insecta guides us to build a hierarchical classifier. With this, we can reduce the range of class predictions through prior human knowledge and focus more on the accuracy of classification tasks for similar classes of pests in different classes of the same order. The method of image similarity calculation has a great impact. Current few-shot detection methods (Li and Yang, 2021; Sun et al., 2021) use the Euclidean distance and the cosine similarity as the metric for the feature distance. As the dimensionality of the data increases, the maximum and minimum Euclidean distance and the cosine similarity approach zero, which makes distinguishing impossible. The Euclidean distance function and the cosine similarity function lose their meanings in a high-dimensional environment. Alternatively, we use the differential hash algorithm to encode image-level visual features, which is essentially a gradual perceptual hash algorithm combining the advantages of an average hash algorithm and a perceptual hash algorithm. We retain recognizable features at the image level through cross-correlation matrix. The internal dhash similarity of Lepidoptera support instances in the FSIP52 data set is shown in Figure 4. We assume that the similarity value between the same categories is 1. We find that although the similarity between different categories within the same superclass varies, their difference in similarity is not significant. Therefore, the problem of distinguishing similar pests remains a big challenge for the performance of few-shot pest detection.

The calculation phases of the cross-correlation matrix elements are as follows. First, we calculate the pairwise differential hash image similarity between each support set image of two random subclasses, c_i and c_j, affiliated to the same superclass, c_l, to obtain the mean average dhash image similarity, c_{kl}. In particular, when k and l are strictly affiliated in prior human knowledge, we have $p_l = 1$; otherwise, the correlation softmax is degenerated. The purpose of this is to distinguish pests with high similarity within the same superclass by increasing the hyperplane distance between different subclasses belonging to the same superclass through loss function design. Also, the subclass distance between different superclasses is widened by having different superclasses. Thus, we fill the cross-correlation matrix with c_{kl}.

4.3 Total loss function design and correlation softmax

The total loss function ($loss_{total}$) deployed in the training process is defined in Equation (1).

$$
\begin{aligned}
loss_{total} = loss_{cls} + loss_{box\quad reg} + loss_{rpn\quad cls} + loss_{rpn\quad loc} \\
+ loss_{cor\quad cls} + loss_{cor\quad super\quad cls},
\end{aligned}
\tag{1}
$$

where $loss_{boxreg}$, $loss_{rpncls}$, and $loss_{rpnloc}$ are typical loss-function terms in Faster R-CNN; $loss_{cor\quad super\quad cls}$ is the label-smooth cross-entropy function; and $loss_{cls}$ is the loss sum of multi-relation detector.

$loss_{cor\ cls}$ with correlation softmax $\alpha_k^{\star}$ is formulated as Equation (2).

$$
\begin{aligned}
loss_{cor\quad cls}(s) = -\beta \sum_{k=1}^{m} p_k \log\left(\alpha_k^{\star}\right), \\
\alpha_k^{\star} = \frac{e^{z_k}}{\sum_{l=1}^{C}(2-p_l)(1-c_{kl})e^{z_l}+e^{z_k}},
\end{aligned}
\tag{2}
$$

where β is a scale variant that balances the numerical magnitude of the correlation softmax loss-function terms with other original loss-function terms, but it does not differentiate between easy and hard examples. Initially, we set $\beta = 0.25$, but β would be optimized after repeated experiments and changes in the data set. Through the correlation softmax function, the original softmax suppression effects between confusing pairs are weakened.

p_k denotes the label of class k regarding bounding box s.

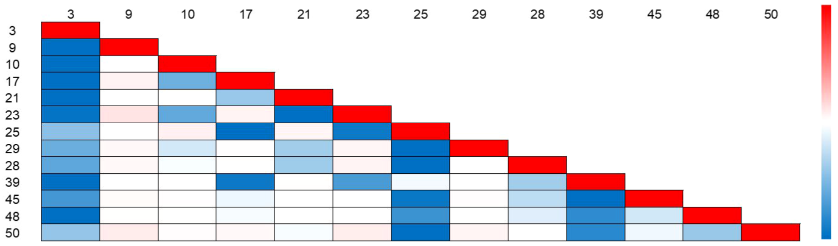


FIGURE 4
Internal dHash similarity of Lepidoptera support instances in the FSIP52 data set. The redder the heat map color block, the higher the visual similarity.

c_{kl} is the mean average image similarity between classes k and l. Conventionally, a simple and intuitive approach would be to transform multiple binary classification problems and fuse the results, but that neglects the relationships between labels because the regular softmax loss function has exclusive semantics between labels. α_i^* outputs logits of correlation softmax.

Output: $P = (\hat{c}_k, \hat{c}_l, \hat{p}_l)$, $c_k \in \{0,1,2,3,\ldots,51\}$, $c_l \in \{0,1,2,3\}$, $p_l \in \{0,1\}$

If there is a hierarchical relation between subclass i and its superclass j, then p_l is set to 1, otherwise, it is set to 0.

The original Faster R-CNN loss function defined in Girshick (2015) is shown as Equation (3) and Equation (4).

$$L(p_i,t_i) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i,p_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\left(t_i,t_i^*\right), \tag{3}$$

Where λ = 1 and

$$L_{cls}\left(p_i,p_i^*\right) = -\log\left(p_i^* * p_i + \left(1-p_i^*\right) * (1-p_i)\right) \tag{4}$$

5 Evaluation metrics

5.1 Few-shot detection metrics

To better explain and illustrate the performance of our proposed model, we briefly describe the evaluation metrics for few-shot detection. We strictly followed the three random concepts in few-shot learning, namely, random L-fold cross-validation, randomly selecting N samples, and K images as support sets. The N-way K-shot definition is as follows: Randomly select N types of samples from the meta-data set, randomly select K+m instances from each type of sample, and then randomly select K instances from the K+m instances of each type of sample as the support set.

To make the obtained accuracy reasonably standardized, we use the mean average precision (mAP) as the metric of the proposed model. The calculation of mAP as defined in COCO is shown in Equation (5).

$$mAP = \frac{1}{10\times N}\sum_{k=0.5:.05}^{.95}(r_i - r_{i-1})\times p, \tag{5}$$

where N denotes the total number of categories. k denotes the IoU threshold. r_i denotes the recall value corresponding to the first interpolation of the precision interpolation segment in ascending order. p denotes the regression value of the observation point on the smoothed Precision-Recall (PR) curve.

$$\begin{aligned} AverageRecall &= 2\int_{0.5}^{1} recall(x)dx \\ &= \frac{2}{n}\sum_{i=1}^{n} max(IoU(gt_i) - 0.5, 0) \end{aligned} \tag{6}$$

The definition of $AverageRecall$ (AR) is first proposed by Hosang et al. (2015), and it can be calculated using Equation (6). The $AverageRecall$ between 0.5 and 1 can also be computed by averaging the overlaps of each annotation gt_i with the closest matched proposal, that is, integrating over the y axis of the plot instead of the x axis. x denotes the IoU overlap. $IoU(gt_i)$ denotes the IoU between the annotation gt_i and the closest detection proposal. AR is twice the area enclosed by the recall-IoU curve. n is the number of overlaps between all GroundTruth bboxes and the nearest DetectionResult bbox in each image, that is, the COCO metric of maxDets. AR is a measure of the accuracy of the positioning of the model's detection boxes. The mean average recall (mAR) can be obtained by averaging the AR of all categories in each novel split.

6 Experiments

6.1 Implementation details

IP102 contains many web images and specimen images, and its image resolution ranges from 87×120 to 6034×6053 with different growth stages. There are many solid-color specimen backgrounds and single close-up images of insects in the IP102 data set, and there are many duplicate or extremely similar images of pests. We deleted some categories with very few samples, and we removed some orders of insects unrelated to what is discussed herein, specifically Hymenoptera, Diptera, Coccinellae, Acarina, Thysanoptera, Acarina, and Orthoptera. For fairness, we removed five duplicate categories between IP102 and our FSIP52 data set, namely, *Protaetia brevitarsis*, *Cicadella viridis*, *Pieris canidia*, *Papilio xuthus*, and *Nilaparvata lugens*. Finally, we removed 34 irrelevant categories from IP102, leaving IP68 to serve as our pre-trained data set. Figure 5 and Table 1 give more details about the distribution of the FSIP52 data set and novel class splits settings in this experiment.

Since pest postures are diverse, we performed random rotation augmentations on pests in advance to compensate for the less-robust rotation invariance of a traditional convolutional neural network. The postures of pests were taken from various angles, and it is not scientific to use only similarity for supervision; the problem of pest posture can be partly solved by rotation enhancement. To analyze the proposed softmax loss and model with a hierarchical structure, we conducted extensive experiments on our well-designed FSIP52 data set. We trained our model on a computer with an Intel 9900K CPU, 128 GB of RAM, and a single NVIDIA Titan RTX GPU. In terms of software experimental conditions, we deployed our algorithm on Ubuntu 18.04.06 LTS equipped with Pycharm 2021.3 Community Edition, CUDA 11.3.1, CUDNN 8.2.1, GCC 7.5.0, Python 3.8.5, Pytorch 1.4.0, and Detectron2 0.6. For pest detection, using default anchor box settings would greatly affect the initial IoU value in the early training stage, resulting in the inability to screen out the optimal prediction box. Furthermore, the original IP102 data set was designed in the Visual object class (VOC) data set style, so its anchor boxes had to be re-clustered according to our data set. Moreover, the K-means++ clustering algorithm can randomly generate custom clustering centers, which ensures a discrete type of initial cluster center, better elevating the effect of anchor box generation. Therefore, new anchor boxes for FSIP52 were generated, including (65,86), (78,148), (119,232), (144,142), (179,339), (220,217), (292,326), (328,512), (601,698), and their re-cluster anchor aspect ratios are [0.51,0.53,0.64,0.76,0.86,0.90,1.01]. Re-clustering priori boxes helps speed up convergence.

We reported our experimental results with ResNet-50 after computing the time consumption and training accuracy, although

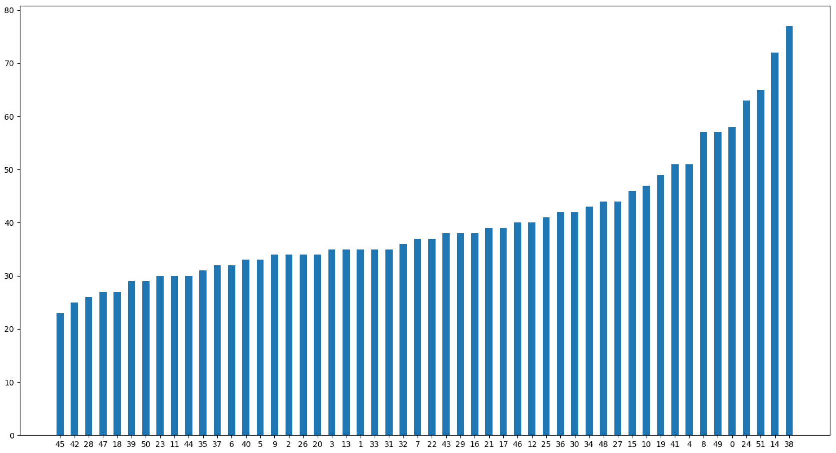


FIGURE 5
FSIP52 data set distribution is presented in ascending order according to the sample numbers. The horizontal axis represents the category ID of FSIP52 and the vertical axis represents the number of instances. Pest samples are difficult to collect due to geography, season, frequency, and pest mobility.

we point out that our model would perform better with other more advanced and complicated backbone networks, for example, ResNet-101 or ResNeXt. The loss curves for the base-class training stage and the novel-class fine-tuning stage are shown in Figure 6.

Our proposed network was trained in a class-specific and end-to-end fashion, and the original input image resolution varied from 640×480 to 3680×2456. We utilized a multi-scale training scheme to resize the input images to x ϵ {660×440,708×472,756×504,804×536,852×568,900×600,1000×667}. Then, the training images were resized to the same aspect ratio as the original input images, and their width and height were determined by the shorter side of the images. We trained our model for 100 epochs using the same default settings for Detectron2 in both the base-class training stage and the novel-class fine-tuning stage to ensure total complete convergence for fair comparison. An early-stopping mechanism was set to capture the best checkpoint with every 5,000 iterations, and the Dropout (Hinton et al., 2012), SoftPool (Stergiou et al., 2021), and DropBlock (Ghiasi et al., 2018) techniques were also introduced in the pre-training, base-class training, and fine-tuning stages.

In the base-class training stage, the learning rate was set to 0.001 with 100 epochs and a batch size of eight. The fraction between positive and negative samples was 0.5 and was kept the same in both the training and testing sets in both stages. Weight-shared ResNet-50 was pretrained on the FSOD data set to extract features from the support and query images, and its output features were the set {*res*2, *res*3,*res*4,*res*5}. Deformable convolution was applied in the feature-extraction and FPN stages, and the non-maximum suppression threshold in RPN was set to 0.7. The smooth L1 beta was 1/9, the *IoU* threshold in Region of Interest (ROI) head was set to 0.3, the weight decay applied to the parameters of the normalization layers was 1×10^{-5}, the momentum was set to 0.937, the warm-up iterations were set to 2 epochs, the default support ways for contrastive learning branch were 2, and the ResNet-50 backbone network was frozen at *res*3. We decoupled the fully connected layers concerning both the cross-correlation matrix and the hierarchical matrix with the original Faster-RNN classifier layer. We applied Kaiming normal weight initialization (He et al., 2015) to all convolutional and fully connected layers and inputted the concatenation of the support and query features. MISO FPN outputted the *res*4 feature for further processing, and group normalization was enabled in FPN. FPN and RPN were jointly optimized in both stages.

In the novel-class fine-tuning stage, the learning rate was set to 0.001 with 100 epochs and a batch size of 12. Most pretrained model parameters or layers were frozen, while only the last few layers' parameters were updated during the novel-class training.

6.2 Comparison experiments and discussion

Research on few-shot object detection has emerged in the past 2 years, and we decided to compare our method with several typical few-shot object detection networks, namely, those by Fan et al. (2020); Sun et al. (2021); Wu et al. (2020) and Wang et al. (2020b). All comparison experiments were conducted on the MMFewShot framework produced by Open MMLab and the Detectron2 framework produced by Facebook, using exactly the same experimental settings. Our model outperformed most state-of-the-art (SOTA) methods without much extra calculation.

TABLE 1 Detailed FSIP52 data set split experimental settings.

FSIP52	Novel split 1	Novel split 2	Novel split 3	Novel split 4
Category ID	0–12	13–25	26–38	39–51
Base class Training	1,556	1,529	1,564	1,588
Novel class fine-tuning	362	389	354	330

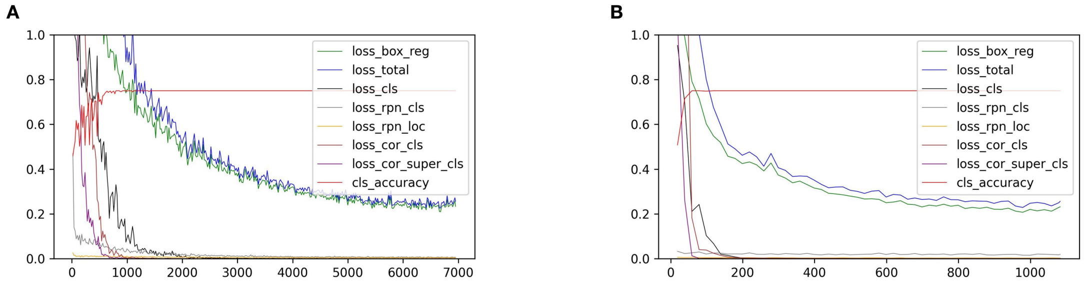


FIGURE 6
(A) shows the loss curves of the novel split 3 in the base-class training stage. **(B)** shows the same in the novel-class fine-tuning stage. The horizontal axis represents the number of iterations, and the vertical axis represents the loss value.

Before fully analyzing and discussing the results of the experiments, it must be pointed out again that our custom data sets were all taken from real natural scenarios that have been strictly selected by the Yun Fei Company, Anhui Academy of Agricultural Sciences, and the Hefei Plant Protection Station pest experts, making the samples rather representative and complex. Note that insects are nonrigid bodies, and their tentacles can easily expand the bounding box unnecessarily and cause a reduction in the signal-to-noise ratio, which then leads to quite large bounding boxes. On the other hand, due to the migratory nature of some pests, close-up photography is not possible, so certain tiny pests add difficulty to the current few-shot pest detection in the wild.

In Table 2, our model achieves the best results on the FSIP52 data set based on a few-shot protocols of 13-way 10 shots on novel splits 1, 3, and 4 and is ahead of SOTA methods by 4%, 2.8%, and 2.2% on mAP, respectively. In Table 3, it is ahead of SOTA methods by 5.9%, 2.8%, and 0.6% on AP50, respectively. In Table 4, our model outperforms SOTA on novel splits 1, 2, and 3 by 7%, 10.9%, and 7.8% on mAR, respectively. The reason for this is that our model was especially designed for pest in wild settings. We availed of multi-task learning to design a logically interpretable prior knowledge learning task, and import the knowledge gained by human experts in the process of pest identification as supervision information to guide the network to achieve better detection performance in the case of extremely limited novel class samples. The use of cosine classifier and contrastive loss coverages very slowly in the set number of iterations by Sun et al. (2021) may not suitable for pest detection, and its coefficients are too many to be fine-tuned.

Nonetheless, note that our model trails that of Wu et al. (2020) by 5.3% and 11.4% on mAP on Novel split 2. A comparison of each category in Novel split 2 shows that the model of Wu et al. (2020) leads our model in categories 13, 14, 18, 19, 21, and 22 by 20.6%, 6.7%, 1.5%, 10.3%, 13.6%, and 29.7%, respectively. Yet the mAR of our model prevails over that of Wu et Al. by 13.5%. We attribute this to the presence of extra-large and tiny targets in these categories; the predominance of frontal and abdominal photographs of the pests, which does not capture the most recognisable parts of the pests; and the fact that our model does not have a re-weighted strategy for these multi-scale positive samples through especially designed reinforcement block. Although we slightly underperformed

TABLE 2 FSIP52 novel classes' mean average precision (mAP) in 13-way 10-shot settings.

Reference	Novel split 1	Novel split 2	Novel split 3	Novel split 4
Wu et al. (2020)	6.7	**22.3**	9.9	10.4
Fan et al. (2020)	12.5	15.4	16.1	11.2
Wang et al. (2020b)	11.6	12.0	12.5	11.7
Sun et al. (2021)	7.2	9.7	9.8	5.3
Ours	**16.5**	17.0	**18.9**	**13.9**

Bold values are to highlight which models achieved the highest accuracy in the different data set splits, in order to provide strong evidence of the advantages of a particular method.

TABLE 3 FSIP52 novel classes' AP50 in 13-way 10-shot settings.

Reference	Novel split 1	Novel split 2	Novel split 3	Novel split 4
Wu et al. (2020)	17.4	**44.3**	23.9	19.7
Fan et al. (2020)	20.3	25.0	27.4	20.0
Wang et al. (2020b)	24.6	29.4	30.0	22.7
Sun et al. (2021)	16.8	21.7	22.2	13.3
Ours	**30.5**	32.9	**35.9**	**23.3**

Bold values are to highlight which models achieved the highest accuracy in the different data set splits, in order to provide strong evidence of the advantages of a particular method.

TABLE 4 FSIP52 novel classes' mean average recall (mAR) in 13-way 10-shot settings.

Reference	Novel split 1	Novel split 2	Novel split 3	Novel split 4
Wu et al. (2020)	38.5	52.4	36.4	39.1
Fan et al. (2020)	57.6	55.0	56.8	**57.3**
Wang et al. (2020b)	42.2	42.4	40.8	40.5
Sun et al. (2021)	37.3	37.9	40.5	37.3
Ours	**64.6**	**65.9**	**64.6**	55.9

Bold values are to highlight which models achieved the highest accuracy in the different data set splits, in order to provide strong evidence of the advantages of a particular method.

compared to Fan et al. by 1.4% in the Novel split 4 mAR comparison, we achieved comparable performance to SOTA in preventing missed detections and were 2.7% and 3.3% ahead of that of Fan et al. in mAP and AP50, respectively, which are often more important in practice than mAP and AP50. Finally, despite the fact that our performance improved compared with the SOTA methods mentioned, we still have a long way to go to be qualified for real-world agricultural production missions.

7 Conclusion

In this paper, a few-shot insect pest detection network is introduced to detect rarely collected pest species. Its novelty lies in combining the hierarchical semantic relationship between superclasses and subclasses according to insect taxonomy, guiding our model to better learn novel concepts through causal intervention, especially when the novel class samples are extremely limited. A new hierarchical data set FSIP52 for few-shot pest detection in natural settings is built based on insect taxonomy. It is emphasized that the presented few-shot pest detection network achieves comparable performance to several representative few-shot detection algorithms in FSIP52 data set through incorporating pest ontology relation module designed specifically for hierarchical structure matching in the proposed framework, and we point out that it could be extended to other similar practical scenarios with hierarchical structures. Last but not the least, apart from the developed fine-tuning-based object detection algorithms, there are other branches of few-shot learning methods (e.g., cross-domain and meta-learning) that are still at a relatively preliminary stage and are quite worthy of follow-up research. The present work highlights a new entry in the field of few-shot pest detection.

Data availability statement

The data sets presented in this article are not readily available because permission must be obtained from the head of the laboratory and the supervisor. Requests to access the data sets should be directed to mydream@mail.ustc.edu.cn.

Author contributions

XW is responsible for the methodology, model building, data set construction, experimental implementation and the first draft of this paper. JD is responsible for the conceptualized guidance of paper writing and the discussion of the overall architecture. CX programmatically pointed out the initial research direction. SW took part in code debugging and idea discussion. XM participated in discussions on initial concept formation and model validation. KL is responsible for the verification of model formulas and manuscript revision. SD proofread the whole paper. TC provided basic agricultural knowledge consultation. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China (No. 32171888), the Dean's Fund of Hefei Institute of Physical Science, Chinese Academy of Sciences (YZJJ2022QN32), and the National Key Research and Development Program of China (2019YFE0125700).

Acknowledgments

The authors thank the Yun Fei Company for providing data support. The authors also thank all the authors cited in this paper and a referee for their helpful comments and suggestions. This manuscript was edited for English language by Charlesworth Author Services.

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

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. (2019). *A closer look at few-shot classification. 7th International Conference on Learning Representations*, New Orleans, LA, USA, May 6-9, 2019 Available at: https://OpenReview.net.

Chen, Y., Wang, X., Liu, Z., Xu, H., and Darrell, T. (2020). *A new meta-baseline for few-shot learning. CoRR* abs/2003.04390. doi: 10.48550/arXiv.2003.04390

Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., and Sun, J. (2021). "You only look one level feature," in Conference on Computer Vision and Pattern Recognition. IEEE. 13039–13048. doi: 10.1109/CVPR46437.2021.01284

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. (2019). A baseline for few-shot image classification. *8th International Conference on Learning Representations*. Addis Ababa, Ethiopia Available at: https://OpenReview.net.

Dong, S., Wang, R., Liu, K., Jiao, L., Li, R., Du, J., et al. (2021). Cra-net: A channel recalibration feature pyramid network for detecting small pests. *Comput. Electron. Agric.* 191, 106518. doi: 10.1016/j.compag.2021.106518

Du, J., Liu, L., Li, R., Jiao, L., Xie, C., and Wang, R. (2022). Towards densely clustered tiny pest detection in the wild environment. *Neurocomputing* 490, 400–412. doi: 10.1016/j.neucom.2021.12.012

Fan, Q., Zhuo, W., Tang, C.-K., and Tai, Y.-W. (2020). "Few-shot object detection with attention-rpn and multi-relation detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: IEEE, 4013–4022. doi: 10.1109/CVPR42600.2020.00407

Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2018). Dropblock: A regularization method for convolutional networks. *Adv. Neural Inf. Process. Syst.* 31, 10750–10760. doi: 10.48550/arXiv.1810.12890

Girshick, R. (2015). "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision* Santiago, Chile, December: IEEE, Vol. 521. 1440–1448. doi: 10.1109/ICCV.2015.169

He, B., Wu, D. Q., and Ni, Y. (2012). "Research in agricultural technician distribution characteristics in guangdong province," in *Advanced Materials and Engineering Materials*, vol. 457. (Trans Tech Publ), 748–753. doi: 10.4028/www.scientific.net/AMR.457-458.748

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*. Santiago, Chile, December: IEEE, 1026–1034. doi: 10.1109/ICCV.2015.123

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov., R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*. abs/1207.0580. doi: 10.48550/arXiv.1207.0580

Hosang, J., Benenson, R., Dollar, P., and Schiele, B. (2015). What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (4), 814–830. doi: 10.1109/TPAMI.2015.2465908

Huo, M., and Tan, J. (2020). "Overview: Research progress on pest and disease identification," in *International Conference on Pattern Recognition and Artificial Intelligence*. 404–415 (Springer). doi: 10.1007/978-3-030-59830-3_35

Jiao, L., Li, G., Chen, P., Wang, R., Du, J., Liu, H., et al. (2022). Global context-aware-based deformable residual network module for precise pest recognition and detection. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.895944

Kandalkar, G., Deorankar, A. V., and Chatur, P. N. (2014). Classification of agricultural pests using dwt and back propagation neural networks. *Int. J. Comput. Sci. Inf. Technol.* 5 (3), 4034–4037.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. (2019). "Few-shot object detection *via* feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Seoul, South Korea: IEEE. 8420–8429. doi: 10.1109/ICCV.2019.00851

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

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., et al. (2008). Automated insect identification through frontiers 16 Wang et al. priori-knowledge-auxiliaried few-shot pest detection in the wild concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects. *Mach. Vision Appl.* 19 (2), 105–123. doi: 10.1007/s00138-007-0086-y

Lin, T.-Y., Dollar, 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*. 2117–2125.

Liu, L., Xie, C., Wang, R., Yang, P., Sudirman, S., Zhang, J., et al. (2020). Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features. *IEEE Trans. Ind. Inf.* 17 (11), 7589–7598. doi: 10.1109/TII.2020.2995208

Li, R., Wang, R., Xie, C., Chen, H., Long, Q., Liu, L., et al. (2022). A multi-branch convolutional neural network with density map for aphid counting. *Biosyst. Eng.* 213, 148–161. doi: 10.1016/j.biosystemseng.2021.11.020

Li, Y., and Yang, J. (2020). Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* 169, 105240. doi: 10.1016/j.compag.2020.105240

Li, Y., and Yang, J. (2021). Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* 182, 106055. doi: 10.1016/j.compag.2021.106055

Murakami, S., Homma, K., and Koike, T. (2005). "Detection of small pests on vegetable leaves using glcm," in *2005 ASAE annual meeting* (American Society of Agricultural and Biological Engineers), 1. doi: 10.13031/2013.19109

Ngugi, L. C., Abelwahab, M., and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition–a review. *Inf. Process. Agric.* 8 (1), 27–51. doi: 10.1016/j.inpa.2020.04.004

Parnami, A., and Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. *Comput. Surv.* 53 (3), 63–34. doi: 10.48550/arXiv.2203.04291

Parsa, S., Morse, S., Bonifacio, A., Chancellor, T. C. B., Condori, B., Crespo-Perez, V., et al. (2014). Obstacles to integrated pest management adoption in developing countries. *Proc. Natl. Acad. Sci.* 111 (10), 3889–3894. doi: 10.1073/pnas.1312693111

Sbai, O., Couprie, C., and Aubry, M. (2020). "Impact of base dataset design on few-shot image classification," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*. 597–613 (Springer). doi: 10.1007/978-3-030-58517-4_35

Stergiou, A., Poppe, R., and Kalliatakis, G. (2021). "Refining activation downsampling with softpool," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10357–10366.

Sun, B., Li, B., Cai, S., Yuan, Y., and Zhang, C. (2021). "Fsce: Few-shot object detection *via* contrastive proposal encoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7352–7362.

Wang, B. (2021). Identification of crop diseases and insect pests based on deep learning. *Sci. Programming*. London, GBR: Hindawi Ltd. 2022. doi: 10.1155/2022/9179998

Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., and Yu, F. (2020b). Frustratingly simple few-shot object detection. *Proceedings of the 37th International Conference on Machine Learning, 2020, 13-18 July 2020, Virtual Event*. 119, 9919–9928. Available at: http://proceedings.mlr.press/v119/wang20j.html.

Wang, F., Liu, L., Dong, S., Wu, S., Huang, Z., Hu, H., et al. (2022). Asp-det: Toward appearance-similar light-trap agricultural pest detection and recognition. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.864045

Wang, F., Wang, R., Xie, C., Zhang, J., Li, R., and Liu, L. (2021). Convolutional neural network based automatic pest monitoring system using hand-held mobile image analysis towards non-site-specific wild environment. *Comput. Electron. Agric.* 187, 106268. doi: 10.1016/j.compag.2021.106268

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020a). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (Csur)* 53 (3), 1–34. doi: 10.1145/3386252

Wan, F.-H., and Yang, N.-W. (2016). Invasion and management of agricultural alien insects in china. *Annu. Rev. Entomol.* 61 (1), 77–98. doi: 10.1146/annurev-ento-010715-023916

Watson, A. T., O'Neill, M. A., and Kitching, I. J. (2004). Automated identification of live moths (macrolepidoptera) using digital automated identification system (daisy). *Syst. Biodiversity* 1 (3), 287–300. doi: 10.1017/S1477200003001208

Wen, C., and Guyer, D. (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, J., Liu, S., Huang, D., and Wang, Y. (2020). "Multi-scale positive sample refinement for few-shot object detection," in *European conference on computer vision*. 12361, 456–472 (Springer). doi: 10.1007/978-3-030-58517-4_27

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

Xiao, Y., and Marlet, R. (2020). "Few-shot object detection and viewpoint estimation for objects in the wild," in Computer Vision - 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII. 192–210 (Springer). doi: 10.1007/978-3-030-58520-4_12

Yang, Z., Yang, X., Li, M., and Li, W. (2021). Small-sample learning with salient-region detection and center neighbor loss for insect recognition in real-world complex scenarios. *Comput. Electron. Agric.* 185, 106122. doi: 10.1016/j.compag.2021.106122

Yu, Y. (2021). Research progress of crop disease image recognition based on wireless network communication and deep learning. *Wirel. Commun. Mob. Comput.*. GBR: John Wiley and Sons Ltd.. 2021, 15. doi: 10.1155/2021/7577349

Zhang, Q., Yu, F., Fu, R., Liu, X., and Zhang, J.-F. (2016). "Agricultural information service based on wechat platform in beijing," in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*. 464–466 (IEEE). doi: 10.1109/ISAI.2016.0104

Zhang, J., Zhang, X., Lv, L., Di, Y., and Chen, W. (2022). An applicative survey on few-shot learning. *Recent Patents Eng.* 16 (5), 104–124. doi: 10.2174/1872212115666210715121344

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proc. IEEE* 109 (1), 43–76.

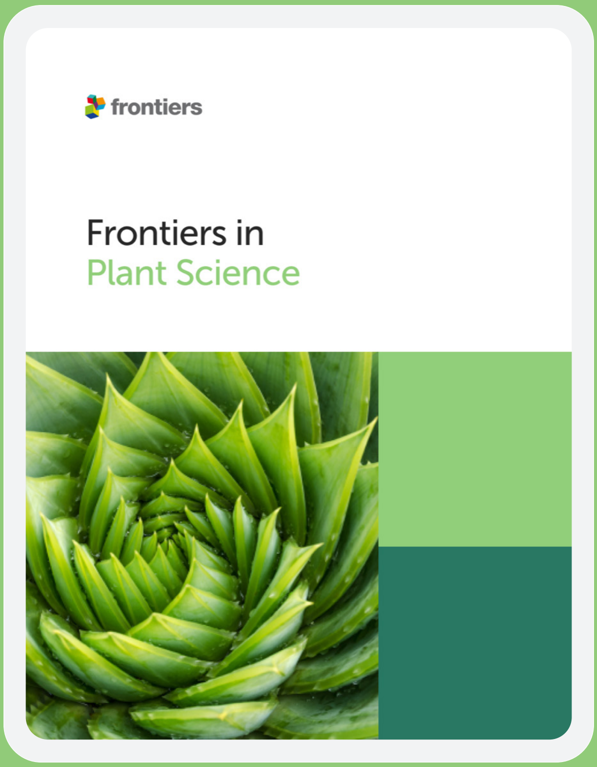
Frontiers in
Plant Science

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

Frontiers in
Plant Science

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact