frontiers | Research Topics

Convolutional neural networks and deep learning for crop improvement and production

Edited by
Wanneng Yang, Kioumars Ghamkhar and Gregorio Egea

Published in
Frontiers in Plant Science

ISSN 1664-8714
ISBN 978-2-83250-936-4
DOI 10.3389/978-2-83250-936-4

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

Convolutional neural networks and deep learning for crop improvement and production

Topic editors
Wanneng Yang — Huazhong Agricultural University, China
Kioumars Ghamkhar — AgResearch Ltd, New Zealand
Gregorio Egea — University of Seville, Spain

Citation
Yang, W., Ghamkhar, K., Egea, G., eds. (2023). *Convolutional neural networks and deep learning for crop improvement and production*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-83250-936-4

Table of contents

frontiers | Frontiers in Plant Science

TYPE Editorial
PUBLISHED 18 November 2022
DOI 10.3389/fpls.2022.1079148

OPEN ACCESS

EDITED AND REVIEWED BY
Shawn Carlisle Kefauver,
University of Barcelona, Spain

*CORRESPONDENCE
Wanneng Yang
ywn@mail.hzau.edu.cn
Gregorio Egea
gegea@us.es
Kioumars Ghamkhar
kioumars.ghamkhar@agresearch.co.nz

SPECIALTY SECTION
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

RECEIVED 25 October 2022
ACCEPTED 04 November 2022
PUBLISHED 18 November 2022

CITATION
Yang W, Egea G and Ghamkhar K
(2022) Editorial: Convolutional neural networks and deep learning for crop improvement and production.
Front. Plant Sci. 13:1079148.
doi: 10.3389/fpls.2022.1079148

Editorial: Convolutional neural networks and deep learning for crop improvement and production

Wanneng Yang[1*], Gregorio Egea[2*] and Kioumars Ghamkhar[3*]

[1]National Key Laboratory of Crop Genetic Improvement, National Center of Plant Gene Research, Hubei Hongshan Laboratory, Huazhong Agricultural University, Wuhan, China, [2]Area of Agroforestry Engineering, School of Agricultural Engineering, University of Seville, Seville, Spain, [3]Margot Forde Germplasm Centre, Grasslands Research Centre, AgResearch, Palmerston North, New Zealand

KEYWORDS

convolutional neural networks (CNN), deep learning (DL), big data, pattern recognition, object recognition, crop improvement, crop production, artificial intelligence

Editorial on the Research Topic
Convolutional neural networks and deep learning for crop improvement and production

With the development of high-throughput phenotyping (HTP) technology, the large amount of phenotypic data has provided the breeders with new opportunities of accurate and repeatable phenotyping or phenomics (Ghamkhar et al., 2019; Roitsch et al., 2019; Yang et al., 2020). There is also the question of what to do with the increasingly substantial amounts of generated data using these technologies. Traditionally, we have used manual/visual methods to estimate or measure plant phenotype. These traditional methods are time-consuming and labour intensive hence the need to technological innovation in phenotypic technologies (Furbank and Tester, 2011; Ubbens and Stavness, 2017). Nowadays, with the advent of image processing enabling software handling large amount of data is more manageable. For phenotyping platforms of industrial scale in controlled environments, a simple background, a controlled environment, and a streamlined image processing method make it possible to fully automate high-throughput phenotypic data acquisition and analysis. However, for complex working conditions, such as field and phenotyping platforms with complex backgrounds, the challenges of image processing increase dramatically. Further, the robustness of the program will also decline due to the challenges of repeatability, which inadvertently will increase the costs of programming and the labour cost for manual intervention. In more complex cases, such as segmentation of specific parts of a plant, image processing methods are more challenging to achieve congruent results due to the complexity of features. Recent advances in deep learning technologies will ease overcoming this bottleneck.

The purpose of object detection is to find out the location of an object in the image and classify it. This is a combination of object localization and image classification. Compared with image classification, where the classification can only get an image of the subject, the object detection task is used to detect the image of a number of different categories of individuals, often used to count or target tasks, and is widely used in automated tasks, so as to realize the recognition of pedestrians, vehicles and traffic light detection (Xiao et al., 2020).

Species identification and discrimination

The application of deep learning technology in phenotyping is mostly in image processing. The core process of image classification task is to assign labels to the images of interest (Bateman et al., 2020; Chen et al., 2021). The use of deep learning network can classify images end-to-end, without the need to extract the features of the target and quantify them into data as in traditional image processing methods. Large-scale data acquisition using UAVs are examples for using deep learning in order to decimate the data processing time. Zhang et al. demonstrate the use of UAV and CNN to identify and map weeds in various areas of the field, which can effectively help the more efficient control and removal of weeds. Application programming interface (API) implementation of the PyTorch deep learning library has been used in this study with a range of precision depending on the weed species and type. Not surprisingly, the authors suggest that more than one model would be needed to improve the weed mapping involving more than one species. Fujiwara et al. applied convolutional neural network on UAV data and quickly classified and segmented grasses in UAV images, thereby quantifying the coverage legumes in the area of interest, effectively achieving the appropriate management of a grass and legume mixture. Yue et al. have applied deep learning as well as partial two pattern recognition models (least squares discriminant analysis (PLS-DA) and support vector machine (SVM)) to identify the medicinal plant *Paris polyphylla* var. *yunnanensis* using spectroscopy data. Their results show that the deep learning model had clear advantage in the identification of this plant. The direct use of two-dimensional correlation spectroscopy (2DCOS) shows the strength of deep learning for multi-class image data.

Crop disease recognition

Convolutional neural network (CNN) can effectively identify plant disease categories that would have only been possible by the experts in the past. Wang et al. use a deep learning model called Coordinated Attention EfficientNet (CA-ENet) to identify different apple diseases. Their method's accuracy reached 98.92%, and the average F1-score reached 0.988, which is superior to many mainstream models and has a certain robustness. Their model learnt both the channel and spatial location information of important features. The targeted design network can better realize the purpose of agricultural application. For example, the proposed deployment based on a dilated convolution capsule network (DCCapsNet), proposed by Xu et al., can quickly capture and define diseased apple leaves, and potentially enable early prevention of apple diseases. Deep learning object detection has obvious advantages in counting, positioning, and judgment, which is a milestone that is difficult to achieve by traditional image processing methods. Zhou et al. used deep learning image classification technology to identify rice diseases. When different diseases cause similar or the same symptoms, simultaneous training is better than separate training. When the symptoms are significantly different, any method can achieve high accuracy.

Reproductive yield measurement

The detection model to identify grains in the rice panicle and whether the grain is full or bare is used by Guo et al., in order to define rice seed setting rate (RSSR) more accurately and measure reproductive yield in a high-throughput manner. In the study of plant phenotype, object detection task has also been very widely used. In general, the object detection task in plant phenotype is to find and define the regions of great significance in the plant, specifically for breeding purposes. Zang et al. use the improved classic YOLOv5s detection model, by introducing an efficient channel attention module (ECA), to identify wheat spikes with a detection accuracy of 71.61%, allowing for rapid and accurate wheat reproductive yield estimation. This method is specifically useful in complex field environments. New methods and new ideas beyond deep learning are also emerging. Ensemble learning, for example (Shahhosseini et al.), predicts grain yield directly from images and some environmental data. Different from mature deep learning application schemes such as network application and modification network, how to mine new applications of deep learning in phenotyping is an important part of the future developments.

Identification of different stages of growth

The use of data collected by UAVs helped effective identification of the growth stages of rice seedlings (Tan et al.), thereby providing valuable time-sensitive advice for cultivation management. This is an alternative high-throughput method to the current labor intensive and subjective manual measurement practice. Histograms of oriented gradients (HOGs) were combined with the support vector machine (SVM) classifier to recognize and classify three growth stages.

Segmentation for morphometrics of micro and macro organs

Compared with the image segmentation based on the traditional image processing technology, image segmentation based on deep learning techniques can handle different scales in the target segmentation task and has the ability to solve the problem of complicated background, therefore it has great application prospect in agriculture (Ghosh et al., 2019). The use of RGB, near-infrared images or a combination of both has been shown to be accurate in seed quality assessment (Hansen et al., 2016). In this issue, Wang et al. combine these two imaging modalities and the watershed algorithm to segment corn seeds and then use deep learning to identify seed defect. The authors report an accuracy of >95%. From macro to micro, deep learning image segmentation technology can also be applied to the segmentation of microscopic images such as stomata (Gibbs et al.), which can realize fully automatic morphological measurement of stomata and maximal conductance estimation of stomata. As this study shows, deep learning image segmentation technology can extract specific targets at the pixel level, and the information obtained is larger, but the drawback is that the difficulty of data labeling is also greatly increased.

Segmentation of wheat leaves under outdoor conditions is a challenging task, but it is also a prerequisite for high-throughput field phenotype. The classical semantic segmentation model DeepLab V3 can effectively segment wheat leaves under complex field background with a mIOU of 0.77, which lays a foundation for quantifying canopy cover and deriving traits in the field (Zenkl et al.). Similarly, by deploying an improved fully convolutional network with channel and spatial attention on an intelligent harvesting robot, the branches and fruits of guava trees have been segmented in real time to plan collision-free paths for fruit picking (Lin et al.).

In pixel-level image segmentation, extracting the image from the area of interest is an important and difficult challenge in automatic image processing. Nowadays, in the application of deep learning in plant phenotyping, data are generally collected by researchers themselves, and the difficulty of data acquisition and data labeling is self-evident. Apart from the industry's data, there are a large number of public data sets, and transfer applications in industry only need to conduct small transfer learning on pre-trained models to obtain reasonable results. Unfortunately, few phenotype-related data are available in publicly available datasets, which makes it more important to develop large-scale phenotype-specific datasets and pre-trained models, which can greatly reduce the input of data acquisition for researchers, such as AgriNet's pioneering work (Al Sahili and Awad). Consistent with the computer industry, actively adopting new technologies, adapting measures to local conditions, and expanding innovation may make deep learning technology play even more a more significant role in future phenotyping.

Author contributions

WY initially drafted the manuscript. All authors listed have made substantial, direct, and intellectual contribution to the work. KG finalized and submitted the manuscript. All authors have approved the final manuscript for publication

Funding

This work was supported by grants from the National Natural Science Foundation of China (U21A20205), Key projects of Natural Science Foundation of Hubei Province (2021CFA059), and Fundamental Research Funds for the Central Universities (2021ZKPY006), HZAU-AGIS Cooperation Fund (SZYJY2022014) for WY. New Zealand's Ministry of Business, Innovation and Employment (Project number C10X1701) and AgResearch Ltd provided financial support for KG.

Acknowledgments

We thank all authors and reviewers for their contributions to this Research Topic and for the support of the editorial office.

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

Bateman, C. J., Fourie, J., Hsiao, J., Irie, K., Heslop, A., Hilditch, A., et al. (2020). Assessment of mixed sward using context sensitive convolutional neural networks. *Front. Plant Sci.* 11, 159. doi: 10.3389/fpls.2020.00159

Chen, L. Y., Li, S. B., Bai, Q., Yang, J., Jiang, S. L., and Miao, Y. M. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sens.* 13 (22), 4712. doi: 10.3390/rs13224712

Furbank, R. T., and Tester, M. (2011). Phenomics–technologies to relieve the phenotyping bottleneck. *Trends Plant Sci.* 16 (12), 635–644. doi: 10.1016/j.tplants.2011.09.005

Ghamkhar, K., Irie, K., Hagedorn, M., Hsiao, J., Fourie, J., Gebbie, S., et al. (2019). Real-time, non-destructive and in-field foliage yield and growth rate measurement in perennial ryegrass (Lolium perenne l.). *Plant Methods* 15 (1), 1–12. doi: 10.1186/s13007-019-0456-2

Ghosh, S., Das, N., Das, I., and Maulik, U. (2019). Understanding deep learning techniques for image segmentation. *ACM Computing. Surveys.* 52(4), 1–35. doi: 10.1145/3329784

Hansen, M. A. E., Hay, F. R., and Carstensen, J. M. (2016). A virtual seed file: the use of multispectral image analysis in the management of genebank seed accessions. *Plant Genet. Resour.* 14 (3), 238–241. doi: 10.1017/S1479262115000362

Roitsch, T., Cabrera-Bosquet, L., Fournier, A., Ghamkhar, K., Jiménez-Berni, J., Pinto, F., et al. (2019). New sensors and data-driven approaches–a path to next generation phenomics. *Plant Sci.* 282, 2–10. doi: 10.1016/j.plantsci.2019.01.011

Ubbens, J. R., and Stavness, I. (2017). Deep plant phenomics: A deep learning platform for complex plant phenotyping tasks. *Front. Plant Sci.* 8. doi: 10.3389/fpls.2017.01190

Xiao, Y. Z., Tian, Z. Q., Yu, J. C., Zhang, Y. S., Liu, S., Du, S. Y., et al. (2020). A review of object detection based on deep learning. *Multimedia. Tools Appl.* 79 (33-34), 23729–23791. doi: 10.1007/s11042-020-08976-6

Yang, W., Feng, H., Zhang, X., Zhang, J., Doonan, J. H., Batchelor, W. D., et al. (2020). Crop phenomics and high-throughput phenotyping: Past decades, current challenges, and future perspectives. *Mol. Plant* 13 (2), 187–214. doi: 10.1016/j.molp.2020.01.008

frontiers in Plant Science

ORIGINAL RESEARCH
published: 02 August 2021
doi: 10.3389/fpls.2021.709008

Corn Yield Prediction With Ensemble CNN-DNN

Mohsen Shahhosseini[1], Guiping Hu[1], Saeed Khaki[1] and Sotirios V. Archontoulis[2]*

[1] *Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA, United States,*
[2] *Department of Agronomy, Iowa State University, Ames, IA, United States*

We investigate the predictive performance of two novel CNN-DNN machine learning ensemble models in predicting county-level corn yields across the US Corn Belt (12 states). The developed data set is a combination of management, environment, and historical corn yields from 1980 to 2019. Two scenarios for ensemble creation are considered: homogenous and heterogenous ensembles. In homogenous ensembles, the base CNN-DNN models are all the same, but they are generated with a bagging procedure to ensure they exhibit a certain level of diversity. Heterogenous ensembles are created from different base CNN-DNN models which share the same architecture but have different hyperparameters. Three types of ensemble creation methods were used to create several ensembles for either of the scenarios: Basic Ensemble Method (BEM), Generalized Ensemble Method (GEM), and stacked generalized ensembles. Results indicated that both designed ensemble types (heterogenous and homogenous) outperform the ensembles created from five individual ML models (linear regression, LASSO, random forest, XGBoost, and LightGBM). Furthermore, by introducing improvements over the heterogenous ensembles, the homogenous ensembles provide the most accurate yield predictions across US Corn Belt states. This model could make 2019 yield predictions with a root mean square error of 866 kg/ha, equivalent to 8.5% relative root mean square and could successfully explain about 77% of the spatio-temporal variation in the corn grain yields. The significant predictive power of this model can be leveraged for designing a reliable tool for corn yield prediction which will in turn assist agronomic decision makers.

Keywords: yield prediction, CNN-DNN, homogenous ensemble, heterogenous ensemble, US Corn Belt

OPEN ACCESS

Edited by:
Kioumars Ghamkhar,
AgResearch Ltd., New Zealand

Reviewed by:
Jingli Lu,
AgResearch Ltd., New Zealand
Monica Herrero Huerta,
Purdue University, United States

****Correspondence:***
Guiping Hu
gphu@iastate.edu

Specialty section:
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

Received: *13 May 2021*
Accepted: *05 July 2021*
Published: *02 August 2021*

Citation:
Shahhosseini M, Hu G, Khaki S and Archontoulis SV (2021) Corn Yield Prediction With Ensemble CNN-DNN. Front. Plant Sci. 12:709008. doi: 10.3389/fpls.2021.709008

INTRODUCTION

Accurate crop yield prediction is essential for agriculture production, as it can provide insightful information to farmers, agronomists, and other decision makers. However, this is not an easy task, as there is a myriad of variables that affect the crop yields, from genotypes, environment, and management decisions to technological advancements. The tools that are used to predict crop yields are mainly divided into simulation crop modeling and machine learning (ML).

Although these models are usually utilized separately, there have been some recent studies to combine them toward improving prediction. The outputs of crop models have served as inputs to multiple linear regression models in an attempt to make better crop yield predictions (Mavromatis, 2016; Busetto et al., 2017; Pagani et al., 2017). Some other studies have made additional advancement and created hybrid crop model-ML methodologies by using crop model outputs as inputs to a ML model (Everingham et al., 2016; Feng et al., 2019). In a recent study, Shahhosseini et al. (2021) designed a hybrid crop model-ML ensemble framework, in which a crop modeling framework (APSIM) was used to provide additional inputs to the yield prediction task (For more information about APSIM refer to https://www.apsim.info/). The results demonstrated that coupling APSIM and ML could improve ML performance up to 29% compared to ML alone.

On the other hand, the use of more complex machine learning models with the intention of better using numerous ecological variables to predict yields has been recently becoming more prevalent (Basso and Liu, 2019). Although there is always a tradeoff between the model complexity and its interpretability, the recent complex models could better capture all kinds of associations such as linear and nonlinear relationships between the variables associated with the crop yields, resulting in more accurate predictions and subsequently better helping decision makers (Chlingaryan et al., 2018). These models span from models as simple as linear regression, k-nearest neighbor, and regression trees (González Sánchez et al., 2014; Mupangwa et al., 2020), to more complex methods such as support vector machines (Stas et al., 2016), homogenous ensemble models (Vincenzi et al., 2011; Fukuda et al., 2013; Heremans et al., 2015; Jeong et al., 2016; Shahhosseini et al., 2019), heterogenous ensemble models (Cai et al., 2017; Shahhosseini et al., 2020, 2021), and deep neural networks (Liu et al., 2001; Drummond et al., 2003; Jiang et al., 2004, 2020; Pantazi et al., 2016; You et al., 2017; Crane-Droesch, 2018; Wang et al., 2018; Khaki and Wang, 2019; Kim et al., 2019; Yang et al., 2019; Khaki et al., 2020a,b). Homogeneous ensemble models are the models created using same-type base learners, while the base learners in the heterogenous ensemble models are different.

Although deep neural networks demonstrate better predictive performance compared to single layer networks, they are computationally more expensive, more likely to overfit, and may suffer from vanishing gradient problem. However, some studies have proposed solutions to address these problems and possibly boost deep neural network's performance (Bengio et al., 1994; Srivastava et al., 2014; Ioffe and Szegedy, 2015; Szegedy et al., 2015; Goodfellow et al., 2016; He et al., 2016).

Convolutional neural networks (CNNs) have mainly been developed to work with two-dimensional image data. However, they are also widely used with one-dimensional and three-dimensional data. Essentially, CNNs apply a filter to the input data which results in summarizing different features of the input data into a feature map. In other words, CNN paired with pooling operation can extract high-level features from the input data that includes the necessary information and has lower dimension. This means CNNs are easier to train and have fewer parameters compared to fully connected networks (Goodfellow et al., 2016; Zhu et al., 2018; Feng et al., 2020).

Since CNNs are able to preserve the spatial and temporal structure of the data, they have recently been used in ecological problems, such as yield prediction. Khaki et al. (2020b) proposed a hybrid CNN-RNN framework for crop yield prediction. Their framework consists of two one-dimensional CNNs for capturing linear and nonlinear effects of weather and soil data followed by a fully connected network to combine high-level weather and soil features, and a recursive neural network (RNN) that could capture time dependencies in the input data. The results showed that the model could achieve decent relative root mean square error of 9 and 8% when predicting corn and soybean yields, respectively. You et al. (2017) developed CNN and LSTM models for soybean yield prediction using remote sensor images data. The developed models could predict county-level soybean yields in the U.S. better than the competing approaches including ridge regression, decision trees, and deep neural network (DNN). Moreover, Yang et al. (2019) used low-altitude remotely sensed imagery to develop a CNN model. The experimental results revealed that the designed CNN outperformed the traditional vegetation index-based regression model for rice grain yield estimation, significantly.

Another set of developed models to capture complex relationships in the input raw data are ensemble models. It has been proved that combining well-diverse base machine learning estimators of any types, can result in a better-performing model which is called an ensemble model (Zhang and Ma, 2012). Due to their predictive ability, ensemble models have also been used recently by ecologists. Several heterogenous ensemble models including optimized weighted ensemble, average ensemble, and stacked generalized ensembles were created using five base learners, namely LASSO regression, linear regression, random forest, XGBoost, and LightGBM. The computational results showed that the ensemble models outperformed the base models in predicting corn yields. Cai et al. (2017) combined several ML estimators to form a stacked generalized ensemble. The back-testing numerical results demonstrate that their model's performance is comparable to the USDA forecasts.

Although these models have provided significant advances toward making better yield predictions, there is still a need to increase the predictive capacity of the existing models. This can be done by improving the data collections, and by the means of developing more advanced and forward-thinking models. The ensemble models are excellent tools that have the potential to turn very good models to outstanding predictor models.

Motivated by the high predictive performance of CNNs and ensemble models in ecology (Cai et al., 2017; You et al., 2017; Yang et al., 2019; Khaki et al., 2020b; Shahhosseini et al., 2020, 2021), we propose a set of ensemble models created from multiple hybrid CNN-DNN base learners for predicting county-level corn yields across US Corn Belt states. Building upon successful studies in the literature (Khaki et al., 2020b; Shahhosseini et al., 2020), we designed a base architecture consisting of two one-dimensional CNNs and one fully connected network (FC) as the first layer networks, and another fully connected network that combined the outputs of the first-layer networks and made final predictions, as the second-layer network. Afterwards, two scenarios are considered for base learner generation: heterogenous and homogenous ensemble creation. In the heterogenous scenario, the base learners are neural networks with the same described architecture, but with different hyperparameters. On the contrary, the homogenous ensembles are created with bagging the same architecture and forming diverse base learners. In each scenario, the generated base learners are combined by several methods including simple averaging, optimized weighted averaging, and stacked generalization.

MATERIALS AND METHODS

The designed ensemble framework uses a combination of historical yield and management data obtained from USDA NASS, historical weather and soil data as the data inputs. The details of the created data set and the developed model will be explained below.

Data Preparation

Data Sources

The main variables that affect corn yields are environment, genotype, and management. Although genotype information are not publicly available, other pieces of information including environment (soil and weather) and some of the management decisions data could be accessed publicly. To this end, we created a data set from environment and management variables that could be used to predict corn yields. This data includes county-level weather, soil, and management data considering 12 US Corn Belt states (Illinois, Indiana, Iowa, Kansas, Michigan, Minnesota, Missouri, Nebraska, North Dakota, Ohio, South Dakota, and Wisconsin). It is also noteworthy that since only some of the locations across US Corn Belt states are irrigated, to keep the consistency across the entire developed data set, we assumed that all farms are rainfed and didn't consider irrigation as a feature. The variables weekly planting progress per state and corn yields per county were downloaded from USDA National Agricultural Statistics Service (NASS, 2019). The weather was obtained from a reanalysis weather database based off of NASA Power[1] and Iowa Environmental Mesonet.[2] Finally, the soil data was created from SSURGO, a soil database based off of soil survey information collected by the National Cooperative Soil Survey (Soil Survey Staff [SSS], 2019). These variables are described below. Across 12 states, on average the data from 950 counties in total were used per year.

- *Planting progress (planting date):* 52 features explaining the weekly cumulative percentage of corn planted within each state. Each of these state-level weekly features represents the cumulative percentage of corn planted until that particular week (NASS, 2019).
- *Weather*: Five weather features accumulated weekly (52 × 5 = 260 features), obtained from NASA Power and Iowa Environmental Mesonet.
  - Daily minimum air temperature in degrees Celsius.
  - Daily maximum air temperature in degrees Celsius.
  - Daily total precipitation in millimeters per day.
  - Shortwave radiation in watts per square meter.
  - Growing degree days.
- *Soil*: The soil features wet soil bulk density, dry bulk density, clay percentage, plant available water content, lower limit of plant available water content, hydraulic conductivity, organic matter percentage, pH, sand percentage, and saturated volumetric water content. All variables determined at 10 soil profile depths (cm): 0–5, 5–10, 10–15, 15–30, 30–45, 45–60, 60–80, 80–100, 100–120, and 120–150 (Soil Survey Staff [SSS], 2019).
- *Corn Yield*: Yearly corn yield in bushel per acre, collected from USDA-NASS (2019).

Data Pre-processing

The following pre-processing tasks were performed on the created data set to make it prepared for training the designed ensemble models.

- Imputing missing planting progress data for the state North Dakota before the year 2000 by considering average progress values of two closest states (South Dakota and Minnesota).
- Removing out-of-season planting progress data before planting and after harvesting.
- Removing out-of-season weather features before planting and after harvesting.
- Aggregating weather features to construct quarterly and annually weather features. The features solar radiation and precipitation were aggregated by summation, while other weather features (minimum and maximum temperature) were aggregated by a row-wise average.
- The observations with the yield less than 10 bu/acre were considered as outliers and dropped from the data set.
- Investigating the historical corn yields over the time reveals an increasing trend in the yield values. This could be explained as the effect of technological advances, like genetic gains, management progress, advanced equipment, and other technological advances. Hence, a new input feature was constructed using the observed trends that enabled the models to account for the increasing yield trend.
  - *yield_trend*: this feature explained the observed trend in corn yields. A linear regression model using the training data was built for each location as the trends for each site tend to be different. The year ($YEAR$) and yield (Y) features served as the predictor and response variables of this linear regression model, respectively. Then the predicted value for each data point ($\hat{Y}$) is added as a new input variable that explains the increasing annual trend in the target variable. The corresponding value for the observations in the test data set was estimated by plugging in their corresponding year in the trained linear regression models ($\hat{Y}_{i,test} \;\; b_{0i} + b_{1i}YEAR_{i,test}$). The following equation shows the trend value ($\hat{Y}_i$) calculated for each location (i), that is added to the data set as a new feature.

$$\hat{Y}_i \;\; b_{0i} + b_{1i}YEAR_i \tag{1}$$

- All independent variables were scaled to be ranged between 0 and 1.

[1] https://power.larc.nasa.gov

[2] https://mesonet.agron.iastate.edu

Base Models Generation

We propose the following CNN-DNN architecture as the foundation for generating multiple base learners that serve as the inputs to the ensemble creation models. The architecture consists of two layers of deep neural networks.

First Layer

Due to the ability of CNNs in capturing the spatial and temporal dependencies that exist in the soil and weather data, respectively, we decided to build two separate set of one-dimensional CNNs for each of the weather (W-CNN) and soil (S-CNN) groups of features. Such networks have been used before in different studies and have been proved to be effective in capturing linear and nonlinear effects in the soil and weather (Ince et al., 2016; Borovykh et al., 2017; Kiranyaz et al., 2019). In addition, a fully connected network (FC1) was built that took planting progress, and other constructed features as inputs and the output is concatenated with the outputs of the CNN components to serve as inputs of the second layer of the networks.

Specifically, the first layer includes three network types:

1 Weather CNN models (W-CNN):
CNN is able to capture the temporal effect of weather data measured over time. In the case of the developed data set, we will use a set of one-dimensional CNNs inside the W-CNN component.
2 Soil CNN models (S-CNN):
CNN can also capture the spatial effect of soil data which is measured over time and on different depths. Considering the data set, we will use a set of one-dimensional CNNs to build this component of the network.
3 Other variables FC model (FC1):
This fully connected network can capture the linear and nonlinear effect of other input features.

Second Layer (FC2)

In the second layer we used a fully connected network (FC2) that aggregates all extracted features of the first layer networks (W-CNN, S-CNN, and FC1), and makes the final yield prediction.

The architecture of the proposed base network is depicted in **Figure 1**. As it is shown in the figure, the W-CNN and S-CNN components of the network each are comprised of a set of CNNs that are in charge of one data input type and their outputs are aggregated with a fully connected network. For the case of W-CNN component, there are five CNNs for each weather data type (precipitation, maximum temperature, minimum temperature, solar radiation, and growing degree days). Similarly, 10 internal CNNs are designed inside S-CNN component for each of the 10 soil data types. The reason we decided to design one CNN for each data type is the differences in the natures of different data types and our experiments showed that separate CNNs for each data type could extract more useful information and will result in better final predictions. The two inner fully connected networks (FC_W and FC_S) both have one hidden layer with 60 and 40 neurons, respectively.

We used VGG-like architecture for the CNN models (Simonyan and Zisserman, 2014). The details about each of the designed CNN networks are presented in **Table 1**. We performed downsampling in the CNN models by average pooling with a stride of size 2. The feed-forward fully connected network in the first layer (FC1) has three hidden layers with 64, 32, and 16 neurons. The final fully connected network of the second layer (FC2) is grown with two hidden layers with 128 and 64 neurons. In addition, two dropout layers with dropout ratio of 0.5 are located at the two last layers of the FC2 to prevent the model from overfitting. We used Adam optimizer with the learning rate of 0.0001 for the entire model training stage and trained the model for 1,000 iterations considering batches of size 16. Rectified linear unit (ReLU) was used as the activation function of all networks throughout the architecture except the output layer that had a linear activation function.

To ensure that the ensemble created from a set of base learners performs better than them, the base learners should have a certain level of diversity and prediction accuracy (Brown, 2017). Hence, two scenarios for generating diverse base models are considered which are systematically different: homogenous and heterogenous ensemble base model generation.

Homogenous Ensembles

The homogenous ensembles are the models whose base learners are all the same type. Random forest and gradient boosting are examples of homogenous ensemble models. Their base learners are decision trees with the same hyperparameter values. Bootstrap aggregating (Bagging) is an ensemble framework which was proposed by Breiman (1996). Bagging generates multiple training data sets from the original data set by sampling with replacement (bootstrapping). Then, one base model is trained on each of the generated training data sets and the final prediction is the average (for regression problems) or voting (for classification problems) of the predictions made by each of those base models. Basically, by sampling with replacement and generating multiple data sets, and subsequently multiple base models, bagging ensures the base models have a certain level of diversity. In other words, bagging tries to reduce the prediction variance by averaging the predictions of multiple diverse base models.

Here, inspired by the way bagging introduces diversity in the base model generation, we design a bagging schema which generates multiple base CNN-DNN models using the same foundation model (**Figure 1**). This is shown in **Figure 2**. Then several ensemble creation methods make use of these bagged networks as the base models to create a better-performing ensemble network. We believe one drawback of bagging is assigning equal weights to the bagged models. To address that, we will use different ensemble creation methods in order to optimally combine the bagged models. We will discuss ensemble creation in the next chapter.

Heterogenous Ensembles

On the other hand, the base models in the heterogenous ensembles are not the same. They can be any machine learning model from the simplest to the most complex models. However, as mentioned before, the ensemble is not expected to perform favorably if the base models do not exhibit a

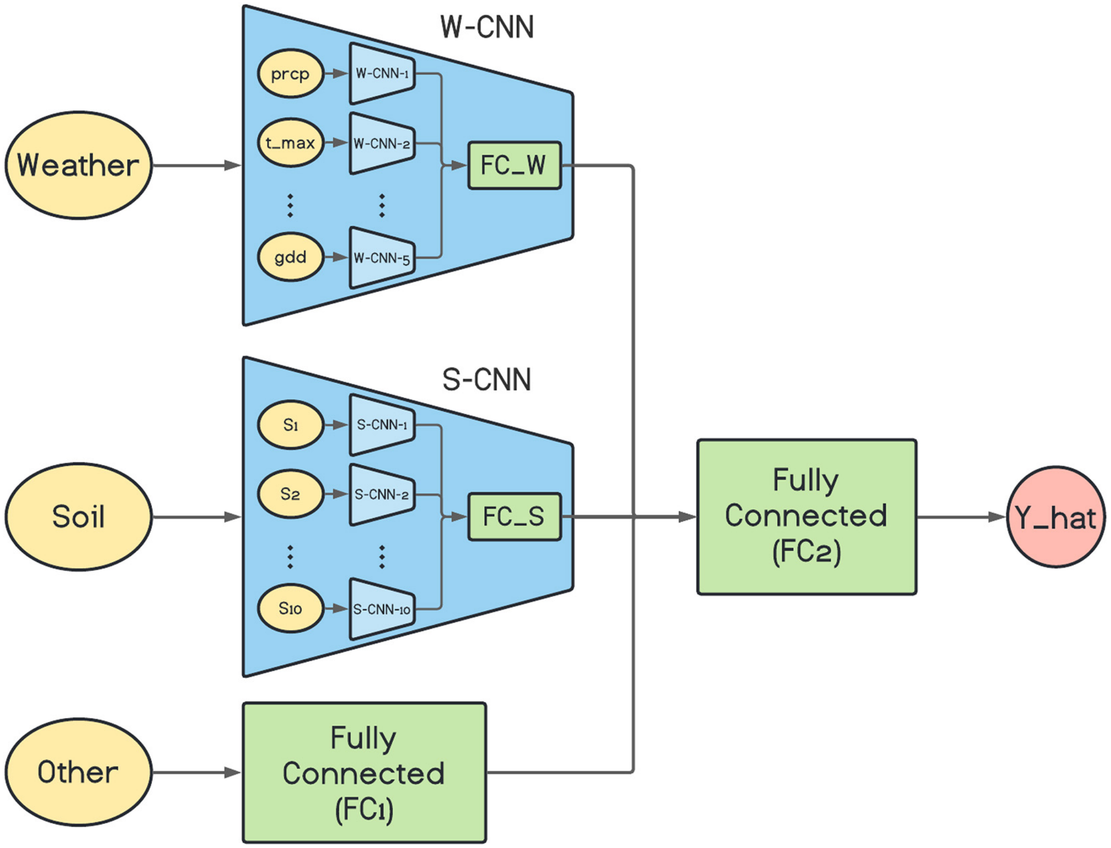


FIGURE 1 | The architecture of the proposed base network. prcp, t_max, and gdd represent precipitation, maximum temperature, and growing degree days, respectively. S1, S2, . . ., and S10 are 10 soil variables which each are measured at 10 depth levels. Y_hat represents the final corn yield prediction made by the model.

TABLE 1 | Detailed structure of the CNN networks of CNN components designed as the foundation for ensemble neural networks.

CNNs in the W-CNN component					**CNNs in the S-CNN component**				
Input size		**32 × 1**			**Input size**		**10 × 1**		
Layer name	**FS**	**NF**	**S**	**P**	**Layer name**	**FS**	**NF**	**S**	**P**
Conv1	6	4	1	Valid	Conv1	3	4	1	Valid
Average pooling 1	2	–	2	Valid	Average pooling 1	2	–	2	Valid
Conv2	3	4	1	Valid	Conv2	3	4	1	Valid
Average pooling 2	2	–	2	Valid	Average pooling 2	2	–	2	Valid
Conv3	3	4	1	Valid	Conv3	3	4	1	Valid
Average pooling 3	2	–	2	Valid	Output size		4 × 1		
Output size		4 × 1							

The table on the left shows the details of the CNNs designed for each weather feature, and the right table presents the ones for the CNNs designed for each soil feature. FS, NF, S, and P represent filter size, number of features, stride, and padding.

certain level of diversity. To that end, we train *k* variations of the base CNN-DNN model presented earlier. The foundation architecture of these *k* models are the same, but their CNN hyperparameters are different. In other words, we preserve the same architecture for all models and change the number of filters inside each CNN network to create various CNN-DNN models. These models will serve as the inputs to the ensemble creation methods explained in the next chapter (see **Figure 3**).

Ensemble Creation

After generating base learners in either of the heterogenous and homogenous methods, they should be combined using a systematic procedure. We have used three different types of

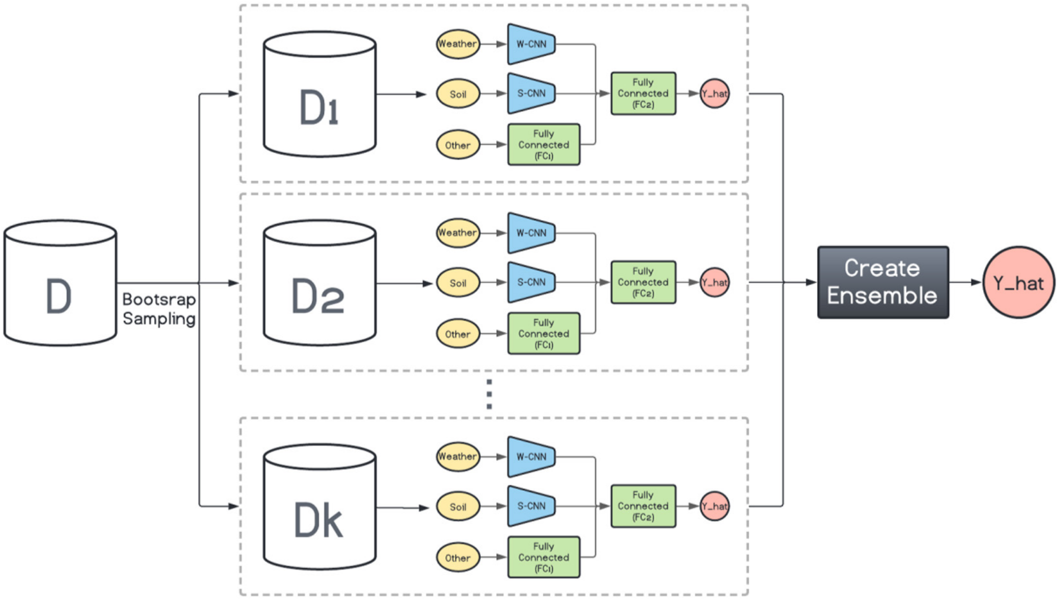


FIGURE 2 | Homogenous ensemble creation with bagging architecture. *k* data sets (D1, D2, . . ., Dk) were generated with bootstrap sampling from the original data set (D) and the same base network is trained on each of them. The ensemble creation combines the predictions made by the base networks.

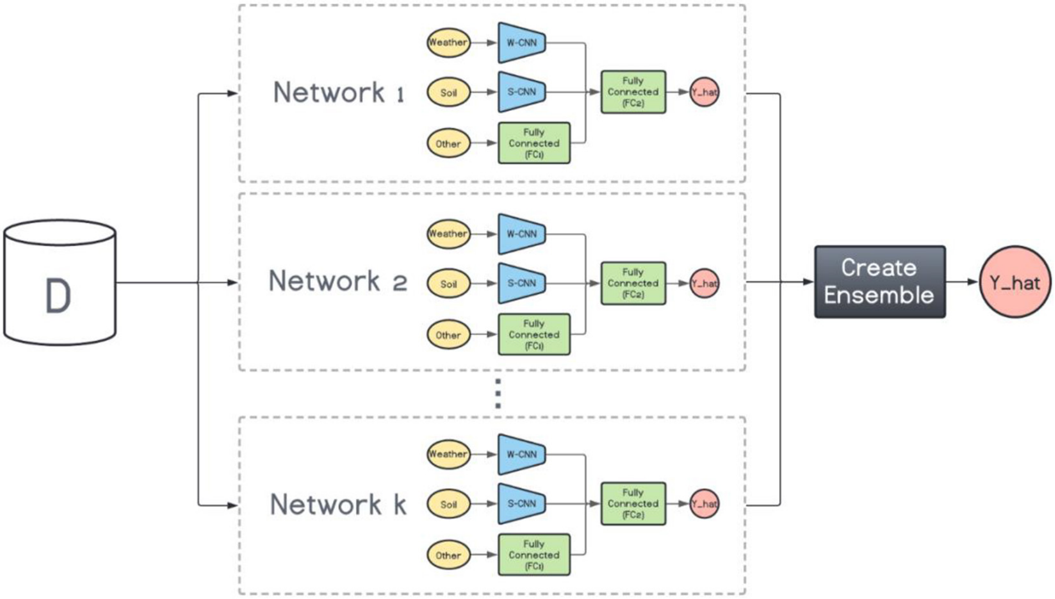


FIGURE 3 | Heterogenous ensemble creation. *k* networks with the same architecture but with different hyperparameters are created using the original data set (D).

ensemble creation methods which are Basic Ensemble Method (BEM), Generalized Ensemble Method (GEM), and stacked generalized ensemble method.

Basic Ensemble Method (BEM)

Perrone and Cooper (1992) proposed BEM as the most natural way of combining base learners. BEM creates a regression ensemble by simple averaging the base estimators. This study claims that BEM can reduce mean squared error of predictions, given that the base learners are diverse.

Generalized Ensemble Method (GEM)

GEM is the general case of a BEM ensemble creation method and tries to create a regression ensemble as the linear combination of the base estimators. Cross-validation is used to generate out-of-bag (OOB) predictions and optimize the ensemble weights and the model was claimed to avoid overfitting the data (Perrone and Cooper, 1992).

The nonlinear convex optimization problem is as follows.

$$Min\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{k} w_j\hat{y}_{ij}\right)^2 \quad (2)$$
$$s.t.$$

$\sum_{j=1}^{k} w_j = 1,\ w_j \geq 0, \forall j = 1\ldots,k$. In which w_j is the weight of base model j ($j = 1\ldots,k$), n is the total number of observations, y_i is the true value of observation i, and $\hat{y}_{ij}$ is the prediction of observation i by base model j.

Stacked Generalized Ensemble Method

Stacked generalization is referred to combining several base estimators by performing at least one more level of machine learning task. Usually, cross-validation is used to generate OOB predictions form the training samples and learn the higher-level machine learning models (Wolpert, 1992). The second level learner can be any choice of ML models. In this study we have selected linear regression, LASSO, random forest and LightGBM as the second level learners.

RESULTS

The historical county-level data of the US Corn Belt states (Illinois, Indiana, Iowa, Kansas, Michigan, Minnesota, Missouri, Nebraska, North Dakota, Ohio, South Dakota, and Wisconsin) spanning across years 1980–2019 were used to train all considered models. The data from the years 2017, 2018, and 2019, in turn, were reserved as the test data and the data from the years before each of them formed the training data.

TABLE 2 | Detailed structure of the CNN networks of CNN components designed for heterogenous ensemble models.

CNNs in the W-CNN component of model 1

Input size	32 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	6	2	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	2	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	2	1	Valid
Average pooling 3	2	–	2	Valid
Output size	2 × 1			

CNNs in the S-CNN component of model 1

Input size	10 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	3	2	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	2	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	2	1	Valid
Output size	2 × 1			

CNNs in the W-CNN component of model 2

Input size	32 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	6	3	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	3	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	3	1	Valid
Average pooling 3	2	–	2	Valid
Output size	3 × 1			

CNNs in the S-CNN component of model 2

Input size	10 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	3	3	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	3	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	3	1	Valid
Output size	3 × 1			

CNNs in the W-CNN component of model 3

Input size	32 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	6	4	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	4	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	4	1	Valid
Average pooling 3	2	–	2	Valid
Output size	4 × 1			

CNNs in the S-CNN component of model 3

Input size	10 × 1			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	3	4	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	4	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	4	1	Valid
Output size	4 × 1			

(Continued)

TABLE 2 | Continued

CNNs in the W-CNN component of model 4				
Input size	**32 × 1**			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	6	5	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	5	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	5	1	Valid
Average pooling 3	2	–	2	Valid
Output size		5 × 1		

CNNs in the S-CNN component of model 4				
Input size	**10 × 1**			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	3	5	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	5	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	5	1	Valid
Output size		5 × 1		

CNNs in the W-CNN component of model 5				
Input size	**32 × 1**			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	6	6	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	6	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	6	1	Valid
Average pooling 3	2	–	2	Valid
Output size		6 × 1		

CNNs in the S-CNN component of model 5				
Input size	**10 × 1**			
Layer name	**FS**	**NF**	**S**	**P**
Conv1	3	6	1	Valid
Average pooling 1	2	–	2	Valid
Conv2	3	6	1	Valid
Average pooling 2	2	–	2	Valid
Conv3	3	6	1	Valid
Output size		6 × 1		

The tables on the left show the details of the CNNs designed for each weather feature, and the right tables present the ones for the CNNs designed for each soil feature. FS, NF, S, and P represent filter size, number of features, stride, and padding.

TABLE 3 | Test prediction error (RMSE) and coefficient of determination (R^2) of designed ensemble models compared to the benchmark ensembles (Shahhosseini et al., 2020, 2021).

ML models	**BEM**		**GEM**		**Stacked regression**		**Stacked LASSO**		**Stacked random forest**		**Stacked LightGBM**	
	RMSE (kg/ha)	**R^2 (%)**	**RMSE (kg/ha)**	**R^2 (%)**	**RMSE (kg/ha)**	**R^2 (%)**	**RMSE (kg/ha)**	**R^2 (%)**	**RMSE (kg/ha)**	**R^2 (%)**	**RMSE (kg/ha)**	**R^2 (%)**
Test year: 2017—Training years: 1980–2016												
Benchmark	960	79.6	1,002	77.7	1,014	77.2	1,012	77.3	1,024	76.7	999	77.9
Heterogenous	1,003	77.7	969	79.2	908	81.8	908	81.7	978	78.8	933	80.7
Homogenous	954	79.8	944	80.3	875	83.0	874	83.1	936	80.6	906	81.8
Test year: 2018—Training years: 1980–2017												
Benchmark	1,145	74.7	1,047	78.8	1,041	79.0	1,041	79.0	1,101	76.6	1,070	77.9
Heterogenous	1,065	78.0	1,094	76.8	1,072	77.8	1,072	77.8	1,116	75.9	1,087	77.2
Homogenous	1,033	79.4	992	81.0	1,058	78.4	1,056	78.4	1,077	77.6	1,065	78.1
Test year: 2019—Training years: 1980–2018												
Benchmark	936	72.6	1,035	66.4	1,028	66.9	1,035	66.5	1,084	63.2	1,029	66.9
Heterogenous	900	74.6	1,083	63.3	1,282	48.5	1,279	48.8	1,225	53.0	1,234	52.3
Homogenous	866	76.5	867	76.5	885	75.5	883	75.6	932	72.8	895	74.9

As mentioned in the section "Ensemble Creation," the ensemble creation methods require OOB predictions from all the input models that represent the test data to optimally combine the base models. The current procedure to create these OOB predictions is using a cross-validation method. However, due to time-dependency in the training data and the fact that in the homogenous ensemble models the training data is resampled k times, it is not possible to find a consistent vector of OOB predictions across all models and use it to combine the base models. Therefore, 20% of the training data was considered as the validation data and was not used in model training. It is noteworthy that the training data is split to 20–80% with a stratified split procedure to ensure the validation data has a similar distribution with the training data.

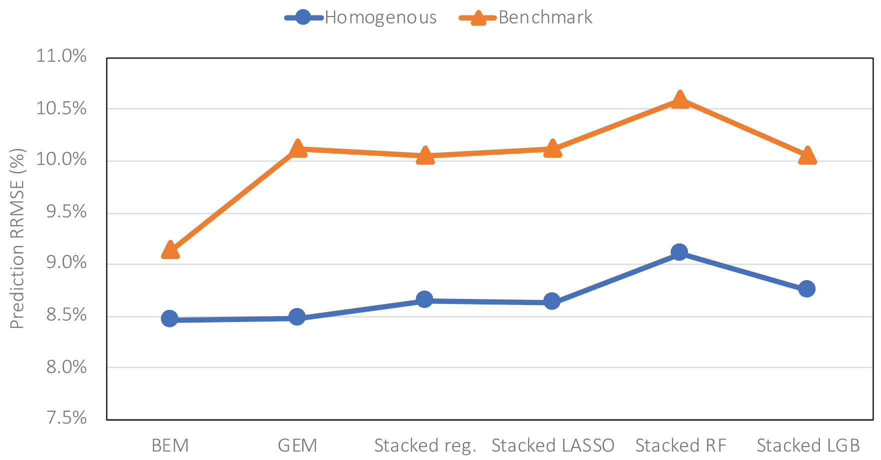


FIGURE 4 | Comparing prediction error (relative RMSE) of the homogeneous model with the benchmark on the data from the year 2019 taken as the test data.

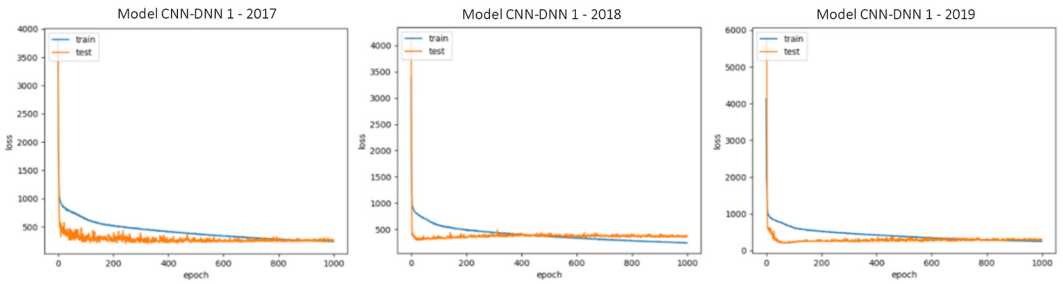


FIGURE 5 | Train and test loss vs. epochs of some of the trained CNN-DNN models. Similar observations were made for all trained models and only some of them are shown for illustration purposes. The shown examples are representative of all the examples.

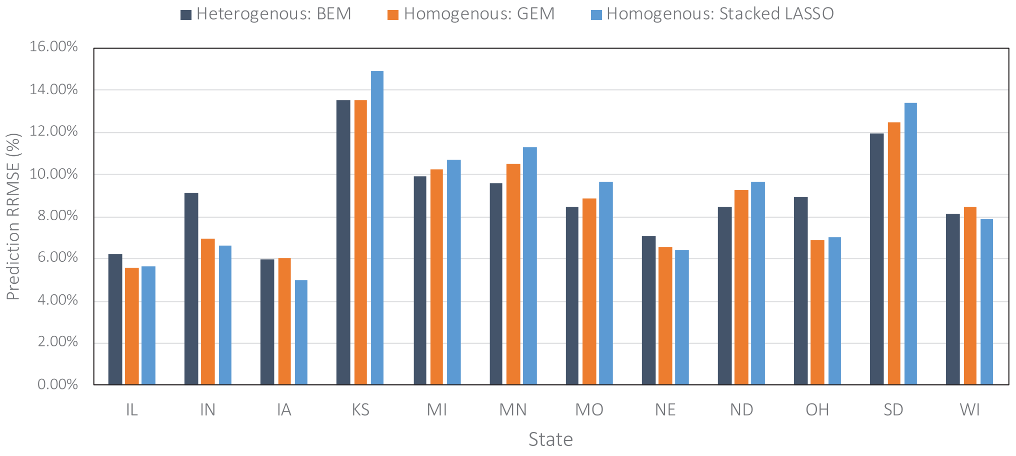


FIGURE 6 | Comparing prediction error (relative RMSE) of some of the designed ensembles across all US Corn Belt states on the data from the year 2019 taken as the test data.

To achieve the stratified splits, we binned the observations in the training data into five linearly spaced bins based on their corresponding yield values.

The CNN structure of the base models trained for creating homogenous ensemble models are same as the one shown in **Table 1**. We have resampled the training data 10 times (with replacement) and trained the same CNN-DNN model on each of the 10 newly created training data. The OOB predictions are the predictions made by each of the 10 mentioned models on the validation data.

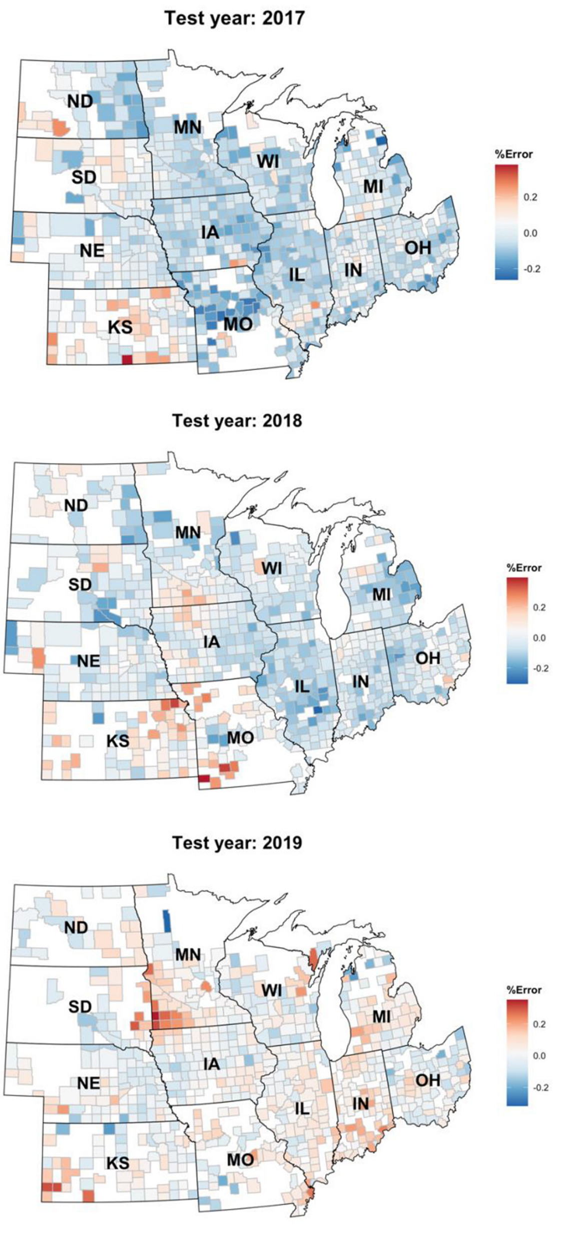


FIGURE 7 | Relative percentage error of the Homogenous GEM predictions shown on a choropleth map of the US Corn Belt.

On the other hand, the base models trained for creating heterogenous ensemble models are not the same and they differ in their CNN hyperparameters (number of filters). We trained five different CNN-DNN base models on the same training data and formed the OOB predictions by each of those five models predicting the observations in the validation data. The details of the CNN components in these five models are shown in the **Table 2**.

To evaluate the performance of the trained heterogenous and homogenous CNN-DNN ensembles, the ensembles created from five individual machine learning models (linear regression, LASSO, XGBoost, random forest, and LightGBM) were considered as benchmark and were trained on the same data sets developed for training the CNN-DNN ensemble models. The benchmark models were run on a computer equipped with a 2.6 GHz Intel E5-2640 v3 CPU, and 128 GB of RAM. The CNN-DNN models were run on a computer with a 2.3 GHz Intel E5-2650 v3 CPU, NVIDIA k20c GPU, and 768 GB of RAM.

The predictive performance of these ensemble models was previously shown in two separate published papers (Shahhosseini et al., 2020, 2021). The results are summarized in the **Table 3** (see **Supplementary Figure 1** for XY plots of some of the designed ensembles).

The heterogenous and homogenous ensemble models both provide improvements over the well-performing ensemble benchmarks in most cases (**Table 3**). However, the heterogenous ensemble model is constantly outperformed by the homogeneous ensemble models. This is in line with what we expected as the homogeneous model inherently introduces more diversity in the ensemble base models which in turn will result in lowering the prediction variance and consequently better generalizability of the trained model. The performance comparison of homogeneous ensemble model compared to the benchmark is shown in the **Figure 4**. Another observation in the **Table 3** is that in case of homogenous ensembles, some of the ensemble creation methods have made better predictions than average homogeneous ensemble (BEM) i.e., bagged CNN-DNN. This again confirms our assertion that assigning unequal weights to the bagged models results in better predictions.

The generalizability of all trained models is proved as we have shown that in three test scenarios, the ensemble models demonstrate superb prediction performance. This also can be observed by looking at the train and test loss vs. epochs graphs.

TABLE 4 | Test prediction error (RMSE) and coefficient of determination (R^2) of designed ensemble models compared to the benchmark ensembles (Shahhosseini et al., 2020, 2021) when applied on 2020 test data.

ML models	BEM		GEM		Stacked regression		Stacked LASSO		Stacked random forest		Stacked LightGBM	
	RMSE (kg/ha)	R^2 (%)	RMSE (kg/ha)	R^2 (%)	RMSE (kg/ha)	R^2 (%)	RMSE (kg/ha)	R^2 (%)	RMSE (kg/ha)	R^2 (%)	RMSE (kg/ha)	R^2 (%)
Test year: 2020—Training years: 1980–2018												
Benchmark	1,115	68.4	1,165	65.5	1,166	65.4	1,170	65.2	1,210	62.8	1,183	64.4
Heterogenous	972	76.0	989	75.1	992	75.0	991	75.0	1,048	72.1	1,000	74.6
Homogenous	982	75.5	958	76.7	1,001	74.5	999	74.6	1,053	71.8	1,018	73.6

Some examples of these graphs are shown in **Figure 5**. As the figure suggests, the dropout layers could successfully prevent overfitting of the CNN-DNN models, and the test errors tend to stay stable across the iterations. The generalizability of the trained models will further be discussed in the chapter 4.

DISCUSSION

Models' Performance Comparison With the Literature

We designed a novel CNN-DNN ensemble model with the objective of providing the most accurate prediction model for county-level corn yield across US Corn Belt states. The numerical results confirmed the superb performance of the designed ensemble models compared to literature models. **Table 3** showed that the homogenous ensemble models outperform the benchmark (Shahhosseini et al., 2020) by 10–16%. In addition, comparing the results with another well-performing prediction model in the literature (Khaki et al., 2020b), the homogeneous ensemble could outperform the prediction results of Khaki et al. (2020b) by 10–12% in common test set scenarios (2017 and 2018 test years). The CNN-RNN model developed by Khaki et al. (2020b) presented test prediction errors of 988 kg/ha (15.74 bu/acre) and 1,107 kg/ha (17.64 bu/acre) for the test years 2017 and 2018, respectively, while the homogeneous ensemble model designed here resulted in test prediction errors of 874 kg/ha (13.93 bu/acre) and 992 kg/ha (15.8 bu/acre) for the test years 2017 and 2018, respectively.

This is the first study that designed a novel ensemble neural network architecture that has the potential to make the most accurate yield predictions. The model developed here is advantageous compared to the literature due to the ability of the ensemble model in decreasing prediction variance by combining diverse models as well as reducing prediction bias by training the ensemble model based on powerful base models. Shahhosseini et al. (2020) had used ensemble learning for predicting county-level yield prediction, but neural network-based architectures were not considered, and the models were trained only on three states (IL, IA, IN). Khaki et al. (2020b) trained a CNN-RNN model for predicting US Corn Belt corn and soybean yields, but the model developed there is unable to make predictions as accurate as the models designed in this study and is not benefitting from the diversity in the predictions.

Including remote sensing data as well as simulated data from crop model like APSIM could potentially improve the predictions made by our models further which can be pursued as the future research direction. In addition, we assumed all considered farms are rainfed, while in states such as Kansas and Nebraska many of the farms are irrigated. Surprisingly, the prediction accuracy in these states was comparable with other states (**Figures 6**, **7**). We believe this is because of the use of average or rainfed corn yields from these states, not irrigated yields to train our models. Including the irrigation data can result in better prediction and perhaps new models for those states and is another possible future research direction.

Comparing the Models' Performance Across US Corn Belt States

Figure 6 compares the prediction errors of the test year of 2019 for some of the designed ensemble models represented by relative root mean squared error (RRMSE) for each of the 12 US Corn Belt states under study. The models performed the best in Iowa, Illinois, and Nebraska, and worst in Kansas and South Dakota. The worse prediction error in Kansas can be explained by the fact that the majority of the farms in Kansas state are irrigated and this irrigation is not considered as one of the variables when training the ensemble models. It is clear that including irrigation variable can improve the predictions. However, that was not the case for Nebraska, suggesting that irrigation may not be the only reason for the low performance in Kansas. Upon further investigate, we realized the corn yields in the Nebraska state are highly correlated with the weather features especially maximum temperature, while the corn yields in the Kansas state don't show this amount of correlation to weather features and are slightly correlated with both weather and soil features. In other words, it seems that although the weather features are adequate for making decent predictions in the Nebraska state, this is not the case for the Kansas.

Figure 7 depicts the relative error percentage of each year's test predictions on a county choropleth map of the US Corn Belt. The errors are calculated by dividing over/under prediction of the homogenous GEM model divided by the yearly average yield. This figure proves that the model is robust and can be easily generalized to other environments/years. One observation is that the model keeps overpredicting the yields in the Kansas state. This could be explained by the irrigation assumption we made when developing the data set. We assumed all the farms are rainfed and did not consider irrigation in states like Kansas in which some of the farms are irrigated.

Generalization Power of the Designed Ensemble CNN-DNN Models

To further test the generalization power of the designed ensembles, we gathered the data of all considered US Corn Belt states for the year 2020 and applied the trained heterogeneous and homogeneous ensemble models as well as the benchmarks on the new unseen observations of the year 2020. As the results imply (**Table 4**), both heterogenous and homogeneous ensemble models provide better predictions than the benchmark ensemble models, with the homogeneous Generalized Ensemble Model (GEM) being the most accurate prediction model. This model could provide predictions with 958 kg/ha root mean squared error and explain about 77% of the total variability in the response variable.

CONCLUSION

In this study we designed two novel CNN-DNN ensemble types for predicting county-level corn yields across US Corn Belt states. The base architecture used for creating the ensembles is a combination of CNNs and deep neural networks. The CNNs were in charge of extracting useful high-level features from the soil and weather data and provide them to a fully connected network for making the final yield predictions. The two ensemble types were heterogeneous and homogeneous which used the same base CNN-DNN structure but generated the base models in different manners. The homogenous ensemble used one fixed CNN-DNN network but applied it on multiple bagged data sets. The bagged data sets introduced a certain level of diversity that the created ensembles had benefited from. On the other hand, the heterogeneous ensemble used different base CNN-DNN networks which shared the same structure but differed in their number of filters. The different numbers of filters were considered as another method of introducing diversity into the ensembles. All base models generated from either of these two ensemble types were combined with each other using three ensemble creation methods: BEM, GEM, and stacked generalized ensembles. The numerical results showed that the ensemble models of both homogeneous and heterogeneous types could outperform the benchmark ensembles which had previously proved to be effective (Shahhosseini et al., 2020, 2021) as well as well-performing CNN-RNN architecture designed by Khaki et al. (2020b). In addition, homogeneous ensembles provide the most accurate predictions across all US Corn Belt states. The results demonstrated that in addition to the fact that these ensemble models benefitted from higher level of diversity from the bagged data sets, they provided a better combination of base models compared to simple averaging in the bagging. The generalization power of the designed ensembles was proved by applying them on the unseen observations of the year 2020. Once again heterogeneous and homogeneous ensemble models outperformed the benchmark ensembles.

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

MS led the research and wrote the manuscript. GH oversaw the research and edited the manuscript. SK contributed to the research idea and data processing. SA provided the data and edited the manuscript. All authors contributed to the article and approved the submitted version.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.709008/full#supplementary-material

REFERENCES

Basso, B., and Liu, L. (2019). "Chapter Four - Seasonal crop yield forecast: Methods, applications, and accuracies," in *Advances in Agronomy*, ed. D. L. Sparks (Cambridge, Massachusetts: Academic Press), 154, 201–255. doi: 10.1016/bs.agron.2018.11.002

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* 5, 157–166. doi: 10.1109/72.279181

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv [Preprint]*. Available online at: arXiv:1703.04691 (accessed April, 2021).

Breiman, L. (1996). Bagging predictors. *Mach. Learn.* 24, 123–140. doi: 10.1007/bf00058655

Brown, G. (2017). "Ensemble Learning," in *Encyclopedia of Machine Learning and Data Mining*, eds C. Sammut and G. I. Webb (Boston, MA: Springer US), 393–402.

Busetto, L., Casteleyn, S., Granell, C., Pepe, M., Barbieri, M., Campos-Taberner, M., et al. (2017). Downstream Services for Rice Crop Monitoring in Europe: from Regional to Local Scale. [Article]. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 10, 5423–5441.

Cai, Y., Moore, K., Pellegrini, A., Elhaddad, A., Lessel, J., Townsend, C., et al. (2017). *Crop yield predictions-high resolution statistical model for intra-season forecasts applied to corn in the US.* Paper presented at the 2017 Fall Meeting. United States: Gro Intelligence Inc.

Chlingaryan, A., Sukkarieh, S., and Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: a review. *Comput. Electron. Agric.* 151, 61–69. doi: 10.1016/j.compag.2018.05.012

Crane-Droesch, A. (2018). Machine learning methods for crop yield prediction and climate change impact assessment in agriculture. *Environ. Res. Lett.* 13:114003. doi: 10.1088/1748-9326/aae159

Drummond, S. T., Sudduth, K. A., Joshi, A., Birrell, S. J., and Kitchen, N. R. (2003). STATISTICAL AND NEURAL METHODS FOR SITE–SPECIFIC YIELD PREDICTION. *Trans. ASAE* 46, 5–14.

Everingham, Y., Sexton, J., Skocaj, D., and Inman-Bamber, G. (2016). Accurate prediction of sugarcane yield using a random forest algorithm. *Agron. Sustain. Dev.* 36:27.

Feng, P., Wang, B., Liu, D. L., Waters, C., and Yu, Q. (2019). Incorporating machine learning with biophysical model can improve the evaluation of climate extremes impacts on wheat yield in south-eastern Australia. *Agric. For. Meteorol.* 275, 100–113. doi: 10.1016/j.agrformet.2019.05.018

Feng, S.-H., Xu, J.-Y., and Shen, H.-B. (2020). "Chapter Seven - Artificial intelligence in bioinformatics: automated methodology development for protein residue contact map prediction," in *Biomedical Information Technology (Second Edition)*, ed. D. D. Feng (Cambridge, Massachusetts: Academic Press), 217–237.

Fukuda, S., Spreer, W., Yasunaga, E., Yuge, K., Sardsud, V., and Müller, J. (2013). Random Forests modelling for the estimation of mango (Mangifera indica L. cv. Chok Anan) fruit yields under different irrigation regimes. *Agric. Water Manage.* 116, 142–150. doi: 10.1016/j.agwat.2012.07.003

González Sánchez, A., Frausto Solís, J., and Ojeda Bustamante, W. (2014). Predictive ability of machine learning methods for massive crop yield prediction. *Span. J. Agric. Res.* 12, 313–328. doi: 10.5424/sjar/2014122-4439

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, (Las Vegas, NV, USA: IEEE).

Heremans, S., Dong, Q., Zhang, B., Bydekerke, L., and Orshoven, J. V. (2015). Potential of ensemble tree methods for early-season prediction of winter wheat yield from short time series of remotely sensed normalized difference vegetation index and in situ meteorological data. *J. Appl. Remote Sens.* 9:097095. doi: 10.1117/1.jrs.9.097095

Ince, T., Kiranyaz, S., Eren, L., Askar, M., and Gabbouj, M. (2016). Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. *IEEE Trans. Industr. Electron.* 63, 7067–7075. doi: 10.1109/tie.2016.2582729

Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv* Available Online at: http://arxiv.org/abs/1502.03167. (accessed April, 2021).

Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., et al. (2016). Random forests for global and regional crop yield predictions. *PLoS One* 11:e0156571. doi: 10.1371/journal.pone.0156571

Jiang, D., Yang, X., Clinton, N., and Wang, N. (2004). An artificial neural network model for estimating crop yields using remotely sensed information. *Int. J. Remote Sens.* 25, 1723–1732. doi: 10.1080/0143116031000150068

Jiang, H., Hu, H., Zhong, R., Xu, J., Xu, J., Huang, J., et al. (2020). A deep learning approach to conflating heterogeneous geospatial data for corn yield estimation: a case study of the US Corn Belt at the county level. *Glob. Chang. Biol.* 26, 1754–1766. doi: 10.1111/gcb.14885

Khaki, S., Khalilzadeh, Z., and Wang, L. (2020a). Predicting yield performance of parents in plant breeding: a neural collaborative filtering approach. *PLoS One* 15:e0233382. doi: 10.1371/journal.pone.0233382

Khaki, S., Wang, L., and Archontoulis, S. V. (2020b). A CNN-RNN Framework for Crop Yield Prediction. *Front. Plant Sci.* 10:1750. doi: 10.3389/fpls.2019.01750

Khaki, S., and Wang, L. (2019). Crop Yield Prediction Using Deep Neural Networks. *Front. Plant Sci.* 10:621. doi: 10.3389/fpls.2019.00621

Kim, N., Ha, K.-J., Park, N.-W., Cho, J., Hong, S., and Lee, Y.-W. (2019). A Comparison Between Major Artificial Intelligence Models for Crop Yield Prediction: case Study of the Midwestern United States, 2006–2015. *ISPRS Int. J. Geo Inform.* 8:240. doi: 10.3390/ijgi8050240

Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., and Gabbouj, M. (2019). "1-D Convolutional Neural Networks for Signal Processing Applications," in *Paper presented at the ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, UK: IEEE).

Liu, J., Goering, C., and Tian, L. (2001). A neural network for setting target corn yields. *Trans. ASAE* 44, 705–713.

Mavromatis, T. (2016). Spatial resolution effects on crop yield forecasts: an application to rainfed wheat yield in north Greece with CERES-Wheat. *Agric. Syst.* 143, 38–48. doi: 10.1016/j.agsy.2015.12.002

Mupangwa, W., Chipindu, L., Nyagumbo, I., Mkuhlani, S., and Sisito, G. (2020). Evaluating machine learning algorithms for predicting maize yield under conservation agriculture in Eastern and Southern Africa. *SN Appl. Sci.* 2:952.

NASS, U. (2019). *Surveys. National Agricultural Statistics Service*. Washington, D.C., United States: U.S. Department of Agriculture.

Pagani, V., Stella, T., Guarneri, T., Finotto, G., van den Berg, M., Marin, F. R., et al. (2017). Forecasting sugarcane yields using agro-climatic indicators and Canegro model: a case study in the main production region in Brazil. *Agric. Syst.* 154, 45–52. doi: 10.1016/j.agsy.2017.03.002

Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., and Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. *Comput. Electron. Agric.* 121, 57–65. doi: 10.1016/j.compag.2015.11.018

Perrone, M. P., and Cooper, L. N. (1992). *When Networks Disagree: Ensemble Methods For Hybrid Neural Networks*. Rhode Island: Brown University in Providence.

Shahhosseini, M., Hu, G., and Archontoulis, S. V. (2020). Forecasting Corn Yield With Machine Learning Ensembles. *Front. Plant Sci.* 11:1120 doi: 10.3389/fpls.2020.01120

Shahhosseini, M., Hu, G., Huber, I., and Archontoulis, S. V. (2021). Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt. *Sci. Rep.* 11:1606. doi: 10.1038/s41598-020-80820-1

Shahhosseini, M., Martinez-Feria, R. A., Hu, G., and Archontoulis, S. V. (2019). Maize yield and nitrate loss prediction with machine learning algorithms. *Environ. Res. Lett.* 14:124026. doi: 10.1088/1748-9326/ab5268

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv* Available Online at: http://arxiv.org/abs/1409.1556. (accessed April, 2021).

Soil Survey Staff [SSS]. (2019). Natural Resources Conservation Service United States Department of Agriculture Web Soil Survey Available Online at: https://websoilsurvey.nrcs.usda.gov/. (accessed April, 2021).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Stas, M., Van Orshoven, J., Dong, Q., Heremans, S., and Zhang, B. (2016). "A comparison of machine learning algorithms for regional wheat yield prediction using NDVI time series of SPOT-VGT," in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*. IEEE. (pp. 1–5).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Paper Presented At The Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, (Boston, MA, USA: IEEE).

Vincenzi, S., Zucchetta, M., Franzoi, P., Pellizzato, M., Pranovi, F., De Leo, G. A., et al. (2011). Application of a Random Forest algorithm to predict spatial distribution of the potential yield of Ruditapes philippinarum in the Venice lagoon, Italy. *Ecol. Model.* 222, 1471–1478. doi: 10.1016/j.ecolmodel.2011.02.007

Wang, A. X., Tran, C., Desai, N., Lobell, D., and Ermon, S. (2018). "Deep transfer learning for crop yield prediction with remote sensing data," in *Paper presented at the Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*. New York, United States: ACM

Wolpert, D. H. (1992). Stacked generalization. *Neural Netw.* 5, 241–259.

Yang, Q., Shi, L., Han, J., Zha, Y., and Zhu, P. (2019). Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images. *Field Crops Res.* 235, 142–153. doi: 10.1016/j.fcr.2019.02.022

You, J., Li, X., Low, M., Lobell, D., and Ermon, S. (2017). "Deep gaussian process for crop yield prediction based on remote sensing data," in *Paper presented at the Thirty-First AAAI conference on artificial intelligence*, (Menlo Park: AAAI).

Zhang, C., and Ma, Y. (2012). *Ensemble Machine Learning: Methods And Applications*. Germany: Springer.

Zhu, W., Ma, Y., Zhou, Y., Benton, M., and Romagnoli, J. (2018). "Deep Learning Based Soft Sensor and Its Application on a Pyrolysis Reactor for Compositions Predictions of Gas Phase Components," in *Computer Aided Chemical Engineering*, eds M. R. Eden, M. G. Ierapetritou, and G. P. Towler (Amsterdam: Elsevier), 44, 2245–2250. doi: 10.1016/b978-0-444-64241-7.50369-4

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.

Superiority Verification of Deep Learning in the Identification of Medicinal Plants: Taking *Paris polyphylla* var. *yunnanensis* as an Example

*JiaQi Yue[1,2], WanYi Li[1] and YuanZhong Wang[1]**

[1] Medicinal Plants Research Institute, Yunnan Academy of Agricultural Sciences, Kunming, China, [2] College of Traditional Chinese Medicine, Yunnan University of Chinese Medicine, Kunming, China

Medicinal plants have a variety of values and are an important source of new drugs and their lead compounds. They have played an important role in the treatment of cancer, AIDS, COVID-19 and other major and unconquered diseases. However, there are problems such as uneven quality and adulteration. Therefore, it is of great significance to find comprehensive, efficient and modern technology for its identification and evaluation to ensure quality and efficacy. In this study, deep learning, which is superior to conventional identification techniques, was extended to the identification of the part and region of the medicinal plant *Paris polyphylla* var. *yunnanensis* from the perspective of spectroscopy. Two pattern recognition models, partial least squares discriminant analysis (PLS-DA) and support vector machine (SVM), were established, and the overall discrimination performance of the three types of models was compared. In addition, we also compared the effects of different sample sizes on the discriminant performance of the models for the first time to explore whether the three models had sample size dependence. The results showed that the deep learning model had absolute superiority in the identification of medicinal plant. It was almost unaffected by factors such as data type and sample size. The overall identification ability was significantly better than the PLS-DA and SVM models. This study verified the superiority of the deep learning from examples, and provided a practical reference for related research on other medicinal plants.

Keywords: deep learning, identification research, medicinal plant, *Paris polyphylla* var. *yunnanensis*, superiority verification, ResNet

OPEN ACCESS

Edited by:
Kioumars Ghamkhar,
AgResearch Ltd., New Zealand

Reviewed by:
Jingli Lu,
AgResearch Ltd., New Zealand
Ke Han,
Harbin University of Commerce,
China

****Correspondence:***
YuanZhong Wang
boletus@126.com

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *05 August 2021*
Accepted: *03 September 2021*
Published: *22 September 2021*

Citation:
Yue JQ, Li WY and Wang YZ (2021) Superiority Verification of Deep Learning in the Identification of Medicinal Plants: Taking Paris polyphylla var. yunnanensis as an Example. Front. Plant Sci. 12:752863. doi: 10.3389/fpls.2021.752863

INTRODUCTION

Medicinal plants are a kind of highly exploitable plants with various values such as medicinal edible ecology. Their research has become the latest source for the emergence of new drugs (Newman and Cragg, 2015). The development potential of the international market for the utilization of medicinal plants is huge, and countries all over the world generally attach importance to its research in order to better transform and utilize medicinal plants, solve the problem of human survival resource shortage, and improve human health (Jamshidi-Kia et al., 2018). Medicinal plants have a wide range of sources. Due to differences in regional natural conditions, climatic conditions, flora and natural resources, they present a unique distribution with great differences in quantity and type (Deng et al., 2016). Many factors have different degrees of influence on the quality of medicinal plants. Therefore, the use of comprehensive, efficient, and modern technical means to clarify the region and part of medicinal plants has far-reaching significance for quality and efficacy.

Traditional identification and evaluation techniques for medicinal plants mainly include the technology of DNA barcoding, macroscopic identification, microscopic identification, chromatography, spectroscopy, etc. (Pang et al., 2011; Pei et al., 2020; Liu et al., 2021). Among them, spectroscopy has the advantages of simplicity, speed, economy, and high throughput, which can fully characterize the chemical information of samples with complex mixed systems (Pasquini, 2018). The identification research of medicinal plants mostly uses spectroscopy combined with chemometrics. Among them, the partial least square discriminant analysis (PLS-DA) and support vector machine (SVM) have excellent performance, and have been successfully applied to the identification and evaluation of a variety of medicinal plants, including species identification, origin identification, age identification, part identification, adulteration identification, etc. (Liu et al., 2020; Shen et al., 2020; Wang et al., 2020) Yang and Wang (2018) compared the effects of PLS-DA and SVM on the identification of *P. polyphylla* var. *yunnanensis* from different regions based on infrared spectroscopy and ultraviolet spectroscopy data. It is found that both models have higher recognition performance, and the accuracy of SVM is higher than that of PLS-DA.

In addition, two-dimensional correlation spectroscopy (2DCOS) is also a powerful tool for identification evaluation. This technology fully combines the advantages of computational chemistry, statistics, spectroscopy and computer science to increase the spectral resolution and enrich the information carried by the spectrum by increasing the dimension (Noda, 1989, 1993). In recent years, reports on the research and application of 2DCOS technology are increasing year by year, covering drug metabolism, drug toxicology, drug structure-activity relationship, traditional Chinese medicine, etc. (Noda, 2004, 2014, 2016; Li et al., 2014). Based on years of research, Sun et al. (2003) wrote a book called *"Atlas of Two-dimensional Correlation Infrared Spectroscopy for Traditional Chinese Medicine Identification,"* which contains the 2DCOS spectra of more than 300 kinds of traditional Chinese medicine, providing a reference for the identification research of related traditional Chinese medicine. However, the artificial identification and analysis of 2DCOS spectra has limitations in time, technology, and experience. Moreover, interdisciplinary research has become a current hot spot and also the trend of future scientific research field. Therefore, it is necessary to combine 2DCOS with more modern, convenient and intelligent technical means of other disciplines to realize the rapid identification of medicinal plants.

Deep learning is the main research method used in the development of artificial intelligence research at the present stage, which has unique advantages in image classification and object recognition (LeCun et al., 2015; Houssein et al., 2021). Combining it with 2DCOS images for the identification of medicinal plants can take advantage of the respective advantages of the two technologies and greatly improve the efficiency of identification and analysis. Deep learning combined with 2DCOS seems to show superior performance in many aspects than traditional spectroscopy combined with chemometrics in identifying medicinal plants (Dong et al., 2020). For example, deep learning can achieve good identification without complex spectral preprocessing, and there is no need to manually extract features in the modeling process, which greatly improves efficiency and reduces various risks caused by human factors (Grinblat et al., 2016). However, these conclusions are all based on theories or the application of a single method, and there has been no actual comparison and discussion on them.

Paris polyphylla var. *yunnanensis* (PPY), as the original plant of the precious Chinese medicine Paridis Rhizoma, is a medicinal plant resource with a representative and global influence (Cunningham et al., 2018). In the market, there are more than 80 commonly used Chinese patent medicines with Paridis Rhizoma as the main raw material, and 107 pharmaceutical companies are involved in the production, which are distributed in 23 provinces of China. They have significant clinical efficacy and economic value (Tao et al., 2020). At present, domestic and foreign scholars have conducted a lot of research on PPY, but the research on the resources evaluation is still in a situation where there are results but no conclusions, and they are all based on the traditional medicinal rhizoma. Moreover, studying the above-ground parts of PPY can promote the development and utilization of non-medicinal parts, and improve economic benefits (Zhao et al., 2021). Besides, there is currently no research on the use of deep learning combined with 2DCOS to identify the parts and regions of PPY.

In conclusion, taking PPY as an example, two pattern recognition models of PLS-DA and SVM, and a deep learning model of Residual neural network (ResNet) were established in this study to explore and verify whether deep learning combined with 2DCOS has advantages in the identification of medicinal plant resources. In order to increase comparability and credibility, we simultaneously identified and evaluated PPY samples of different regions and parts. In addition, we also compared the impact of different sample sizes on model identification performance to explore whether the three models are dependent on sample size. This research not only provided a reasonable, standardized, fast and effective method for the identification of regions and parts of PPY, but also verified the superiority of the deep learning model in the identification of medicinal plants and the response of the three models to sample size. This is conducive to the development and utilization of advanced deep learning models such as ResNet in other fields.

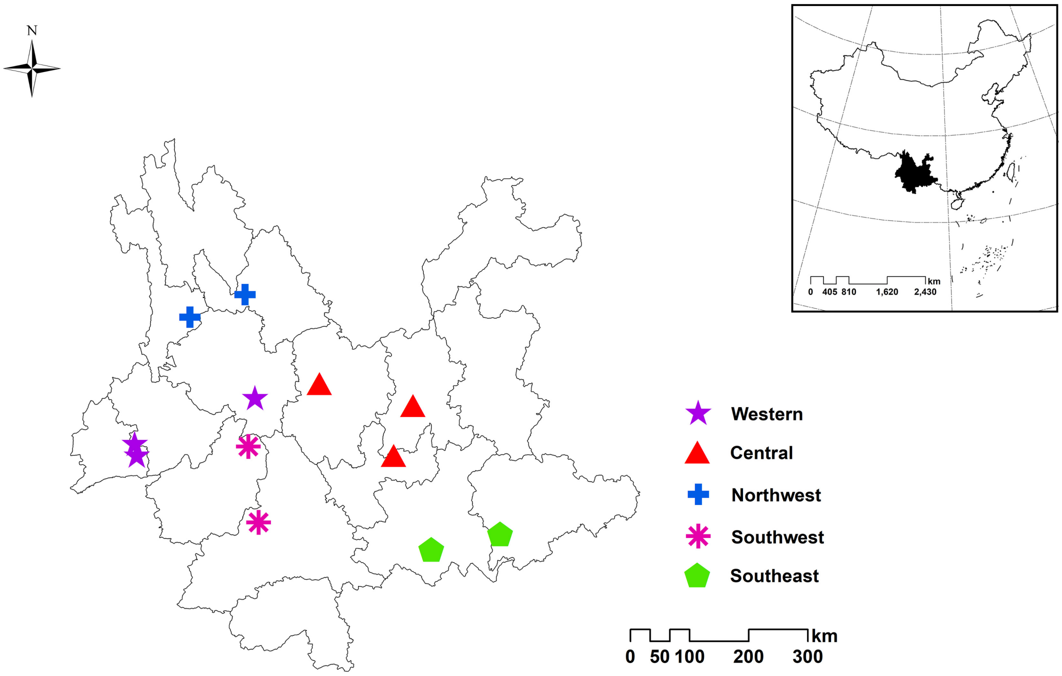


FIGURE 1 | Location distribution of *Paris polyphylla* var. *yunnanensis* samples in western, central, northwest, southwest and southeast of Yunnan.

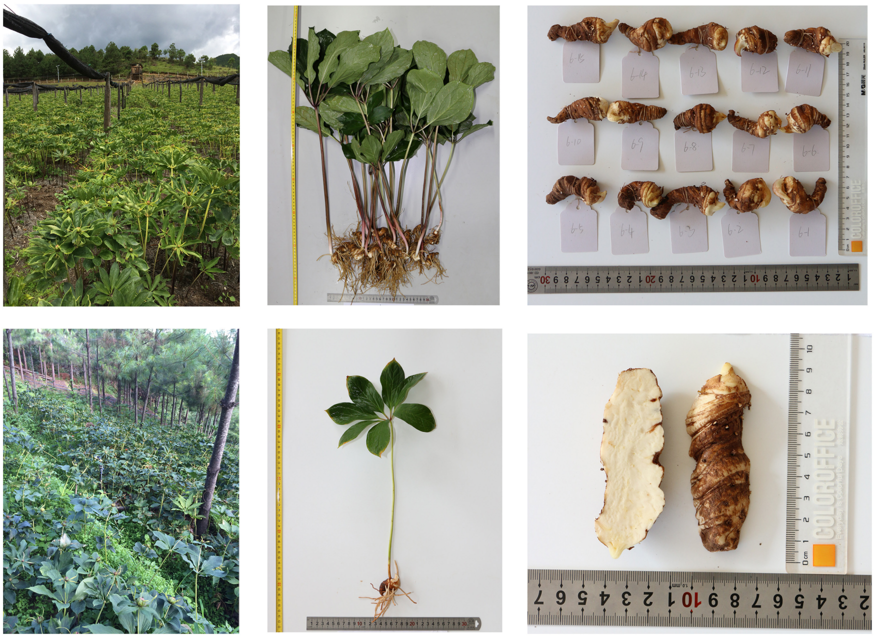


FIGURE 2 | Sample picture of the planting site, whole plant and rhizome of *Paris polyphylla* var. *yunnanensis*.

MATERIALS AND METHODS

Sample Information

A total of 772 individuals were collected in 12 sampling sites in central, northwest, southeast, southwest and western Yunnan (**Figure 1**). All samples were identified as *Paris polyphylla* var. *yunnanensis* by Professor Hang Jin from the Institute of Medicinal Plants, Yunnan Academy of Agricultural Sciences. Some samples are shown in **Figure 2**. Afterward, all the samples were cleaned and divided into four parts: rhizome, stem, leaf and

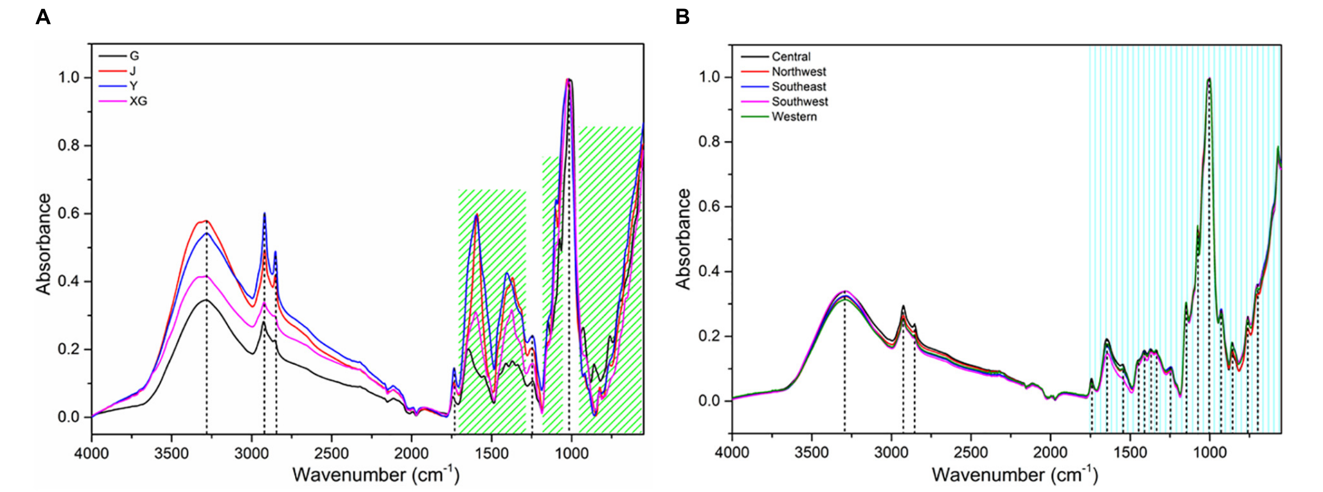


FIGURE 3 | Averaged raw spectra of *Paris polyphylla* var. *yunnanensis*. **(A)** parts; **(B)** regions. The G, J, Y, and XG represent the rhizome (G), stem (J), leaf (Y) and fibrous root (XG), respectively.

fibrous root. Then the samples were dried to a constant weight at 50°C in an electric thermostatic drying oven. Next, the samples were passed through a 100-mesh sieve. Finally, the fine powders were stored in self-sealed bags and kept in a dry environment away from light for subsequent analysis. The detailed information of the samples is shown in **Supplementary Table 1**. There are a total of 772 rhizomes, all of which were used for regions identification analysis. Rhizome (G: 142), stem (J: 107), leaf (Y: 137), and fibrous root (XG: 107) from Dehong and Yuxi were selected for identification of parts.

FT-MIR Spectra Acquisition

The Fourier transform mid-infrared spectra were collected by a Fourier transform infrared spectrometer equipped with an attenuated total reflection accessory (Perkin Elmer, Norwalk, CT, United States). Sample powder (2 ± 0.2 mg) was placed in the center of the metal ring (ZnSe crystal surface), and the manometer knob was adjusted to a uniform progress bar of 131 ± 1 to form sample powder sheets with the same thickness. The infrared spectrum scanning range was set to be 4,000–550 cm^{-1} with a spectral resolution of 4 cm^{-1}. Sixteen times of scanning were carried out, and each sample was measured in parallel for three times. Finally, the average spectrum was taken. Before the sample scanning, the infrared spectrum of the blank crystal surface is collected, and the interference of air and the scattering spectrum of the crystal part was deducted. During the spectrum measurement, keep the laboratory temperature at 25°C and the relative air humidity at 30%.

Data Processing and Exploratory Analysis

Although the spectral data preprocessing and the characteristic variable selection have been proved by previous studies to be effective for optimizing identification model (Obaid et al., 2019), the complex data preprocessing process will greatly reduce the recognition efficiency. Moreover, the preprocessing methods and characteristic variable selection methods used for different data sets cannot be unified, which requires a lot of time and resource costs to verify. Therefore, this study directly used original spectral data for subsequent identification analysis without considering data preprocessing and characteristic variable selection, so as to fairly compare the recognition performance of the three types of models and verify whether the ResNet model has advantages in the identification research.

In addition, in order to explore the impact of sample size on the recognition ability of the three types of models, we divided the data sets of region and part into low sample size group (10%), medium sample size group (50%), and high sample size group (100%), and the percentage in parentheses is the proportion of each group of samples (**Supplementary Table 2**). The Kennard-stone algorithm was performed to divide the data of all groups into training set (2/3) and test set (1/3), which was directly used to build PLS-DA and SVM models. The data for establishing the ResNet model is the 2DCOS images of all groups, and the generation method is shown in the following section.

Exploratory analysis used the unsupervised analysis method of t-distributed stochastic neighbor embedding (t-SNE) to summarize the distribution of grouped samples in a multivariate space. By identifying the distribution trend of samples, high-dimensional data can be visualized as data points in two-dimensional or three-dimensional graphs. The above process was completed by MATLAB software.

Two-Dimensional Correlation Spectroscopy Spectra Image Acquisition

The generalized two-dimensional correlation spectrum is an effective method to improve spectral resolution and solve spectral overlap by designing disturbance variables, which is obtained by discrete generalized 2DCOS algorithm. Its dynamic spectrum is expressed as ***S***, and the expression is as follows, where *v* is variable

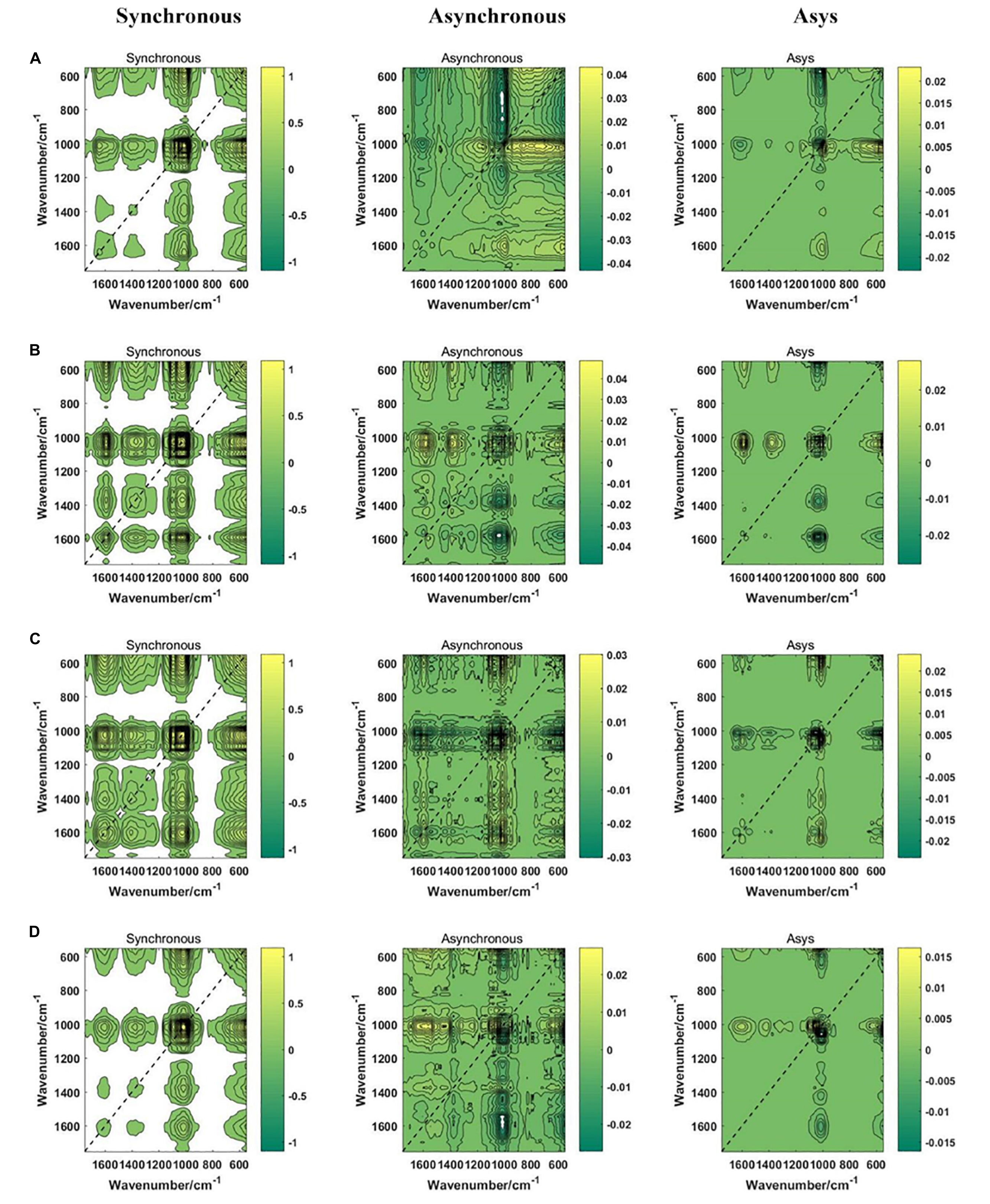


FIGURE 4 | The synchronous, asynchronous and integrated 2DCOS images of parts. **(A)** rhizome; **(B)** stem; **(C)** leaf; **(D)** fibrous root. Asys images are i2DCOS images.

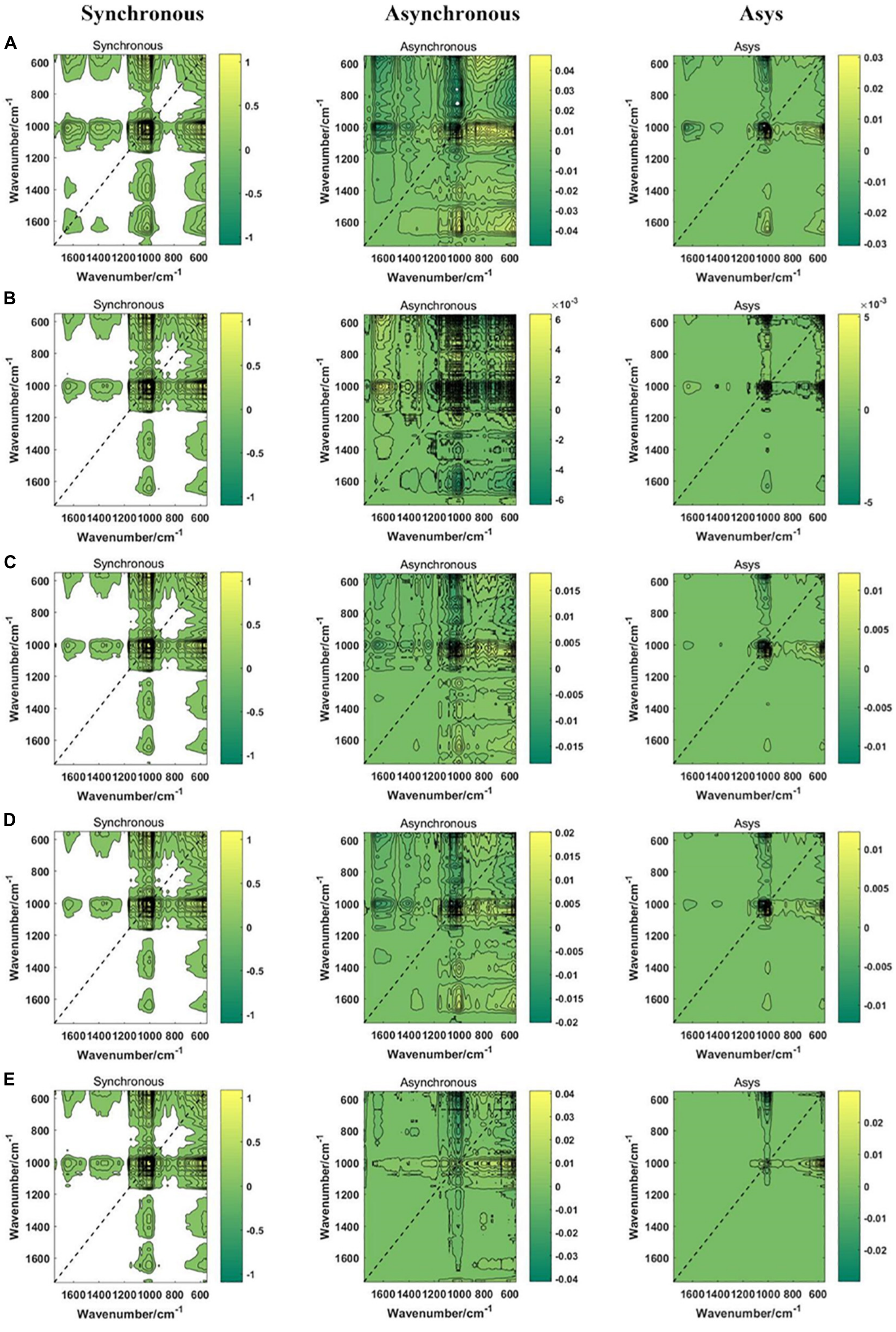


FIGURE 5 | The synchronous, asynchronous and integrated 2DCOS images of regions. **(A)** central; **(B)** northwest; **(C)** southeast; **(D)** southwest; **(E)** western. Asys images are i2DCOS images.

TABLE 1 | Parameters for PLS-DA models in parts and regions discrimination based on three levels of data sets.

Data	Model	LVs	R^2	Q^2	RMSEE	RMSECV	RMSEP	Accuracy (%)	
								Training set	Test set
Parts	PLS-DA-L	1	0.198	0.159	0.374135	0.37687	0.295164	51.52	55.56
	PLS-DA-M	11	0.899	0.831	0.143237	0.167712	0.0758287	99.39	100
	PLS-DA-H	11	0.918	0.887	0.120499	0.138129	0.0669199	99.39	100
Regions	PLS-DA-L	/	/	/	/	/	/	/	/
	PLS-DA-M	14	0.584	0.333	0.349237	0.441024	0.325103	87.92	88.46
	PLS-DA-H	20	0.698	0.544	0.266242	0.351347	0.266231	95.34	92.22

and t is the external disturbance (Noda, 2018).

$$S\ (v)\ =\ \begin{bmatrix} y(v,t_1) \\ y(v,t_2) \\ y(v,t_3) \\ \cdot \\ \cdot \\ \cdot \\ y(v,t_m) \end{bmatrix} \tag{1}$$

The synchronous spectral intensity $\Phi(\boldsymbol{v_1},\boldsymbol{v_2})$ is equal to the cross product of the dynamic spectral intensity at (v_1, v_2). The asynchronous spectral intensity $\Psi(v_1,v_2)$ is equal to the cross product of the Hilbert-Noda matrix defined as N_{jk} for the dynamic spectral intensity at (v_1, v_2). Their expressions are as follows:

$$\Phi\ (v_1, v_2)\ =\ \frac{1}{m-1} S\,(v_1)^T \cdot S\,(v_2) \tag{2}$$

$$\Psi\ (v_1, v_2)\ =\ \frac{1}{m-1} S\,(v_1)^T \cdot N \cdot S\,(v_2) \tag{3}$$

$$N_{jk} = \begin{cases} 0 & j = k \\ \frac{1}{\pi(k-j)} & j \neq k \end{cases} \tag{4}$$

The product of a pair of synchronous and asynchronous correlation intensities can obtain the integrated two-dimensional correlation intensity, which is expressed as I (v_1, v_2) (Chen et al., 2018).

$$\mathrm{I}(v_1, v_2) = [\Phi(v_1, v_2)] \cdot [\Psi(v_1, v_2)]$$

$$= \frac{1}{(m-1)^2} [S(v_1)^T \cdot S(v_2)] \cdot [S(v_1)^T \cdot N \cdot S(v_2)] \tag{5}$$

Spectral data matrix S(m × n) contains two spectra, the first is the average FT-MIR of each class, and the second is the ith FT-MIR spectra of each class. The synchronous 2DCOS spectra, asynchronous 2DCOS spectra and integrative 2DCOS (i2DCOS) spectra for the ith sample of each category can be obtained by equation (2), (3) and (4). In order to reduce the amount of calculation, save computer resources and speed up the calculation efficiency, the fingerprint area of 1,750–550 cm^{-1} was selected, and the synchronous 2DCOS, asynchronous 2DCOS and i2DCOS spectral images were automatically generated by the software Matlab2017b. The image size can be chosen according to the processing power of the computer (32 × 32 pixel, 64 × 64 pixel and 128 × 128 pixel), and the generated 2DCOS images were stored in JPEG image format with the size as 64 × 64 pixel in the corresponding folder for building ResNet model. Using the Kennard-stone algorithm, all datasets were divided into training set (60%), test set (30%), and external validation set (10%). The process of generating all types of 2DCOS spectra images is shown in **Supplementary Figure 1**.

Partial Least Squares Discrimination Analysis

Partial least squares discriminant analysis is a linear supervised classification method established on the basis of the standard PLS regression algorithm. It searches for the variable with the largest covariance of the classification matrix Y from the variable matrix X. Y is divided into two categories, where Y = 1 represents that the sample belongs to a specific category, and Y = 0 represents that the sample does not belong to a specific category. Finally, the probability of each sample classified into each category is obtained. In the calculation, the observed X matrix is transformed into a set of several intermediate linear latent variables (LVs). The first n LVs are selected according to the maximum eigenvalue greater than 1. The statistical parameters of accuracy, model fitting determination coefficient R^2, Q^2, root mean square error of estimation (RMSEE), root mean square error of cross validation (RMSECV), and root mean square error of prediction (RMSEP) are used to evaluate the performance of the model. Permutation

TABLE 2 | The accuracy of SVM models for parts and regions identification based on three levels of data sets.

Data	Model	Best *c*	Best *g*	Accuracy (%)	
				Training set	Test set
Parts	SVM-L	2,048.00	0.000043	72.73	100.00
	SVM-M	181.02	0.00069	98.18	100.00
	SVM-H	5.66	0.016	99.39	100.00
Regions	SVM-L	1.00	0.10	0.00	46.15
	SVM-M	11,585.24	0.00017	87.92	92.31
	SVM-H	46,340.95	0.000031	94.17	97.28

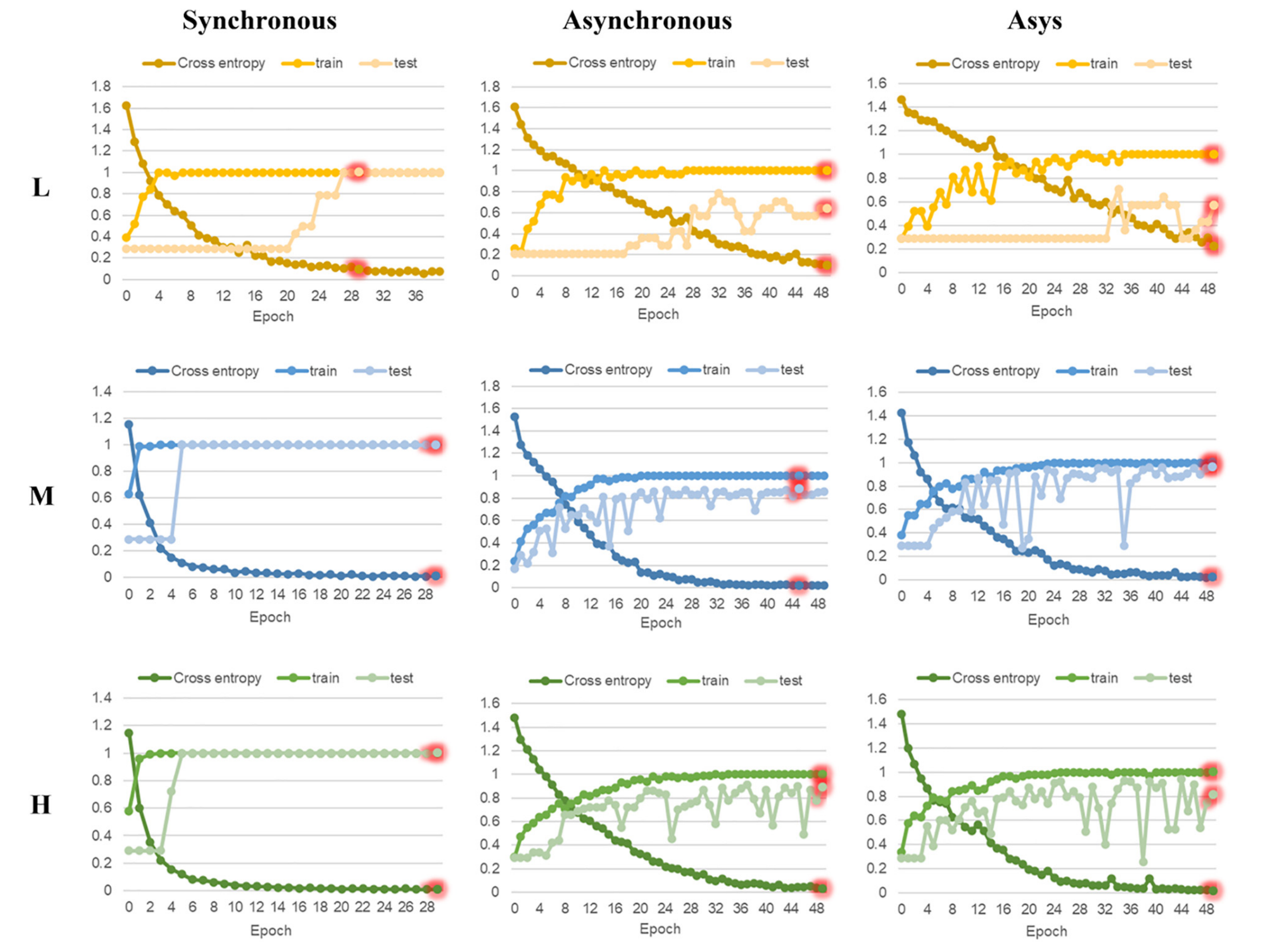


FIGURE 6 | The accuracy curves and cross-entropy cost function of ResNet models based on part data with different sample size. L, low sample size; M, medium sample size; H, high sample size.

test was performed on the established model with a total of 50 iterations. And according to the R^2-intercept and Q^2-intercept results, the fitting degree of the model was verified. The process of establishing PLS-DA model was carried out on SIMCA-P+14.1 software.

Support Vector Machine

Support vector machine is a supervised pattern recognition method that can identify unknown samples and has the ability to analyze the data with high collinearity and high noise. The libsvm-3.20 toolbox developed by the Institute of Industrial Engineering, National Taiwan University, Lin Zhiren, etc., was used to establish SVM discriminant models to identify the region and part of *P. polyphylla* var. *yunnanensis*. The 1,789 data points of the original FT-MIR spectra were used as the X variable, and the classification labels were used as the *Y* variable. The training set was used to establish discriminant models, and the text set was used to externally verify the accuracy of models. The best kernel functions *c* and *g* were obtained by cross validation of grid search method. The SVM models were implemented using Matlab software.

Residual Neural Network

In this study, a 12-layer ResNet was established with a weight attenuation coefficient λ of 0.0001 and a learning rate of 0.01. **Supplementary Table 3** showed the ResNet network parameter configuration. The model was completed by the anaconda data processing hardware platform, and MXNet was selected as the deep learning framework. The model contains two kinds of residual block, namely the identity residual block (**Supplementary Figure 2**) and the convolutional residual block (**Supplementary Figure 3**). The block is selected according to whether the dimensions of the input and output are consistent. When the dimensions of the input and output are the same, the identity residual block is used to build the model. When the input and output dimensions are inconsistent, we introduce the convolutional residual block with

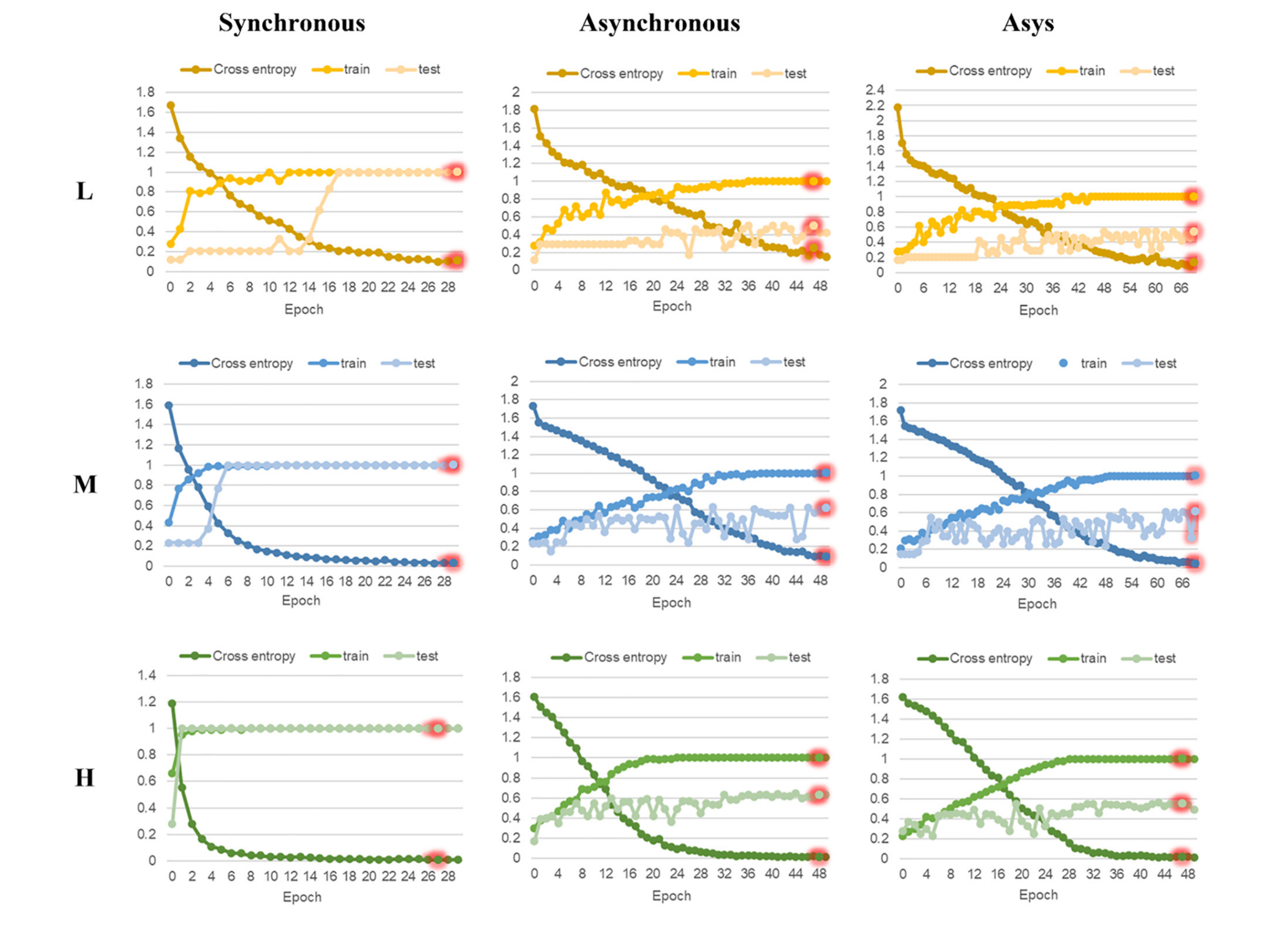


FIGURE 7 | The accuracy curves and cross-entropy cost function of ResNet models based on region data with different sample size. L, low sample size; M, medium sample size; H, high sample size.

a convolution kernel size of 1 × 1 to match the dimensions of the input and output. The model structure is shown in **Supplementary Figure 4**, where the input data is synchronous 2DCOS, asynchronous 2DCOS and i2DCOS spectral images. The identification flow chart of ResNet is shown in **Supplementary Figure 5**. The training set is used to train the model. The Stochastic Gradient Descent (SGD) method is used to find the optimal parameters for minimizing the loss function value to obtain the optimal model. The test set is used to verify whether the performance of the final model is optimal. The external validation set is used to verify the generalization ability of the model.

RESULTS AND DISCUSSION

FT-MIR Spectra Analysis

Figure 3 shows the average FT-MIR spectra of four parts and five regions of PPY. 3,350, 2,940, 1,645, 1,387, 1,069, 931, 581 cm^{-1} are the main characteristic absorption peaks of PPY samples. The absorption peak of O-H stretching vibration is mainly near 3,350 cm^{-1} (Pei et al., 2018). The absorbance intensity around 2,940 cm^{-1} is related to the stretching vibration of C-H absorption of lipids (Pei et al., 2019). The absorption peak at 1,645 cm^{-1} is assigned to the C = C and C = O stretching vibration of steroid saponin and flavonoid (Wu et al., 2019). The absorption peak near 1,387 cm^{-1} is -CH_3 symmetrical bending vibration (Yang et al., 2019). In the region of 1,300–550 cm^{-1}, the absorption peaks correspond to the stretching vibration peak of C-O and the bending vibration of O-H, which belong to substances such as sugars and saponins (Wu et al., 2018). It is concluded that the main components in the plant of PPY are flavonoids, starch and glycosides.

As shown in **Figure 3A**, the absorption peak intensity of rhizome, stem, leaf and fibrous root is significantly different, especially the absorption peak in the band of 4,000–1,200 cm^{-1}. On the whole, the order of absorption intensity of four parts is Y > J > XG > G. It may imply that the distribution and content of active components in different parts of PPY are significantly different, and the components content of non-medicinal parts (Y, J, and XG) may be higher than the medicinal parts (G), which is nearly consistent with the research results

TABLE 3 | The accuracy of ResNet models for parts and regions identification based on three levels of data sets.

Data	Code	Type	Epoch	Loss value	Accuracy		
					Train (%)	Test (%)	External validation (%)
Parts	Resnet-L	**Synchronous**	**29**	**0.091**	**100**	**100**	**100**
		Asynchronous	49	0.102	100	64	100
		Asys	49	0.219	100	57	100
	Resnet-M	**Synchronous**	**29**	**0.012**	**100**	**100**	**100**
		Asynchronous	45	0.021	100	88	87.5
		Asys	49	0.021	100	96	100
	Resnet-H	**Synchronous**	**29**	**0.009**	**100**	**100**	**100**
		Asynchronous	49	0.027	100	89	90
		Asys	49	0.017	100	81	88
Regions	Resnet-L	**Synchronous**	**29**	**0.114**	**100**	**100**	**100**
		Asynchronous	47	0.248	100	50	25
		Asys	69	0.132	100	54	37.5
	Resnet-M	**Synchronous**	**29**	**0.030**	**100**	**100**	**100**
		Asynchronous	49	0.088	100	62	56.4
		Asys	69	0.045	100	61	66.7
	Resnet-H	**Synchronous**	**27**	**0.009**	**100**	**100**	**100**
		Asynchronous	48	0.011	100	63	62.7
		Asys	47	0.020	100	55	64

Note: The bold value are the optimal results of models under the certain data set.

of Feng et al. (2015). However, the differences of peak shape and absorption intensity in different regions (**Figure 3B**) are much lower than those in different parts, which indicates that the differences within individuals may be greater than the differences between individuals, and it's easier to identify parts than regions. Nonetheless, further modeling analysis and more studies are needed to support this conclusion.

The Two-Dimensional Correlation Spectroscopy Spectra Images

In this study, a total of 6,135 2DCOS images were drawn, including synchronous 2DCOS, asynchronous 2DCOS and i2DCOS images of PPY in different parts (**Figure 4**) and different regions (**Figure 5**). The synchronous 2DCOS images are symmetric along diagonals, and the correlation peaks may appear on or off the diagonal. The correlation peak on the diagonal line is called the auto peak, which is expressed as the value of the auto-correlation function of spectral intensity change (Huang et al., 2003). The peaks on both sides of the diagonal are called cross peaks and represent synchronous changes of spectral signals at different wavenumbers. The asynchronous 2DCOS images characterize the asynchronous characteristics of the absorption intensity measured at two different wavenumbers. It is anti-symmetric on both sides of the diagonal, and it has only cross peaks and no automatic peaks (Noda, 1990). The i2DCOS is defined as the product of the synchronous and asynchronous two-dimensional correlation intensities. It can provide correlation spectra with equal resolution, and its characteristics are clearer than asynchronous 2DCOS (van der Maaten and Hinton, 2008). By comparing the synchronous, asynchronous and integrated 2DCOS, it is not difficult to see that the colors and lines of the synchronous images are clearer and richer, and it is easy to analyze the differences and intensity changes of auto peaks and cross peaks between different samples. However, asynchronous and integrated images are complex and changeable, and cannot be distinguished by naked eyes. This may be caused by the complex characteristics of traditional Chinese medicine. In addition, the 2DCOS images of different parts has more significant differences than that of different regions, which is consistent with the results presented by the one-dimensional spectral analysis.

In summary, synchronous 2DCOS has better performance of visual recognition. Different parts are easier to distinguish than different regions. Although 2DCOS overcame the shortcomings of one-dimensional spectral peak overlap and improved its apparent resolution, it was very difficult to recognize different parts and regions by visual analysis alone, so we need to rely on machine learning methods.

Exploratory Analysis of t-Distributed Stochastic Neighbor Embedding

As a relatively novel non-parametric dimensionality reduction technology, t-SNE can visualize high-dimensional data to obtain the position of each data point on a two-dimensional or

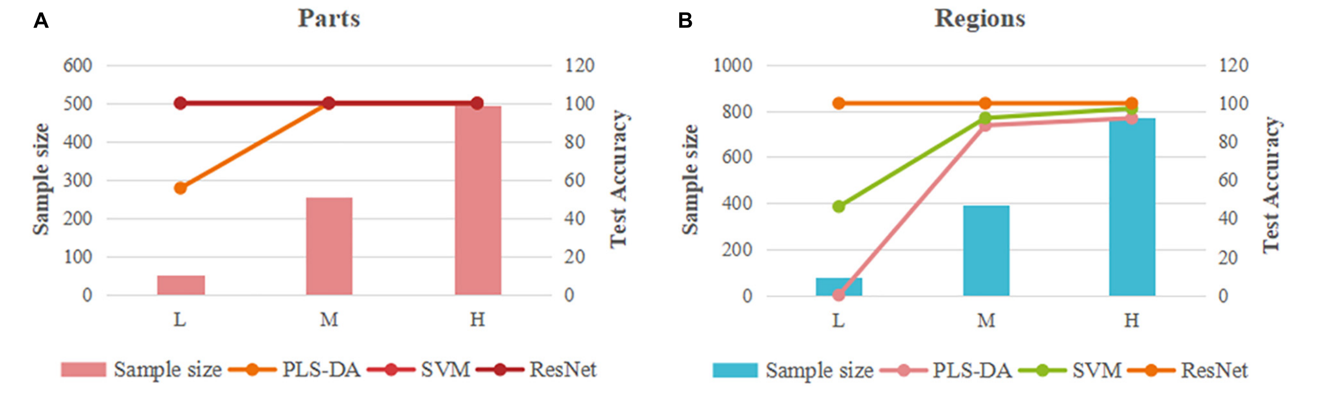


FIGURE 8 | Comparison of the overall identification performance of PLS-DA, SVM and ResNet models. **(A)** parts; **(B)** regions.

three-dimensional map. Its focus is to maintain the basic structure of the data matrix to reveal outliers or similarities and differences between groups of observed variables. As shown in **Supplementary Figure 6**, t-SNE was used in this study to conduct a preliminary visual evaluation of the spectral data sets. The ellipses in the figure represented the detailed trends of different types of samples. **Supplementary Figure 6A** showed the distribution of FT-MIR data sets of different parts, in which there were obvious outliers in both fibrous roots and roots. But in general, most samples could be clustered according to different category, and a few samples were mixed together. **Supplementary Figure 6B** showed the distribution of FT-MIR data sets of different regions, which formed a sharp contrast with the data set of different regions. The samples from the five regions were almost completely blended together. The two-dimensional visual results showed that the FT-MIR information of PPY samples in different regions was relatively similar, and it is not easy to distinguish. The results of these exploratory data analysis were consistent with the results of spectrum analysis, that is, the difference between different parts of PPY was higher than that of different regions. Obviously, in the process of data visualization, the vast majority of samples cannot be classified according to their pre-identified labels of different sources. Therefore, further in-depth modeling analysis should be considered.

Discrimination Results of Partial Least Squares-Discriminant Analysis Model

The PLS-DA models for the parts and regions of PPY based on different sample size data sets were, respectively, established. **Table 1** lists all the model parameters and the results of discrimination accuracy. From the table, we can clearly know that the models of different parts, different regions and different sample sizes have significant differences in the identification ability and model performance. In addition, in order to assess whether the PLS-DA model has an over-fitting problem, a permutation test was performed on all models. Generally, if the intercept of R^2 is less than 0.4, there is no risk of over-fitting. **Supplementary Figure 7** shows the results of the permutation test of five classification models (PLS-DA model cannot be established based on the low sample size data of the region). The results show that the R^2 intercepts of the five models are all less than 0.4, and there is no risk of over-fitting. The confusion matrices of the established PLS-DA models based on the data set of parts and regions are shown in **Supplementary Tables 4**, **5**, respectively.

First of all, from the models based on different parts of the data set, we can see that the R^2 and Q^2 of the PLS-DA-L model are only 0.198 and 0.159, respectively, which are both lower than 0.5, and the recognition accuracy of the test set is only 55.56%. Therefore, the model based on the low sample size data set has poor performance and low discrimination ability, and cannot realize the discrimination of different parts of PPY. The PLS-DA-M and PLS-DA-H models based on the data sets of parts have high R^2 and Q^2 values greater than 0.8 and low RMSEE, RMSECV and RMSEP values. The accuracy of the test sets of the two models is 100%, which has a very good recognition performance.

Secondly, as shown in the table, the PLS-DA-L model based on regions data cannot be fitted. This result may be related to the amount of data being too small or the data is not preprocessed. Although the PLS-DA-M model has a test set accuracy rate of 88.46%, the model performance is poor with low Q^2 and high RMSEE, RMSECV, and RMSEP values. The PLS-DA-H model is better than the low sample size model and the medium sample size model in terms of model performance and recognition accuracy, so that it can well identify PPY in different regions.

Finally, from the perspective of sample size, whether it is PLS-DA models based on part data or models based on region data, the recognition performance is dependent on the sample size. And it shows that the larger the sample size, the better the model performance and the stronger the recognition ability. However, with the increase of the sample size, the recognition efficiency of the models will be greatly reduced. In addition, through comparison, it can be concluded that the PLS-DA models based on part data is better than that based on region data, regardless of model parameter results or recognition accuracy.

Discrimination Results of Support Vector Machine Model

Support vector machine is a supervised classification tool. It searches for the optimal separation hyperplane between different data categories by maximizing the distance between the classification hyperplane and various sample points. SVM contains two parameters, *c* is used as a penalty parameter, which can control the generalization ability of the model and reduce the over-fitting phenomenon, and the kernel function parameter *g* is related to the stability of the model. **Supplementary Figures 8**, **9** are the optimal separation hyperplane graph and classification result graph of the SVM model based on parts and regions data, respectively. The detailed results of the six SVM models are shown in **Table 2**. Best *c* and Best *g,* respectively, represent the best penalty parameter and kernel function parameter of the model.

The accuracy difference between the training set and the test set of the SVM-L model based on part data and region data is more than 20%, while the accuracy of the training set and the test set of the SVM-M and SVM-H models based on part data and region data is less than 5%. This shows that the reliability of the SVM models established with low sample size data is poor. The SVM-M and SVM-H models based on part data both have high identification accuracy and low Best *c* value, so the model performance is good and have the ability to identify different parts of PPY. However, although the SVM-M and SVM-H models based on region data have high identification accuracy, their Best *c* values are abnormally high, indicating that the performance of the two models is poor and there may be over-fitting, which can't well identify the PPY in different regions. The above results show that although a larger sample size can improve the identification accuracy of the SVM model, the establishment of a high-performance model cannot be achieved for data that has not been preprocessed and has small differences between different categories. In addition, as with the results of the PLS-DA model, it is easier to identify the parts of PPY than the regions.

In conclusion, although the SVM model has the advantage of solving the problems of small sample, nonlinear and high-dimensional data (Noble, 2006), the unpreprocessed small sample data in this study is not applicable to the SVM model, indicating that data preprocessing is very necessary to improve the discrimination performance of traditional models such as SVM. In addition, a larger sample size increases the over-fitting risk of SVM model while improving the recognition accuracy, which leads to poor model performance and low reliability.

Discrimination Results of Residual Neural Network Model

In this research, ResNet models based on 2DCOS images (including synchronous, asynchronous and integrated images) of FT-MIR were established. **Figures 6**, **7** are the results of 18 ResNet models based on the data sets of parts and regions, respectively, showing the accuracy curves and cross-entropy cost function curves. The accuracy curves, includes the training set and the test set, were used to evaluate the discrimination ability of the model. The closer its value is to 1, the stronger the discrimination ability of the model. The cross-entropy loss function was used to explain the convergence effect of the model. The closer its value is to zero, the better the convergence effect of the model. In addition, the external validation set was classified using the models established above, and the classification result of the external validation set of different parts and regions was shown in the confusion matrix in **Supplementary Figures 10**, **11**, respectively. External validation is used to judge and evaluate the pros and cons of the model to ensure the stability of the established model. **Table 3** summarized the result parameters of all models, including accuracy (training set, test set and external validation set), epoch, and loss value.

Comparing the models based on synchronous, asynchronous and integrated 2DCOS images, we can get that the model of synchronous 2DCOS images has the best discrimination effect, and the accuracy of the training set, test set and external verification set is 100%. The modeling results are consistent with the results of image vision analysis, that is, the synchronized 2DCOS images have clearer characteristic peaks and can better characterize different types of samples. Comparing the models with low, medium and high sample sizes showed that the ResNet model had no dependence on the sample size, and there was no obvious rule between the identification accuracy and the sample size. However, too small sample size will lead to poor performance and over-fitting of model. This result can be derived from the identification results of low sample size models based on asynchronous and integrated 2DCOS images. The difference of identification accuracy between the external validation set and the test set was large, and the loss value of models was significantly higher than that of the medium sample size and high sample size models. In addition, the accuracy curves of the training set and test set of the medium sample size and high sample size models showed a consistent upward trend, which also showed that these two types of models had no risk of over-fitting and were robust. However, the accuracy curve of the training set and the test set of the low sample size model had a poor consistency in the upward trend, even for the optimal model of synchronous 2DCOS images, which indicated that the low sample size would reduce the performance of the ResNet model. Finally, on the whole, the recognition effect of the ResNet model based on the part data set was better than that of the ResNet model based on the region data set.

In summary, the recognition accuracy of the models based on synchronous 2DCOS images is the best, which is almost not affected by sample size, part, region and other factors, and is most suitable for the identification of medicinal plants. However, too small sample size does have a small negative impact on the performance of the ResNet model. Therefore, it is worth thinking about how to use an appropriate method to solve the negative impact of low samples on model performance. This is conducive to solving the identifying problem of research subjects with a small sample size. These research objects have very limited data, and it is expensive or impossible to obtain more data, such as scarce and precious animal and plant resources.

Comparison Analysis of Models

Partial least squares discriminant analysis, SVM, and ResNet models showed significant differences in their ability to identify the parts and regions of the PPY, the responses to different sample sizes, and the comprehensive performance of models. As shown in **Figure 8**, we have made a visual comparison of three type of models.

In terms of the identification ability of parts and regions, the three types of models show consistent results, that is, the identification ability of parts is better than that of regions, which indicates that the difference of parts data of PPY is greater than that of regions data. This result implies that the difference in component within the sample may be greater than that between samples. This causes us to think about the resource evaluation and the effective development and utilization of the non-medicinal parts of PPY. In addition to the evaluation of the advantages and disadvantages of the medicinal parts between individuals in different origins, the development and utilization of non-medicinal parts within individuals is also very worthy of attention.

From the perspective of different sample sizes, the three models have different responses to low, medium, and high sample size data. The PLS-DA model has a very significant sample size dependence. As the sample size increases, the discrimination ability and the performance of the model have been significantly improved. It can be concluded that the overall performance of the PLS-DA model is positively correlated with the sample size. This result is confirmed by two types of models based on part and region data, which greatly reduces the chance. There is a certain correlation between the merits and demerits of SVM model and the sample size, but not a complete positive or negative correlation. The identification accuracy of the model increases with the increase of the sample size, while the performance of the model based on region data evaluated by parameters will deteriorate with the increase of the sample size. It can be concluded from this study that there are two important factors affecting the overall performance of SVM model, one is the quality of data itself, the other is the sample size. The ResNet model based on the synchronous 2DCOS images has a very perfect overall discrimination performance, both in terms of the discrimination accuracy and the model parameters. It is not limited by the sample size and is almost unaffected by the data itself. Whether it is based on easy-to-identify part data or region data with small differences, it can achieve 100% recognition accuracy.

In summary, the PLS-DA model has the strongest dependence on the sample size, followed by SVM, and the ResNet model based on synchronized 2DCOS images has almost no dependence on the sample size. In addition, the traditional pattern recognition model is also affected by the quality of data itself. Therefore, the ResNet model based on synchronized 2DCOS images occupies an absolute advantage in the identification of medicinal plants. The model is universal and does not require preprocessing or artificial extraction of characteristic variables. It has good discrimination accuracy regardless of the sample size or the quality of the data.

CONCLUSION

In this study, we used three kinds of models to identify the part and region of PPY. PLS-DA and SVM are traditional pattern recognition models, which have been widely used in the past research. ResNet model is a representative dominant model in deep learning. The effects of different types of data and different sample sizes on the discrimination ability and performance of the three models were discussed without any data preprocessing. By comparing the ability of the traditional model and the deep learning model for the identification of PPY, we found that the identification performance of PLS-DA and SVM models was easily affected by the data type, sample size and other factors, and the overall identification ability of both models was not as good as the ResNet model based on synchronous 2DCOS images. Different from the previous single theory or single model analysis, this study verified the superiority of deep learning model in the identification research of medicinal plant resources from the actual and multiple perspectives.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

JY: conceptualization, software, formal analysis, writing–original draft preparation, and writing—review and editing. WL: methodology, resources, and software. YW: supervision, project administration, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

FUNDING

This work was supposed by National Natural Science Foundation of China (Grant Number: 31860584), the Reserve Talents of Young and Middle-Aged Academic Leaders in Yunnan Province (Grant Number: 202005AC160032), and the Major Science and Technology Projects in Yunnan Province of "Digital Development and Application of Biological Resources" (Grant Number: 202002AA100007).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.752863/full#supplementary-material

REFERENCES

Chen, J. B., Wang, Y., Rong, L. X., and Wang, J. J. (2018). Integrative two-dimensional correlation spectroscopy (i2DCOS) for the intuitive identification of adulterated herbal materials. *J. Mol. Struct.* 1163, 327–335. doi: 10.1016/j.molstruc.2018.02.061

Cunningham, A. B., Brinckmann, J. A., Bi, Y. F., Pei, S. J., Schippmann, U., Luo, P., et al. (2018). Paris in the spring: a review of the trade, conservation and opportunities in the shift from wild harvest to cultivation of Paris polyphylla (*Trilliaceae*). *J. Ethnopharmacol.* 222, 208–216. doi: 10.1016/j.jep.2018.04.048

Deng, Q., Lang, T., and Xia, J. X. (2016). Present situation and development of medicinal plant resources' utilization. *J. MUC* 25, 55–59.

Dong, J. E., Wang, Y., Zuo, Z. T., and Wang, Y. Z. (2020). Deep learning for geographical discrimination of Panax notoginseng with directly near-infrared spectra image. *Chemometr. Intell. Lab. Syst*. 197:103913. doi: 10.1016/j.chemolab.2019.103913

Feng, L. L., Zhang, L., Li, H. F., and Zhang, C. G. (2015). Quality evaluation of Paris polyphylla var. yunnanensis and accumulation law analysis of its steroidal saponins. *Chin. J. Exp. Tradit. Med. Formul.* 21, 41–45. doi: 10.13422/j.cnki.syfjx.2015130041

Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* 127, 418–424. doi: 10.1016/j.compag.2016.07.003

Houssein, E. H., Emam, M. M., Ali, A. A., and Suganthan, P. N. (2021). Deep and machine learning techniques for medical imaging-based breast cancer: a comprehensive review. *Expert Syst. Appl.* 167:114161. doi: 10.1016/j.eswa.2020.114161

Huang, H., Malkov, S., Coleman, M., and Painter, P. (2003). Application of two-dimensional correlation infrared spectroscopy to the study of miscible polymer blends. *Macromolecules* 36, 8156–8163. doi: 10.1021/ma0259463

Jamshidi-Kia, F., Lorigooini, Z., and Amini-Khoei, H. (2018). Medicinal plants: past history and future perspective. *J. Herbmed Pharmacol.* 7, 1–7. doi: 10.15171/jhp.2018.01

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

Li, J. R., Sun, S. Q., Wang, X. X., Xu, C. H., Chen, J. B., Zhou, Q., et al. (2014). Differentiation of five species of Danggui raw materials by FTIR combined with 2D-COS IR. *J. Mol. Struct.* 1069, 229–235. doi: 10.1016/j.molstruc.2014.03.067

Liu, L., Zuo, Z. T., Wang, Y. Z., and Xu, F. R. (2020). A fast multi-source information fusion strategy based on FTIR spectroscopy for geographical authentication of wild Gentiana rigescens. *Microchem. J.* 159:105360. doi: 10.1016/j.microc.2020.105360

Liu, Z. M., Yang, M. Q., Zuo, Y. M., Wang, Y. Z., and Zhang, J. Y. (2021). Fraud detection of herbal medicines based on modern analytical technologies combine with chemometrics approach: a review. *Crit. Rev. Anal. Chem.* [Epub Online ahead of print]. doi: 10.1080/10408347.2021.1905503

Newman, D. J., and Cragg, G. M. (2015). Natural products as sources of new drugs from 1981 to 2014. *J. Nat. Prod.* 79, 629–661. doi: 10.1021/acs.jnatprod.5b01055

Noble, W. S. (2006). What is a support vector machine? *Nat. Biotechnol.* 24, 1565–1567. doi: 10.1038/nbt1206-1565

Noda, I. (1989). Two-dimensional infrared spectroscopy. *J. Am. Chem. Soc.* 111, 8116–8118. doi: 10.1021/ja00203a008

Noda, I. (1990). Two-dimensional infrared (2D IR) spectroscopy: theory and applications. *Appl. Spectrosc.* 44, 550–561. doi: 10.1366/0003702904087398

Noda, I. (1993). Generalized two-dimensional correlation method applicable to infrared, raman, and other types of spectroscopy. *Appl. Spectrosc.* 47, 1329–1336. doi: 10.1366/0003702934067694

Noda, I. (2004). Advances in two-dimensional correlation spectroscopy. *Vib. Spectrosc.* 36, 143–165. doi: 10.1016/j.vibspec.2003.12.016

Noda, I. (2014). Frontiers of two-dimensional correlation spectroscopy. Part 1. New concepts and noteworthy developments. *J. Mol. Struct.* 1069, 3–22. doi: 10.1016/j.molstruc.2014.01.025

Noda, I. (2016). Techniques useful in two-dimensional correlation and codistribution spectroscopy (2DCOS and 2DCDS) analyses. *J. Mol. Struct.* 1124, 29–41. doi: 10.1016/j.molstruc.2016.01.089

Noda, I. (2018). Two-trace two-dimensional (2T2D) correlation spectroscopy-a method for extracting useful information from a pair of spectra. *J. Mol. Struct.* 1160, 471–478. doi: 10.1016/j.molstruc.2018.01.091

Obaid, H. S., Dheyab, S. A., and Sabry, S. S. (2019). "The Impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning," in *9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference*, (Jaipur, India: IEEE).

Pang, X. H., Song, J. Y., Zhu, Y. J., Xu, H. X., Huang, L. F., Chen, S., et al. (2011). Applying plant DNA barcodes for Rosaceae species identification. *Cladistics* 27, 165–170. doi: 10.1111/j.1096-0031.2010.00328.x

Pasquini, C. (2018). Near infrared spectroscopy: a mature analytical technique with new perspectives - a review. *Anal. Chim. Acta* 1026, 8–36. doi: 10.1016/j.aca.2018.04.004

Pei, Y. F., Zhang, Q. Z., and Wang, Y. Y. (2020). Application of authentication evaluation techniques of ethnobotanical medicinal plant genus Paris: a review. *Crit. Rev. Anal. Chem.* 50, 405–423. doi: 10.1080/10408347.2019.1642734

Pei, Y. F., Zhang, Q. Z., Zuo, Z. T., and Wang, Y. Z. (2018). Comparison and identification for rhizomes and leaves of Paris yunnanensis based on Fourier transform mid-infrared spectroscopy combined with chemometrics. *Molecules* 23:3343. doi: 10.3390/molecules23123343

Pei, Y. F., Zuo, Z. T., Zhang, Q. Z., and Wang, Y. Z. (2019). Data fusion of Fourier transform mid-infrared (MIR) and near-infrared (NIR) spectroscopies to identify geographical origin of wild Paris polyphylla var. yunnanensis. *Molecules* 24:2559. doi: 10.3390/molecules24142559

Shen, T., Yu, H., and Wang, Y. Z. (2020). Discrimination of Gentiana and its related species using IR spectroscopy combined with feature selection and stacked generalization. *Molecules* 25:1442. doi: 10.3390/molecules25061442

Sun, S. Q., Zhou, Q., and Qin, Z. (2003). *Atlas Of Two-Dimensional Correlation Information Spectroscopy For Traditional Chinese Medicine Identification*. Beijing: Chemical Industry Press.

Tao, A. E., Zhao, F. Y., Li, R. S., Qian, J. F., and Xia, C. L. (2020). Industrialization condition and development strategy of Paridis Rhizoma. *Chin. Tradit. Herb. Drugs* 51, 4809–4815. doi: 10.7501/j.issn.0253-2670.2020.18.026

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

Wang, Y., Huang, H. Y., and Wang, Y. Z. (2020). Authentication of Dendrobium Officinale from similar species with infrared and ultraviolet-visible spectroscopies with data visualization and mining. *Anal. Lett.* 53, 1774–1793. doi: 10.1080/00032719.2020.1719126

Wu, X. M., Zhang, Q. Z., and Wang, Y. Z. (2019). Traceability the provenience of cultivated Paris polyphylla Smith var. ynnanensis using ATR-FTIR spectroscopy combined with chemometrics. *Spectrochim. Acta A Mol. Biomol. Spectrosc.* 212, 132–145. doi: 10.1016/j.saa.2019.01.008

Wu, Z., Zhang, J., Zuo, Z. T., Xu, F. R., Wnag, Y. Z., Zhang, J. Y., et al. (2018). Rapid discrimination of the different processed Paris poly phylla var. yunnanensis with infrared spectroscopy combined with chemometrics. *Spectrosc. Spect. Anal.* 38, 1101–1106. doi: 10.3964/j.issn.1000-0593201804-1101-06

Yang, Y. G., and Wang, Y. Z. (2018). Characterization of Paris polyphylla var. yunnanensis by infrared and ultraviolet spectroscopies with chemometric data fusion. *Anal. Lett.* 51, 1730–1742. doi: 10.1080/00032719.2017.1385618

Yang, Y. G., Zhao, Y. L., Zuo, Z. T., and Wang, Y. Z. (2019). Determination of total flavonoids for Paris Polyphylla var. yunnanensis in different geographical origins using UV and FT-IR spectroscopy. *J. AOAC Int.* 102, 45 7–464.

Zhao, F. Y., Tao, A. E., Guan, X., Qian, J. F., and Xia, C. L. (2021). Research progress on chemical constituents, pharmacological effects and resource utilization modes of non-medicinal parts of Paridis Rhizoma. *Chin. Tradit. Herb. Drugs* 52, 2449–2457. doi: 10.7501/j.issn.0253-2670.2021.08.030

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: 28 September 2021
doi: 10.3389/fpls.2021.723294

Identification of Apple Leaf Diseases by Improved Deep Convolutional Neural Networks With an Attention Mechanism

Peng Wang[1,2,3], *Tong Niu*[1,2,3], *Yanru Mao*[1,2,3], *Zhao Zhang*[1,2,3], *Bin Liu*[2,3,4]* and *Dongjian He*[1,2,3]*

[1] College of Mechanical and Electronic Engineering, Northwest A&F University, Xianyang, China, [2] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Xianyang, China, [3] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Services, Xianyang, China, [4] College of Information Engineering, Northwest A&F University, Xianyang, China

OPEN ACCESS

Edited by:
Marciel Stadnik,
Federal University of Santa Catarina, Brazil

Reviewed by:
Hilman Pardede,
Indonesian Institute of Sciences, Indonesia
Leonardo Araujo,
Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina, Brazil

***Correspondence:**
Bin Liu
liubin0929@nwsuaf.edu.cn
Dongjian He
hdj168@nwsuaf.edu.cn

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: 10 June 2021
Accepted: 23 August 2021
Published: 28 September 2021

Citation:
Wang P, Niu T, Mao Y, Zhang Z, Liu B and He D (2021) Identification of Apple Leaf Diseases by Improved Deep Convolutional Neural Networks With an Attention Mechanism. Front. Plant Sci. 12:723294. doi: 10.3389/fpls.2021.723294

The accurate identification of apple leaf diseases is of great significance for controlling the spread of diseases and ensuring the healthy and stable development of the apple industry. In order to improve detection accuracy and efficiency, a deep learning model, which is called the Coordination Attention EfficientNet (CA-ENet), is proposed to identify different apple diseases. First, a coordinate attention block is integrated into the EfficientNet-B4 network, which embedded the spatial location information of the feature by channel attention to ensure that the model can learn both the channel and spatial location information of important features. Then, a depth-wise separable convolution is applied to the convolution module to reduce the number of parameters, and the h-swish activation function is introduced to achieve the fast and easy to quantify the process. Afterward, 5,170 images are collected in the field environment at the apple planting base of the Northwest A&F University, while 3,000 images are acquired from the PlantVillage public data set. Also, image augmentation techniques are used to generate an Apple Leaf Disease Identification Data set (ALDID), which contains 81,700 images. The experimental results show that the accuracy of the CA-ENet is 98.92% on the ALDID, and the average F1-score reaches .988, which is better than those of common models such as the ResNet-152, DenseNet-264, and ResNeXt-101. The generated test dataset is used to test the anti-interference ability of the model. The results show that the proposed method can achieve competitive performance on the apple disease identification task.

Keywords: apple disease, CA-ENet, attention mechanism, CA block, diseases identification

INTRODUCTION

The apple industry is one of the most important fruit industries in China. However, the frequent occurrence of apple leaf diseases may seriously restrict the healthy and stable development of the apple industry. At present, the diseases of a large number of industrialized apple orchards mainly rely on human vision for recognition, which requires a high degree of reliance on disease experts. The identification task is huge, especially since the visual inspection of fruit farmers or experts is prone to misjudgment due to their subjective perception and visual fatigue, and it is difficult to meet the demand for high-precision identification for intelligent orchards (Dutot et al., 2013). The problems previously discussed will lead to a large lag in the tracking management process of orchard diseases, which causes the improper use of pesticides and reduces the quality of fruit. Therefore, the accurate identification of diseases is of great significance to improve the yield and quality of apples and to cultivate disease-resistant varieties.

With the development of computer vision, machine learning techniques have been widely used in the agricultural field in recent years, and a series of approaches have been achieved in crop disease identification (Aravind et al., 2018; Kour and Arora, 2019; Mohammadpoor et al., 2020). In recent years, the main techniques, which are widely used in crop disease identification include artificial neural network (ANN) (Sheikhan et al., 2012), the K Nearest Neighbors (KNN) algorithm (Guettari et al., 2016), random forests (RF) (Kodovsky et al., 2012), and so on. For example, Wang et al. (2019) proposed a method for identifying cucumber powdery mildew based on a visible spectrum by extracting the spectral features and training a Support Vector Machine (SVM) classifier to establish a classification model, optimizing the radial basis kernel function, and the recognition accuracy of the method reached 98.13%. In contrast, Prasad et al. (2016) proposed a mobile client-server architecture for leaf disease detection and diagnosis based on the combination of a Gabor Wavelet Transform (GWT) and a Gray-Level Co-occurrence Matrix (GLCM). The mobile terminal captures the object image and then transmits it to the server after pre-processing. The server then performs GWT-GLCM feature extraction and classification based on the KNN algorithm. The system can monitor farmland information through the mobile terminal at any stage. Although the previously discussed studies achieved outstanding performances in disease identification tasks, the low-level feature representations extracted from them are limited to intuitive shallow features, such as the colors, textures, and shapes of the images. Thus, it is difficult to achieve competitive performance on apple leaf disease identification tasks.

Compared with machine learning algorithms that require cumbersome image pre-processing and feature extraction (Kulin et al., 2018; Zhang et al., 2018b), convolutional neural networks (CNNs) can directly learn robust high-level feature representations of apple diseases from images. The extracted high-level feature representation is richer and better compared with the method of manually extracting features; therefore, CNNs have achieved excellent results in multiple visual tasks (Ren et al., 2017; Liu et al., 2018; Bi et al., 2020). In recent years, with the continuous emergence of advanced deep learning architectures such as the ResNet (He et al., 2016), ResNeXt (Xie et al., 2017), and DenseNet (Huang et al., 2017), the recognition accuracy and speed are constantly being refreshed on the public dataset, ImageNet. In order to solve the problem of the mobile deployment of the model, scholars have proposed various lightweight architectures, such as Xception (Chollet, 2017), MobileNet (Howard et al., 2017; Sandler et al., 2018), ShuffleNet (Ma et al., 2018; Zhang et al., 2018a), and so on. In order to provide a stable, efficient, low-cost, and highly intelligent disease identification method, Chao et al. (2020) proposed that the XDNet combined with DenseNet and Xception can enhance the feature extraction capability of the model. The model achieved an accuracy of 98.82% in identifying five apple leaf diseases with fewer parameters. Liu et al. (2020) adopted the Inception structure and introduced a dense connection strategy to build a new neural network model, which realized the real-time and accurate identification of six different kinds of grape leaf diseases. In addition, Ramcharan et al. (2019) deployed a trained cassava disease recognition model for a mobile terminal. Tests under natural conditions in the field found that complex conditions, such as different angles, brightness, and the occlusion of the image taken, could adversely affect the performance of the model, which also proves that image classification under the complex background of the field is challenging.

An attention mechanism can provide a novel solution for feature extraction. The attention mechanism can assign larger weights to regions of interest and smaller weights to backgrounds and extract information that contributes more to classification to optimize the model and to make judgments that are more accurate. In other studies, attention mechanisms have achieved excellent performance in tasks, such as classification, detection, and segmentation (Hu et al., 2018; Karthik et al., 2020; Mi et al., 2020; Hou et al., 2021). Inspired by the above researches, this study proposes a new CNN for apple diseases recognition. The main contributions and innovations of this study are summarized as follows:

1. A new Apple Leaf Disease Identification Data set (ALDID) is generated by using image generation techniques. In order to enhance the generalization performance of the model, image augmentation techniques are used to expand the data set and simulate apple leaf disease images collected under different conditions, laying a foundation for the training of the model.
2. A novel attention-based apple leaf disease recognition model, namely, the Coordination Attention EfficientNet (CA-ENet), is proposed. A network search technique is first used to determine the optimal structure of the model, and the optimal parameters of network depth, width, and input image resolution are obtained. Then, the deep separable convolution is applied to the coordination attention convolution (CA-Conv) infrastructure to greatly reduce the number of parameters and avoid an overfitting problem. Finally, a coordinated attention block is embedded in the infrastructure to realize the integration of characteristic channel information and spatial information attention and to strengthen the learning ability of the model for important information in the lesion area.

The remainder of the study is organized as follows: In section Materials and Methods, the detailed information of the dataset is introduced and expanded by data augmentation techniques. The model proposed in this study and the related content of attention visualization is introduced in detail. The section Results and Discussion presents the experiments for evaluating the performance of the model and analyzes the results of the experiments, discussed the impact of data augmentation and external interference on the performance of the model. The last section, Conclusion and Future Work, summarizes the work of this study and prospects for further research.

MATERIALS AND METHODS

This section introduces the materials and methods used in the study in detail, including the collected apple diseased leaf images and the ALDID established after augmentation. It also presents the proposed model and the attention visualization method.

Image Acquisition

The study was conducted from July 2020 to October 2020, at the apple planting experimental station of the Northwest A&F University in Qianxian County, Shaanxi province. By using a variety of different types of mobile devices, a huge number of field environment apple leaf images under different angles and distances are collected. There are a total of 5,170 disease images with a resolution of 3,000 × 3,000 pixels, including those of five species of the Glomerella leaf spot (*Colletotrichum fructicola*), Apple leaf mites (*Panonychus ulmi*), Mosaic (Apple mosaic virus), Apple litura moth (*Spodoptera litura Fabricius*), and Healthy leaves. In addition, 3,000 disease images under a single background of three kinds of laboratories, namely, Black rot (*Physalospora obtuse*), Scab (*Venturia inaequalis*), and Rust (*Gymnosporangium yamadai*), were collected from the public dataset PlantVillage. The above two data sets are shuffled and mixed to generate the original data set of common apple diseases.

Figure 1 shows random samples of each category in the data set. There are a large number of complex background images in the data set. At the same time, it can be seen that Apple litura moth (G) and Apple leaf mites (H) leaves have relatively similar geometric features. The difference between the two diseases can be expressed as a fine-grained image classification problem. A variety of different forms of samples can increase the diversity of the data set, making it closer to various different situations that may occur in the real situation. However, it also constitutes a greater test for the image classification task and puts forward higher requirements for the comprehensive performance of the model.

Image Augmentation

When acquiring the apple disease images, the samples obtained varied in the apple leaf growth position, weather condition, shooting angle, and there are interference factors such as equipment noise. In order to enable the model to learn as many irrelevant patterns as possible and avoid overfitting problems, the images of the dataset need to be expanded and normalized.

In the data expansion, Gaussian blurring, contrast enhancement by 30% and decrease by 30%, and brightness enhancement by 30% and decrease by 30% are adopted to simulate different weather conditions for all samples of the original dataset. The images are also rotated by 90°, 270°, a horizontal flip, and a vertical flip to simulate the change of shooting angle, then the original data set is added. A Mosaic disease image is randomly selected to enhance and display the

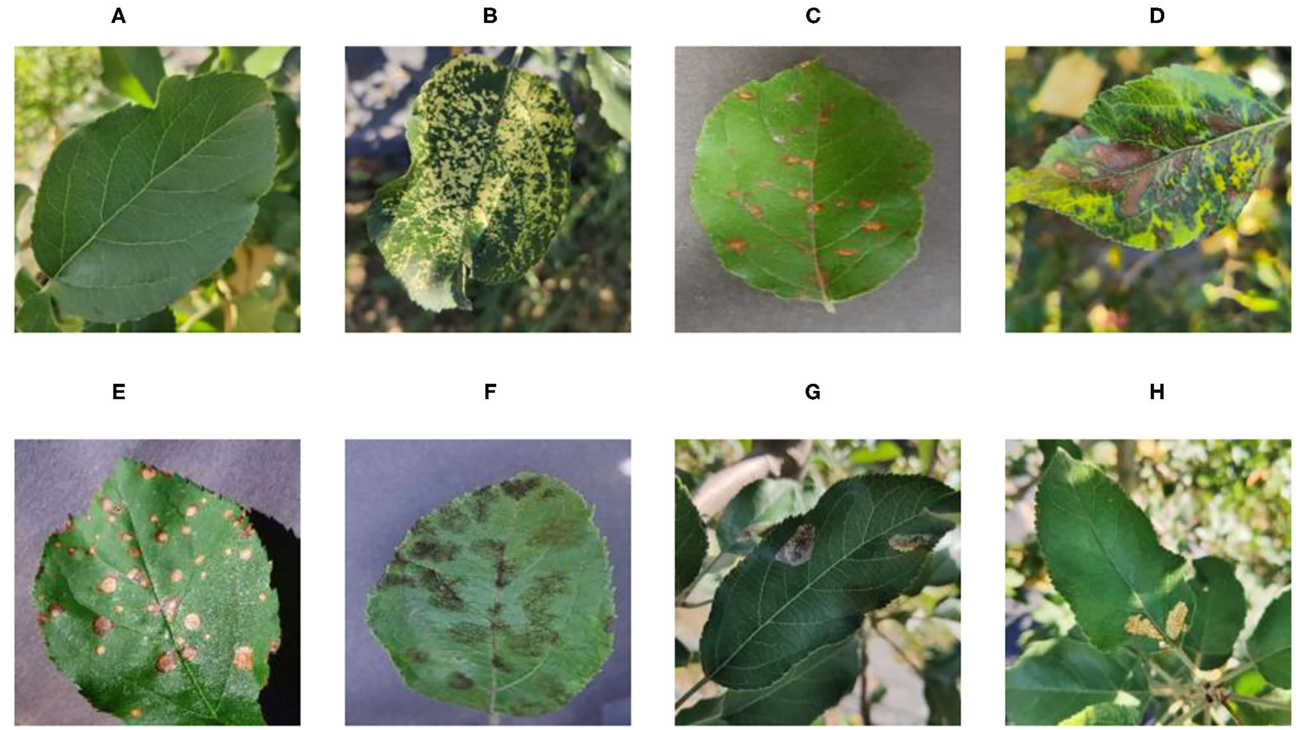


FIGURE 1 | Eight common apple leaf disease types. **(A)** Healthy, **(B)** Mosaic, **(C)** Rust, **(D)** Glomerella leaf spot, **(E)** Black rot, **(F)** Scab, **(G)** Apple litura moth, and **(H)** Apple leaf mites.

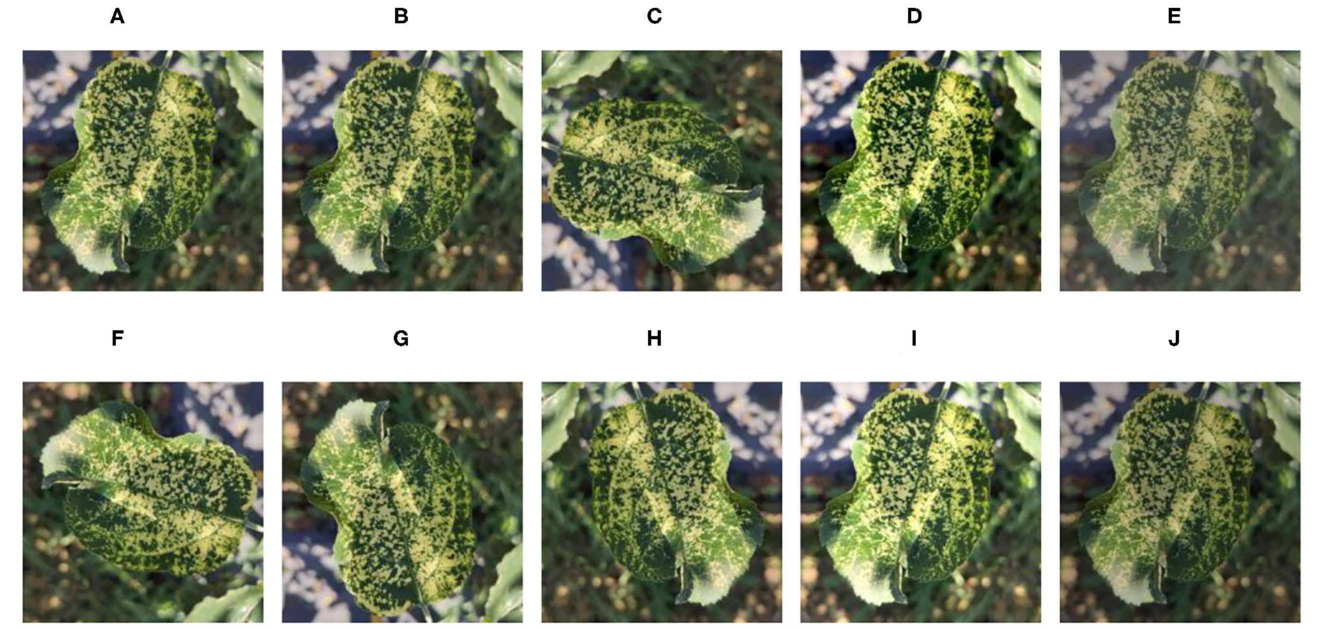


FIGURE 2 | Image enhancement example of the mosaic disease. **(A)** Original image, **(B)** Gaussian blur, **(C)** 90° rotation, **(D)** High contrast, **(E)** Low contrast, **(F)** 270° rotation, **(G)** Horizontal symmetry, **(H)** Vertical symmetry, **(I)** High brightness, and **(J)** Low brightness.

effect as shown in **Figure 2**. **Table 1** represents the structure information of the ALDID. It can be seen from **Table 1** that the sample distribution is balanced after image expansion, which is in line with the actual application scenario. It can ensure that the model extracts different features of each category in a balanced manner, ensuring its correct training and avoiding overfitting. This study also divides the ALDID according to the ratio of training set: validation set = 4:1 for model training and validation. The training set is used to train the model, and the validation set is used to check whether the model training process converges normally and whether there is an overfitting problem.

During the training process, a large fluctuation of the feature value range will affect the convergence of the model, which is not conducive to the model learning different feature differences, and the images need normalization. In order to test the stability of the model, 500 images were randomly selected from each type of disease image in the original data set, and a total of 4,000 images were selected from eight different diseases. After scrambling these 4,000 images, five different interference factors, namely, Gaussian noise, salt and pepper noise, 180° rotation, 30% sharpness enhancement, and 30% sharpness reduction were randomly added, and a Model Robustness Test Data set (MRTD) was generated. After the training process is completed, the MRTD is then used to test the model to verify the effect of the model training. The above work laid the foundation for the use of the model.

CA-ENet Network

The existing CNN methods of increasing network depth, width, and input image resolution can obtain richer and higher fine-grained features, but, there will be serious problems such as gradient disappearance and model degradation. The problem is that only changing a single variable cannot achieve better results. The basic network architecture EfficientNet-B0 (Tan and Quoc, 2019), which uses neural architecture search (NAS) techniques to optimize the above three factors at the same time, balances the three dimensions of depth, width, and resolution, and can be further adjusted by the scaling factor. Therefore, in this study, we use the EfficientNet architecture as the feature extraction network.

Different types of apple leaf diseases have different morphological characteristics with regard to lesions, but there is a high degree of similarity between certain types of diseases, which means apple disease classification can be viewed as a fine-grained image classification problem, and existing models still have difficulty achieving satisfactory results. Therefore, in order to enhance model effectiveness, attention to the lesion area is the key to solving this problem. The widely used channel attention mechanism, SENet (Hu et al., 2018), has a significant effect on improving final performance, but this operation ignored the location information of the features, which is also important for generating spatial selective attention maps. In order to identify these differences, the CA-ENet is proposed to achieve real-time and accurate apple disease identification. The overall structure of the model is shown in **Figure 3**.

The model mainly included three parts: the pre-network for the Batch Normalization of input images, the backbone network CA-Conv for feature extraction, and the rear part that outputs the recognition result through the fully connected layer. Pre-network uses a layer of 3 × 3 ordinary convolutions with a step of 1 to perform the convolution operation on the input image, the input image resolution is 380 × 380, and the feature map with the depth

TABLE 1 | The composition of apple leaf disease identification data set (ALDID).

Category	Healthy	Mosaic	Rust	Glomerella leaf spot	Black rot	Scab	Apple litura moth	Apple leaf mites
Training	8,240	8,240	7,000	8,360	7,000	7,000	8,200	8,320
Validation	2,060	2,060	2,000	2,090	2,000	2,000	2,050	2,080
Total	10,300	10,300	10,000	10,450	10,000	10,000	10,250	10,400

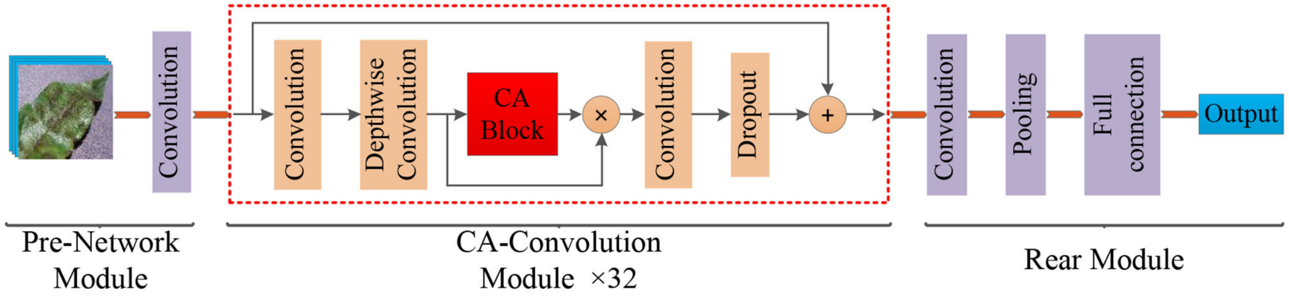


FIGURE 3 | Structure of the Coordination Attention EfficientNet (CA-ENet) for apple disease identification.

of the output feature matrix of 48 is obtained. Then, the obtained feature matrix are input into the 32 CA-Conv module embedded with the CA block. Finally, the 3 × 3 ordinary convolutions and pooling are used to further abstract features and then output through a fully connected layer with eight nodes.

During the model optimization process, a NAS technique is used to search for the optimal model structure. The operation process can be abstractly expressed as Equation (1):

$$\mathbf{N\left(d,w,r\right)=\odot_{i=1,2,\ldots,s}F_i^{L_i}\left(X_{(H_i,W_i,C_i)}\right)} \tag{1}$$

where $\odot$ is the multiplication symbol. F_i^{Li} means arithmetic operation, it is repeatedly executed L_i times in the operation F_i. X is the input feature matrix. (H_i, W_i, *and* C_i) represents the height, width, and output channels of X. The NAS process can be optimized by adding the constraints of model accuracy, parameter, and calculation amount with Equations (2) and (5).

$$\max_{(\mathbf{d,w,r})}\left[\mathbf{Accuracy(N(d,w,r))}\right] \tag{2}$$

$$\mathbf{N\left(d,w,r\right)=\odot_{i=1,2,\ldots,s}\hat{F}_i^{d\cdot\hat{L}_i}(X_{(r\cdot\hat{H}_i,r\cdot\hat{W}_i,r\cdot\hat{C}_i)})} \tag{3}$$

$$\mathbf{Memory(N)\leq tar\_memory} \tag{4}$$

$$\mathbf{FLOPs(N)\leq tar\_flops} \tag{5}$$

The *d*, *w*, and *r* are the sparseness that scales the depth, width, and resolution of the network, respectively, the *tar_memory* and *tar_flops* are the constraints on the number of parameters and calculations. Through the above optimization calculation, the best *d*, *w*, and *r* values of the EfficientNet-B0 structure can be obtained, and on this basis, the magnification factors *d* and *w* of EfficientNet-B4 are 1.8 and 1.4, respectively, and the input image resolution *r* is 380 × 380 pixels. From the discussed method, the optimal CA-ENet structure parameters can be calculated and are shown in **Table 2**.

The operators in **Table 2** perform arithmetic operations on the input features. The magnification of each CA-Conv6 in Stage 3–Stage 8 is 6; that is, in the first layer of convolution, the depth of the feature matrix of the input layer is increased to 6 times of the input, and the size of the convolution kernel is 3× 3 or 5 × 5. The resolution, output channels, and repeat correspond to the resolution of the input layer, the depth of the output feature matrix, and the number of repetitions of the layer structure in the depth direction. The steps given by first-stride are only for the first layer structure of each stage, and the steps of the other layer structures are all 1. The network is composed of seven-stage CA-Conv blocks, and its structure is shown in **Figure 4**. First, the input feature matrix is sent to CA-Conv through an ordinary 1 × 1 convolution for dimension upgrade. After the h-swish activation function, the feature is extracted through the deep separable convolution with a convolution kernel size of k × k (k = 3 or 5) and a step of 1 or 2. The use of a deep separable convolution structure greatly reduces the number of model parameters, and at the same time, can play an important role in avoiding model overfitting. Then, the obtained feature matrix is divided into two branches, one of which is assigned a weight to each channel by a Coordinate Attention Block (CAB), and another one without any processing is multiplied by the two weights passed through the CAB to obtain the weighted feature matrix. Finally, the dimension is reduced by 1 × 1 convolution and output to the subsequent structure after adding with the input feature matrix.

The global pooling method can compress the global spatial information into the channel descriptor, but this results in a lack of location information. In order to capture the precise location information of the features, in the CAB in **Figure 4**, the

TABLE 2 | Details about coordination attention EfficientNet (CA-ENet).

Stage	Operator	Resolution	Output channels	Repeat	First-stride
1	Conv, 3 × 3	380 × 380	48	1	2
2	CA-Conv1, 3 × 3	190 × 190	24	2	1
3	CA-Conv6, 3 × 3	190 × 190	32	4	2
4	CA-Conv6, 5 × 5	95 × 95	56	4	2
5	CA-Conv6, 3 × 3	48 × 48	112	6	2
6	CA-Conv6, 5 × 5	24 × 24	160	6	1
7	CA-Conv6, 5 × 5	24 × 24	272	8	2
8	CA-Conv6, 3 × 3	12 × 12	448	2	1
9	Conv, 3 × 3	12 × 12	1,792	1	1
10	Avg Pooling, 1 × 1	12 × 12	1,792	1	1
11	fc	1 × 1 × 1,792	8	1	1

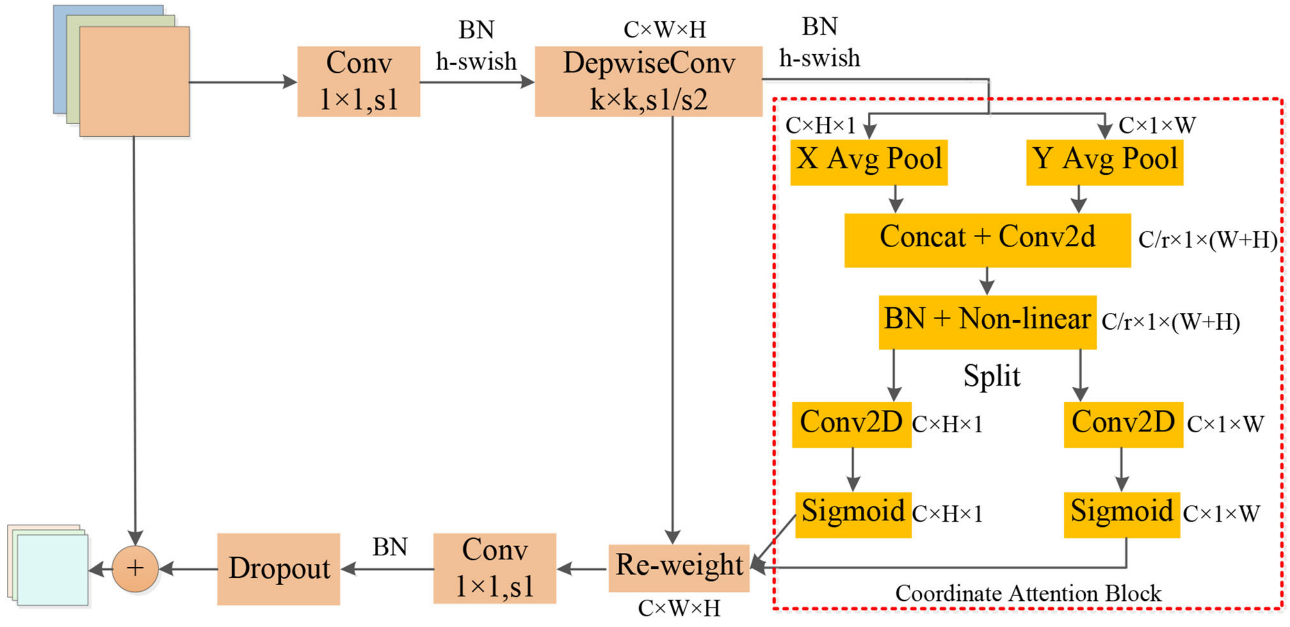


FIGURE 4 | Structure of coordination attention convolutional (CA-Conv).

global pooling is decomposed into two one-dimensional feature encoding processes according to Equation (6). Furthermore, two one-dimensional average pooling operations along the horizontal and vertical directions are used to aggregate the input features into two separate direction-aware feature maps. This operation captures both direction-aware and position-sensitive information, thus enabling the model to locate the region of interest more accurately. The generated two separate direction-aware feature maps are concatenated in the depth direction, and the feature channel attention weight is generated through a 1 × 1 convolution compression channel, and the position information is embedded in the channel attention. Then, the Batch Nomalization (BN) operation is applied to the feature matrix and divided into two parts through a non-linear activation function, the feature depth is adjusted to be consistent with the input feature through 1 × 1 convolution, and the position information is saved in the generated attention map. Finally, the weights of the two attention maps are multiplied by the input features to strengthen the feature representation of the attention region and improve the ability of the network to locate the regions of interest accurately.

$$\mathbf{z_c}=\frac{\mathbf{1}}{\mathbf{H\times W}}\sum_{\mathbf{i=1}}^{\mathbf{H}}\sum_{\mathbf{i=1}}^{\mathbf{W}}\mathbf{x_c(i,j)} \tag{6}$$

As the above-mentioned information embedding method can directly obtain the global receptive field and encode the accurate position information, so the transformation operation is performed on it using the 1 × 1 convolution transformation function F_1. As shown in Equation (7), $[z^h, z^w]$ is the splicing operation along a spatial dimension, δ is the non-linear activation function, and f is the intermediate feature map that encodes the spatial information in both horizontal and vertical directions. Then, through two 1 × 1 convolutions, f^h and f^w are transformed into tensors with the same number of channels, respectively. As shown in Equations (8) and (9), attention weights can be calculated, and the output of the CA block after the Re-weight is calculated by Equation (10).

$$\mathbf{f}{=}\delta(\mathbf{F_1}(\left[\mathbf{z^h},\mathbf{z^w}\right])) \tag{7}$$

$$\mathbf{g^h}{=}\sigma(\mathbf{F_h}(\mathbf{f^h})) \tag{8}$$

$$\mathbf{g^w}{=}\sigma(\mathbf{F_w}(\mathbf{f^w})) \tag{9}$$

$$\mathbf{y_c}(\mathbf{i},\mathbf{j})\ {=}\mathbf{x_c}(\mathbf{i},\mathbf{j}){\times}\mathbf{g_c^h}(\mathbf{i}){\times}\mathbf{g_c^w}(\mathbf{i}) \tag{10}$$

In order to reduce the amount of calculation and speed up reasoning while ensuring the effect of the activation function, a new activation function, h-swish, is applied into CA-Conv (Howard et al., 2019). The activation functions of sigmoid and h-sigmoid are shown in Equations (11) and (12). It can be seen from **Figure 5** that the above two activation functions are relatively close and the calculation process of h-sigmoid is more concise, so h-sigmoid can be used to replace sigmoid in Equations (13) and (14). **Figure 6** shows the approximation of the effect of h-swish on the swish activation function. It can be seen that the two curves are basically the same, and the calculation speed of the h-swish is faster.

$$\mathbf{sigmiod}\ (\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{1+e^{-x}}} \tag{11}$$

$$\mathbf{h} - \mathbf{sigmiod}\ (\mathbf{x}) = \frac{\mathbf{relu6(x+3)}}{\mathbf{6}} \tag{12}$$

$$\mathbf{swish}\ (\mathbf{x})\ {=}\mathbf{x} \cdot \mathbf{sigmoid}(\mathbf{x}) \tag{13}$$

$$\mathbf{h} - \mathbf{swish}\ (\mathbf{x})\ {=}\mathbf{x}{\cdot}(\mathbf{h} - \mathbf{sigmoid}) \tag{14}$$

EXPERIMENTAL RESULTS AND DISCUSSION

Model Training Details

In order to verify the performance of the proposed method, a proposed network is trained *via* the ALDID. Thus, the proposed method is realized on the Pytorch 1.7.1 deep learning framework, while all experiments were conducted on an Intel® Xeon(R) Gold 5217 CPU@3.00 GHz server equipped with an NVIDIA Tesla V100 (32GB) GPU. The operating system is Ubuntu 18.04.5 LTS 64. In order to accelerate the model convergence while keeping stable training, the initial learning rate is set to .01, and it decays according to the cosine learning rate change curve during the training process, and finally decays to .001. The number of training iterations for all models is 50 epochs.

Performance of Proposed CA-ENet

In order to evaluate the performance of the proposed method, multiple state-of-the-art methods were applied to the MRTD. In order to ensure that the results are comparable, the same training strategy was used. The test result is visually displayed with a confusion matrix. In order to facilitate the display of labels, the full names of some diseases are abbreviated. In this case, "GLS" in the confusion matrix stands for Glomerella leaf spot, "ALM" stands for Apple litura moth, and "ALMS" stands for Apple leaf mites.

Figure 7 can intuitively show the classification performance of the Coordination Attention EfficientNet, with the final accuracy reaching 98.92%. The misclassification mainly occurred between Apple leaf mites and Apple litura moth and between Apple litura

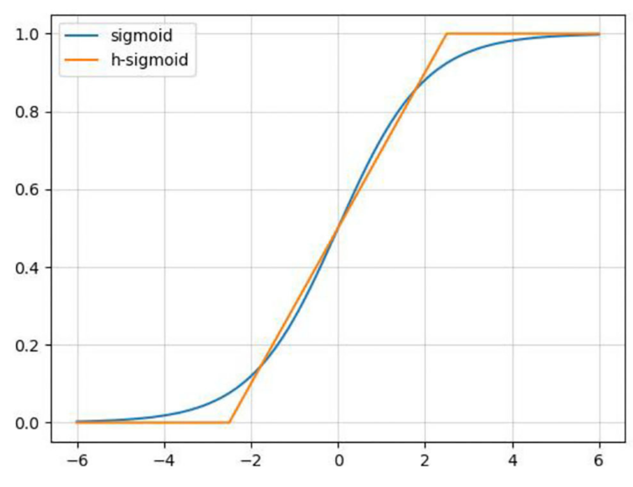


FIGURE 5 | Schematic diagram of the sigmoid and h-sigmoid activation functions.

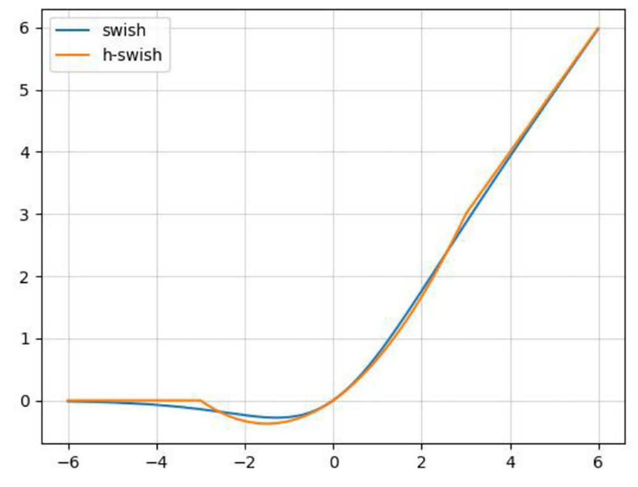


FIGURE 6 | Schematic diagram of the swish and h-swish activation functions.

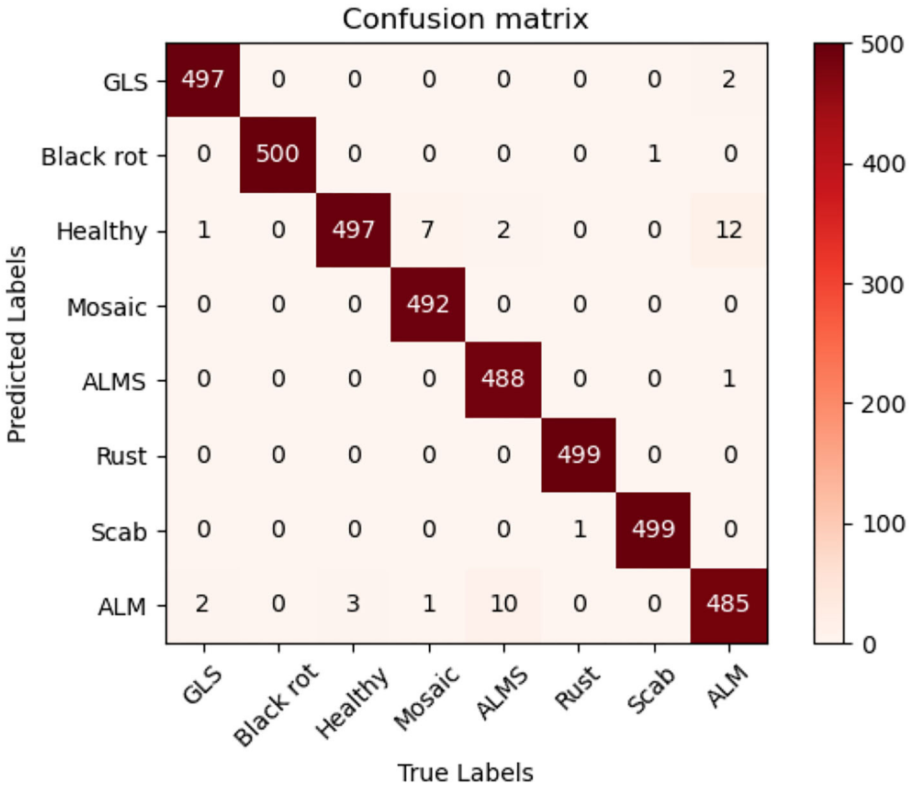


FIGURE 7 | Confusion matrix of the CA-ENet.

moth and Healthy leaves. The main feature of the apple leaf mites is that the damaged leaves show many dense chlorosis gray-white spots. In contrast, after being damaged by apple litura moth, the insect spots formed on the leaves were elliptical and dense, and the leaf surface was wrinkled. The above two kinds of leaf spots have certain similarities in geometric and color characteristics, leading to misjudgment. Furthermore, affected by the complex background, a small number of leaves damaged by apple litura moths were mistakenly identified as healthy leaves. It can be seen that accurate recognition in a complex background has been a great challenge, but the number of misjudgments in this model is still within an acceptable range and can be maintained at a low level. The proposed CA-Conv structure can extract richer fine-grained features of the image and perceive the regions of interest with a higher degree of attention. It can also be seen that the model shows a good recognition effect and has strong robustness to the problem of apple leaf disease recognition.

Performance Comparison

The performance comparison between CA-ENet and the standard method is shown in **Table 3**. It can be seen from **Table 3** that the proposed model has the best recognition performance on MRTD, with an accuracy of 98.92%. In this study, multiple metrics including accuracy, precision, recall, F1-score, parameter, and calculation are used as evaluation indicators. ResNet-152 takes advantage of the residual structure to make sure it has a strong feature learning ability, so it can reach an accuracy of 93.75%. The Dense Block, the basic structure of DenseNet-264, also has the advantages of enhanced feature propagation and incentive feature reuse, making it achieve a higher accuracy rate with nearly half of the parameters of ResNet-152. Furthermore, the accuracy of ResNeXt-101 reaches 95.67%, which is due to the use of grouped convolution, so it can achieve better results with fewer convolutional layers than ResNet-152. Although this structure can improve the final accuracy, the degree of network fragmentation is very high due to the existence of a large number of parallel branches, which greatly reduces the computation efficiency of the model.

EfficientNet uses NAS techniques to simultaneously search and optimize model depth, width, and input image resolution, and rationally expand the model architecture to achieve a high degree of coordination of structural proportions. It has obvious advantages in extracting more robust and reliable feature representations and can reach an overall accuracy of 97.27%. The strong learning ability of the CA module in CA-Conv may cause attention drift and affect model convergence, while the inverted residual structure in CA-Conv can suppress features that are not conducive to classification, ensuring model stability while further improving the recognition performance, and the effectiveness of the attention mechanism is verified.

Traditional CNNs do not distinguish the importance of information when extracting disease features, and there is a large number of convolutions that repeatedly extract low-contribution information, which causes a waste of computation resources. The attention mechanism can automatically extract high-contribution feature components, with only small parameters

TABLE 3 | Performance comparison of the CA-ENet with other classical networks.

	Accuracy/%	Average precision/%	Average recall/%	Average F1-score	Params/M	FLOPs/B
ResNet-152	93.75	94.01	93.75	0.936	60	11.0
DenseNet-264	94.90	95.27	94.90	0.949	34	6.0
ResNeXt-101	95.67	96.05	95.67	0.957	84	32.0
EfficientNet-B4	97.27	97.41	97.27	0.972	19	4.2
CA-ENet	**98.92**	**98.95**	**98.92**	**0.988**	21	4.3

Bold values indicate the best results under each index.

TABLE 4 | Performance of the CA-ENet before and after data augmentation.

Dataset	Metrics	Glomerella leaf spot	Black rot	Healthy	Mosaic	Apple leaf mites	Rust	Scab	Apple litura moth
Original	Precision/%	96.9	93.5	99.1	95.0	96.1	100.0	100.0	96.0
		99.6	99.8	95.8	100.0	99.8	100.0	99.8	96.8
	Recall/%	99.0	100.0	89.6	99.4	99.2	99.7	92.8	95.8
ALDID		99.4	100.0	99.4	98.4	97.6	99.8	99.8	97.0
	F1-score	0.979	0.966	0.941	0.972	0.976	0.998	0.963	0.959
		0.995	**0.989**	**0.976**	**0.992**	**0.987**	**0.999**	**0.998**	**0.969**

Bold values indicate the best results under each index.

and calculations increases. The experimental results also show that, in the identification of apple leaf diseases, the proposed CA-ENet model is superior to other models in all evaluation indicators with fewer parameters and can classify apple disease images more accurately.

Effect of Data Augmentation on Identification Performance for Each Class

A variety of data expansion methods is used in the ALDID to improve the anti-interference ability of the model in complex situations and prevent the problem of overfitting. In order to verify the effect of data augmentation, a set of comparative experiments is designed to evaluate its impact on the final classification performance. **Table 4** shows the accuracy, recall, and F1-score performance indicators of the proposed model for each category on the MRTD. The first row of values in each performance index is the performance obtained after training on the original dataset, and the second row of values is the performance obtained after training on the ALDID. It can be seen from **Table 4** that the image diversity of the original data set is insufficient, and the average F1-score of the proposed method on the original dataset is 0.969, which is slightly lower than the performance of the model obtained on the ALDID, but it can still accurately classify apple leaf diseases. The results show that the augmented data set is closer to the actual situation, the ability of the model to adapt to complex scenes is enhanced, and the anti-interference ability is improved to a certain extent. The leverage of the deep separable convolution can effectively reduce the number of model parameters and greatly increase training speed.

Feature and Network Attention Visualization

Understanding and analyzing the hidden layer structure of the model is an important method to comprehensively recognize the proposed network structure. CNNs are usually trained in the form of black-box testing and the evaluations of model performance are limited to the final accuracy and other indicators, which have certain deficiencies. Visualization techniques are the way to explore how CNNs learn features and distinguish categories. So, this section uses the visualization of layer activation and class activation heatmaps to analyze the performance of the proposed model. The visualization of layer activation helps to understand how the continuous convolutional layer performs feature extraction and completes the conversion of input features. **Figures 8**, **9** show the output features of the first 20 channels of the CA-Conv structure in the first and last layers of the model, respectively. The given example category is apple leaf mites. In the superficial features of the model, it is obvious that the lesion area and the background are separated, and the characteristics of the disease location can be accurately extracted. The model has high efficiency in extracting deep features and only contains a few failed convolutions. The channel output features given here are all valid. Therefore, the stacking of the CA-Conv structure does not affect feature learning ability of the model, and the adopted separable convolution can effectively reduce the feature redundancy and lead to higher efficiency.

Class Activation Mapping (CAM) (Selvaraju et al., 2020) helps to understand which feature components the model relies on to make decisions. **Table 5** shows the original image of the class activation and the attention heatmaps of the commonly used models. The sample images of Glomerella leaf spot, black rot, apple litura moth, and rust are randomly selected for testing. Due to the introduction of the attention module CAB, CA-ENet has a stronger ability to focus on the lesion area. Compared with other models, CA-ENet has a good positioning effect and can accurately locate the interest area, whether it is a leaf lesion in a complex or a simple background. In contrast, ResNet-152, DenseNet-264, and ResNeXt-101 have deviations or even errors in their focus positions, which are what affect the robustness and

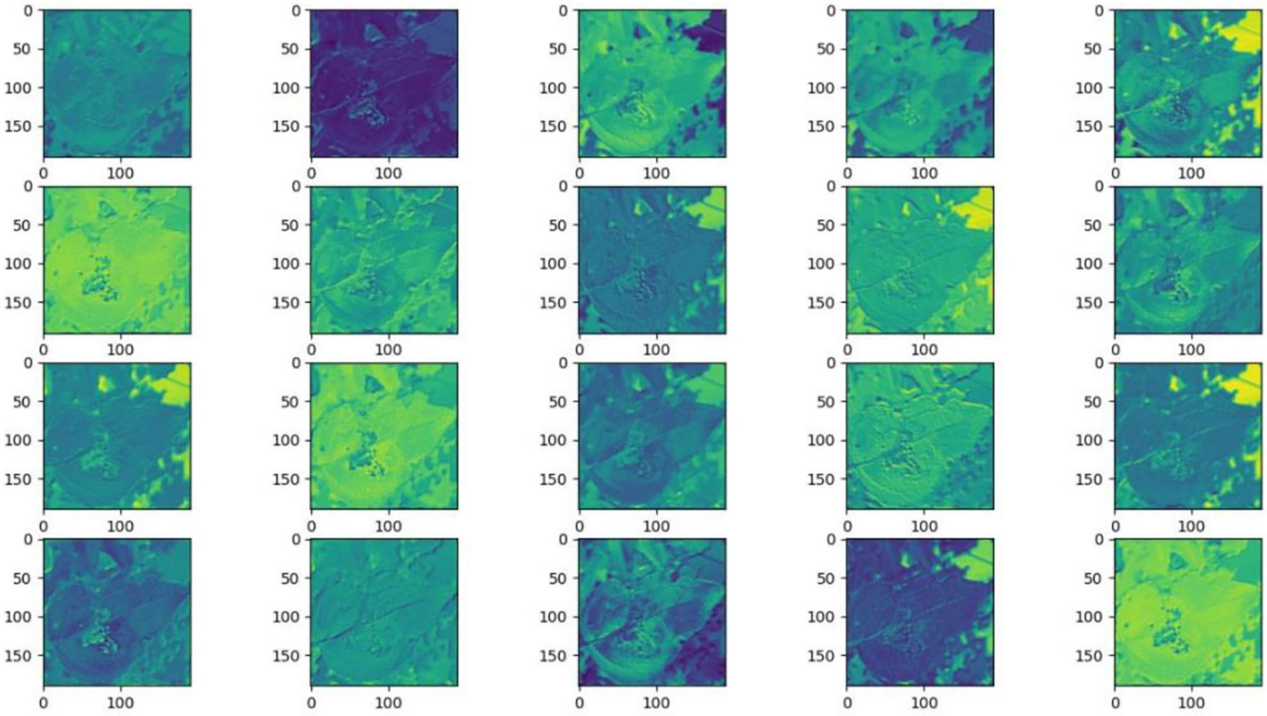


FIGURE 8 | Partial output feature maps of the first CA-Conv.

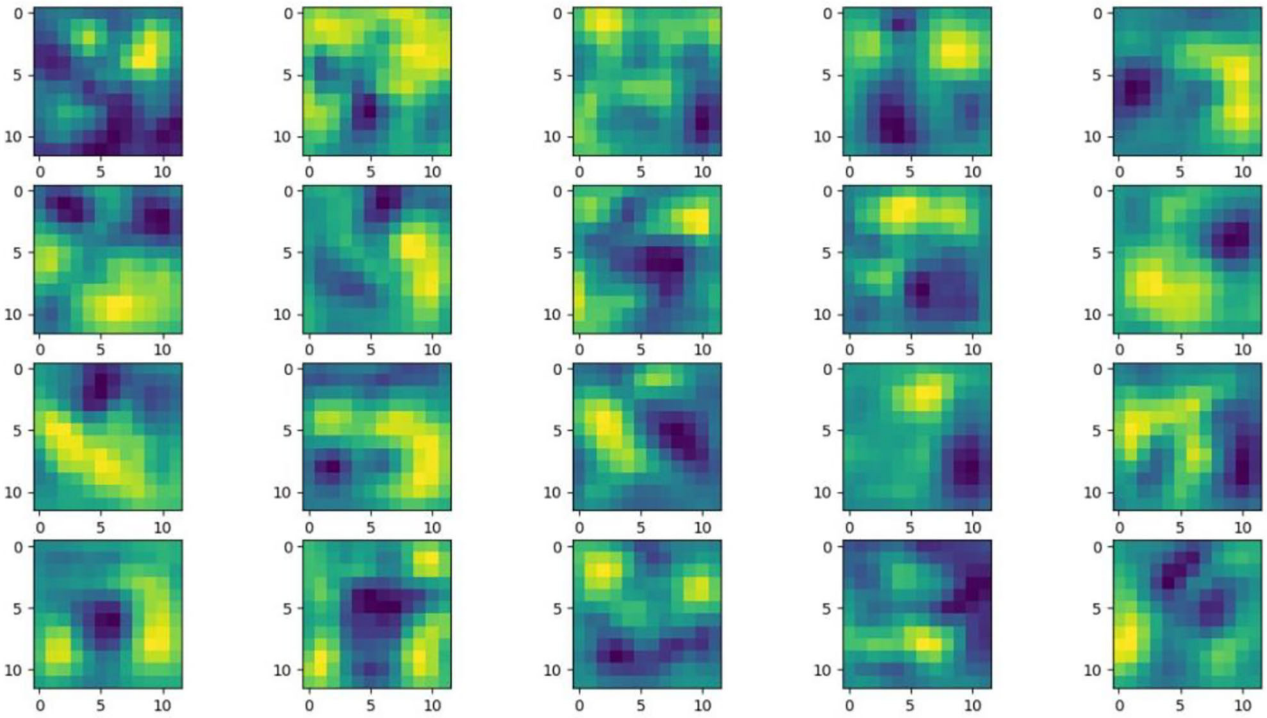
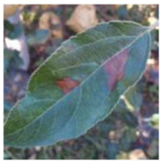
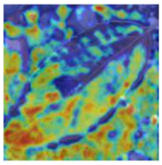
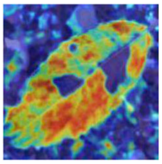
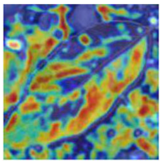
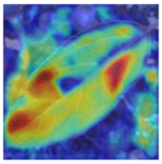
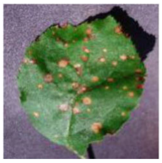
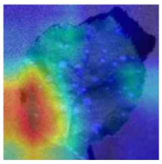
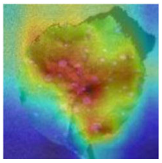
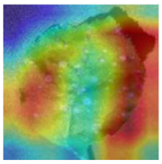
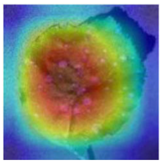
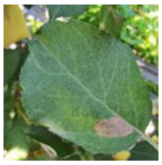
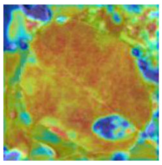
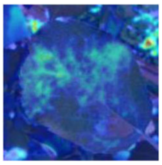
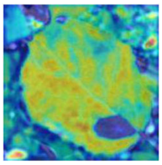
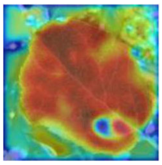
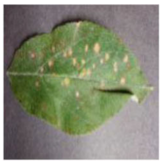
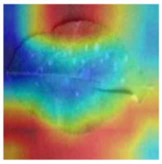
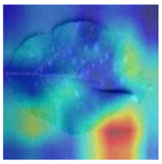
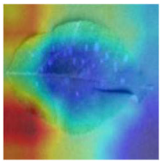
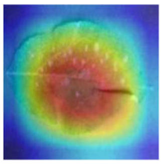


FIGURE 9 | Partial output feature maps of the last CA-Conv.

TABLE 5 | Comparison of attention heatmaps of different models.

Class	Original image	ResNet-152	DenseNet-264	ResNeXt-101	CA-ENet
Glomerella leaf spot					
Black rot					
Apple litura moth					
Rust					

accuracy of a model. The visual test results of the class activation heatmaps of the apple leaf diseases show that the model fully takes the characteristics of the disease spots into account and achieves superior recognition performance on apple leaf diseases.

CONCLUSION AND FUTURE WORK

An improved attention-based deep CNN to identify common apple leaf diseases to support the efficient management of orchards is proposed in this study. Due to the complex environment of orchards, in order to be close to the real application scenarios, 5,170 apple leaf images were collected by multiple mobile devices and 3,000 disease images were obtained from a public dataset. Image augmentation techniques are used to generate the ALDID containing 81,700 diseased images. By embedding a CA block into a CA-Conv module, the integration of characteristic channel and location information was realized. A deep separable convolution is also used to reduce the number of parameters, and the h-swish activation function is used to speed up the model convergence. The proposed model is training with ALDID and testing with MRTD and conducts a large number of comparative experiments including various performance evaluation indicators and process visualizations. The experimental results show that the method proposed in this study achieves a recognition accuracy of 98.92%, which is better than that of other existing deep learning methods and achieves competitive performance on apple leaf disease identification tasks, which provides a reference for the application of deep learning methods in crop disease classification. The proposed model has the advantages of a simple structure, fast running speed, good generalization performance, and robustness, and has great potential application value. In the future, a ground mobile inspection platform equipped with cameras will be built to replace manual operations and to realize the rapid diagnosis and early warning of apple diseases.

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

PW designed and performed the experiment, selected the algorithm, analyzed the data, trained the algorithms, and wrote the manuscript. PW, TN, YM, and ZZ collected data. BL monitored the data analysis. DH conceived the study and participated in its design. All authors contributed to the article and approved the submitted version.

FUNDING

This research is sponsored by the Key Research and Development Program of Shaanxi (2021NY-138 and 2019ZDLNY07-06-01), by CCF-Baidu Open Fund (NO. 2021PP15002000).

ACKNOWLEDGMENTS

We would like to thank the reviewers for their valuable suggestions on this manuscript.

REFERENCES

Aravind, K. R., Raja, P., Mukesh, K. V., Aniirudh, R., Ashiwin, R., and Szczepanski, C. (2018). "Disease classification in maize crop using bag of features and multiclass support vector machine," in *2nd International Conference on Inventive Systems and Control* (Coimbatore), 1191–1196. doi: 10.1109/ICISC.2018.8398993

Bi, C., Wang, J., Duan, Y., Fu, B., Kang, J., and Shi, Y. (2020). MobileNet based apple leaf diseases identification. *Mobile Netw. Appl*. doi: 10.1007/s11036-020-01640-1

Chao, X., Sun, G., Zhao, H., Li, M., and He, D. (2020). Identification of apple tree leaf diseases based on deep learning models. *Symmetry* 12:1065. doi: 10.3390/sym12071065

Chollet, F. (2017). Xception: deep learning with depthwise separable convolutions. *IEEE Conf. Comput. Vision Pattern Recogn*. 1800–1807. doi: 10.1109/CVPR.2017.195

Dutot, M., Nelson, L., and Tyson, R. (2013). Predicting the spread of postharvest disease in stored fruit, with application to apples. *Postharvest Biol. Technol.* 85, 45–56. doi: 10.1016/j.postharvbio.2013.04.003

Guettari, N., Capelle-Laizé, A. S., and Carré, P. (2016). Blind image steganalysis based on evidential k-nearest neighbors. *IEEE Int. Conf. Image Process*. 2742–2746. doi: 10.1109/ICIP.2016.7532858

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conf. Comput. Vision Pattern Recogn*. 770–778. doi: 10.1109/CVPR.2016.90

Hou, Q., Zhou, D., and Feng, J. (2021). Coordinate attention for efficient mobile network design. *arXiv [Preprint]*. arXiv:2103.02907v1.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., et al. (2019). Searching for MobileNetV3. *IEEE Int. Conf. Comput. Vision.* 1314–1324. doi: 10.1109/ICCV.2019.00140

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., and Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv [Preprint]*. arXiv:1704.04861.

Hu, J., Shen, L., Albanie, S., and Sun, G. (2018). Squeeze-and-excitation networks. *IEEE Trans. Pattern. Anal*. 7132–7141. doi: 10.1109/CVPR.2018.00745

Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. (2017). Densely connected convolutional networks. *IEEE Conf. Comput. Vision Pattern Recogn*. 2261–2269. doi: 10.1109/CVPR.2017.243

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

Kodovsky, J., Fridrich, J., and Holub, V. (2012). Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Foren Sec.* 7, 432–444. doi: 10.1109/tifs.2011.2175919

Kour, V. P., and Arora, S. (2019). Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants. *IEEE Access* 7, 29374–29385. doi: 10.1109/ACCESS.2019.2901900

Kulin, M., Kazaz, T., Moerman, I., and Poorter, E. D. (2018). End-to-end learning from spectrum data: a deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE Access* 6, 18484–18501. doi: 10.1109/ACCESS.2018.2818794

Liu, B., Ding, Z., Tian, L., He, D., Li, S., and Wang, H. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Front. Plant Sci* 11:1082. doi: 10.3389/fpls.2020.01082

Liu, B., Zhang, Y., He, D., and Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* 10:11. doi: 10.3390/sym10010011

Ma, N., Zhang, X., Zheng, H. T., and Sun, J. (2018). Shufflenet v2: practical guidelines for efficient CNN architecture design. *arXiv [Preprint].* arXiv:1807.11164v1.

Mi, Z., Zhang, X., Su, J., Han, D., and Su, B. (2020). Wheat stripe rust grading by deep learning with attention mechanism and images from mobile devices. *Front. Plant Sci.* 11:558126. doi: 10.3389/fpls.2020.558126

Mohammadpoor, M., Nooghabi, M. G., and Ahmedi, Z. (2020). An intelligent technique for grape fanleaf virus detection. *Int. J. Interact. Multimedia Artif. Intell.* 6, 62–67. doi: 10.9781/ijimai.2020.02.001

Prasad, S., Peddoju, S. K., and Ghosh, D. (2016). Multi-resolution mobile vision system for plant leaf disease diagnosis. *Signal Image Video Process.* 10, 379–388. doi: 10.1007/s11760-015-0751-y

Ramcharan, A., McCloskey, P., Baranowski, K.,Mbilinyi, N., Mrisho, L., and Ndalahwa, M., et al. (2019). A mobile-based deep learning model for cassava disease diagnosis. *Front. Plant Sci.* 10:272. doi: 10.3389/fpls.2019.00272

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.* 39, 1137–1149. doi: 10.1109/tpami.2016.2577031

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). MobileNetV2: inverted residuals and linear bottlenecks. *IEEE Conf. Comput. Vision Pattern Recogn*. 4510–4520. doi: 10.1109/CVPR.2018.00474

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra., D. (2020). "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–626. doi: 10.1109/ICCV.2017.74

Sheikhan, M., Pezhmanpour, M., and Moin, M. S. (2012). Improved contourlet-based steganalysis using binary particle swarm optimization and radial basis neural networks. *Neural Comput. Appl.* 21, 1717–1728. doi: 10.1007/s00521-011-0729-9

Tan, M., and Quoc, V. L. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. *arXiv [Preprint].* arXiv:1905.11946.

Wang, X., Zhu, C., Fu, Z., Zhang, L., and Li, X. (2019). Research on cucumber powdery mildew recognition based on visual spectra. *Spectrosc. Spectr. Anal.* 39, 1864–1869. doi: 10.3964/j.issn.1000-0593 (2019)06-1864-06

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. *IEEE Conf. Comput Vision Pattern. Recogn*. 5987–5995. doi: 10.1109/CVPR.2017.634

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018a). ShuffleNet: an extremely efficient convolutional neural network for mobile devices. *IEEE Conf. Comput. Vision Pattern. Recogn*. 6848–6856. doi: 10.1109/CVPR.2018.00716

Zhang, Y., Gravina, R., Lu, H. M., Villari, M., and Fortino, G. (2018b). PEA: parallel electrocardiogram-based authentication for smart healthcare systems. *J. Netw. Comput. Appl.* 117, 10–16. doi: 10.1016/j.jnca.2018.05.007

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: 26 November 2021
doi: 10.3389/fpls.2021.702626

Unmanned Aerial System-Based Weed Mapping in Sod Production Using a Convolutional Neural Network

Jing Zhang[1], Jerome Maleski[1], David Jespersen[2], F. C. Waltz Jr.[2], Glen Rains[3] and Brian Schwartz[1]*

[1] Department of Crop and Soil Sciences, University of Georgia, Tifton, GA, United States, [2] Department of Crop and Soil Sciences, University of Georgia, Griffin, GA, United States, [3] Department of Entomology, University of Georgia, Tifton, GA, United States

OPEN ACCESS

Edited by:
Kioumars Ghamkhar,
AgResearch Ltd., New Zealand

Reviewed by:
Shangpeng Sun,
McGill University, Canada
Michael Gomez Selvaraj,
Consultative Group on International Agricultural Research (CGIAR),
United States

***Correspondence:**
Jing Zhang
jingzhang687@gmail.com

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: 29 April 2021
Accepted: 25 October 2021
Published: 26 November 2021

Citation:
Zhang J, Maleski J, Jespersen D, Waltz FC Jr, Rains G and Schwartz B (2021) Unmanned Aerial System-Based Weed Mapping in Sod Production Using a Convolutional Neural Network. Front. Plant Sci. 12:702626. doi: 10.3389/fpls.2021.702626

Weeds are a persistent problem on sod farms, and herbicides to control different weed species are one of the largest chemical inputs. Recent advances in unmanned aerial systems (UAS) and artificial intelligence provide opportunities for weed mapping on sod farms. This study investigates the weed type composition and area through both ground and UAS-based weed surveys and trains a convolutional neural network (CNN) for identifying and mapping weeds in sod fields using UAS-based imagery and a high-level application programming interface (API) implementation (Fastai) of the PyTorch deep learning library. The performance of the CNN was overall similar to, and in some classes (broadleaf and spurge) better than, human eyes indicated by the metric recall. In general, the CNN detected broadleaf, grass weeds, spurge, sedge, and no weeds at a precision between 0.68 and 0.87, 0.57 and 0.82, 0.68 and 0.83, 0.66 and 0.90, and 0.80 and 0.88, respectively, when using UAS images at 0.57 cm–1.28 cm pixel^{-1} resolution. Recall ranges for the five classes were 0.78–0.93, 0.65–0.87, 0.82–0.93, 0.52–0.79, and 0.94–0.99. Additionally, this study demonstrates that a CNN can achieve precision and recall above 0.9 at detecting different types of weeds during turf establishment when the weeds are mature. The CNN is limited by the image resolution, and more than one model may be needed in practice to improve the overall performance of weed mapping.

Keywords: Bermudagrass, artificial intelligence, Fastai, ResNet, RGB imagery

INTRODUCTION

Weeds are a persistent problem on sod farms. Herbicides to control different weed species are one of the largest chemical inputs (Satterthwaite et al., 2009; Wojciech and Landry, 2009; Yi, 2012) and often their control requires multiple applications throughout the growing season. A variety of annual and perennial broadleaf and grassy weeds are usually present in Georgia sod farms including annual bluegrass (*Poa annua*), goosegrass (*Eleusine indica*), crabgrass (*Digitaria* spp.), dallisgrass (*Paspalum dilatatum*), sedges (*Cyperus* spp.), spurge (*Euphorbia* spp.), chickweed (*Stellaria media* L.), and pigweed (*Amaranthus* spp.) (Colvin et al., 2013). Regulations limiting the broadcast application of certain chemicals in sod production (USEPA, 2009), due to concerns about the environmental impacts of the herbicide, create difficulty in effectively controlling weeds. Aside from the environmental cost of herbicides, there are significant financial costs in purchasing the herbicide and the labor and fuel used in application. Site-specific weed management, such as applying herbicides only where the weeds are located, instead of whole-field broadcast applications would significantly reduce herbicide use, thereby improving economic and environmental sustainability in sod production. The presence of weeds negatively affects turfgrass certification programs by increasing inspection times of sod that is being guaranteed as weed-free and uniform before being sold to consumers for uses such as sports fields, golf courses, and home lawns. Thus, the ability to quickly identify and respond to areas with weed issues is an attractive proposition for both sod growers and inspection agencies.

One of the key components for site-specific weed management is the generation of a weed map. Recent technical advances in unmanned aerial systems (UAS) have allowed for fast image acquisition and weed mapping using UAS in crops such as sunflower (*Helianthus* spp.) (Torres-Sánchez et al., 2013), cotton (de Castro et al., 2018), and rice (*Oryza sativa*) (Huang et al., 2018; Stroppiana et al., 2018). In these field crops, weed mapping was often conducted early in the growing season before canopy closure (Torres-Sánchez et al., 2013; López-Granados et al., 2016; Pérez-Ortiz et al., 2016; Stroppiana et al., 2018). Torres-Sánchez et al. (2013) evaluated image spatial and spectral properties for discriminating weeds in sunflower fields and reported adequate separation among weeds, crops, and bare soil using Excess Green Index, Normalized Green-Red Difference Index, and Normalized Difference Vegetation Index (NDVI) at a 30-m altitude. López-Granados et al. (2016) implemented object-based image analysis (OBIA) to extract the crop row and used both the relative position of vegetation to the crop row and spectral features to locate weeds. Successful late-season weed mapping using a UAS in oat fields was possible by taking advantage of greater spectral differences between oats and perennial weeds, as cereal crops become yellow during their senescence phase (Gašparović et al., 2020). Machine learning algorithms such as k-means clustering and random forest combined with OBIA were used for image classification.

However, only a few pieces of research have been conducted on how to best implement UAS-based weed mapping for sod production. Knowledge gained from the previous study of row crops is difficult to directly apply to turfgrass systems because they have unique challenges when it comes to weed mapping. First, there is no crop row in sod production to a pattern where the turfgrass should be. Second, regular mowing in sod production removes the morphological distinction of weeds, and it is not a common practice in other crops. Also, as a perennial crop, weeds are a year-round problem. Furthermore, and possibly most problematic, weed mapping in turfgrass production requires the differentiation of weeds against a green vegetation background instead of soil. Deep learning neural networks may be a good approach to address these challenges, and there is a growing set of literature developing weed image recognition models (Mahmudul Hasan et al., 2021). These often depend on high-resolution images of the weed leaf with or without background vegetation (Olsen et al., 2019; Espejo-Garcia et al., 2020; Hu et al., 2020). Yu et al. (2019a,b,c) reported several deep convolutional neural network (CNN) models that are exceptionally accurate (F1 score > 0.92, accuracy = 0.99) at detecting several broadleaf weeds in dormant and non-dormant Bermuda grass (*Cynodon* spp.) and perennial ryegrass (*Lolium perenne* L.) using images taken at the ground level (0.05 cm pixel^{-1}). The best-performing image classifiers for detecting three broadleaf types in active-growing Bermuda grass including *Hydrocotyle* spp., *Hedyotis corymbosa*, and *Richardia scabra* were trained using the architecture VGGNet consisting of 16 layers (Yu et al., 2019b). VGGNet is a CNN architecture proposed in 2014 (Simonyan and Zisserman, 2014). These previous examples exploited either very high-resolution images or distinct cropping system features to aid in identifying weeds.

There is a lack of information to quantify the potential savings of using site-specific weed management in sod production, which will likely be critical before end-users such as farmers and certification agencies adopt this new technology. Thus, the objectives of this study were (1) to investigate weed-type composition and distribution through both ground and UAS-based weed surveys on sod farms and (2) to assess the feasibility of training and using a CNN for weed mapping in sod fields using UAS-based imagery. Therefore, our hypotheses were that (1) the percentage of the area without weeds was high even in a weed-infested area from human eyes and (2) a CNN can be trained with reasonable performance to detect the generic type of weed in the sod production field.

MATERIALS AND METHODS

Ground Survey

Turfgrass weed surveys were carried out on sod production fields, on six different occasions during the growing season in 2019 and 2020 (**Table 1**). Ground weed surveys were conducted shortly after UAS flights for ground truth labeling of the images for deep learning. For the ground survey, a grid was laid on the area where UAS flew over with sizes ranging from 30 to 91 m squares using measuring tapes. Four ground targets were placed on the four corners of the whole grid to help generate shapefile later during the image process and labeling (**Figure 1**). The cell size of each grid was 1.5 m by 1.5 m. People who conducted the survey walked through the area in one direction, visually assessed, and recorded every 1.5 m on a notepad whether or not a certain type of weed was present, and then the measuring tape was moved down 1.5 m in the other direction. Broad categories of weeds included broadleaf, grass weeds, and sedge. The category "grass weeds" present in our study included crabgrass, goosegrass, and dallisgrass. In one of the surveys, spotted spurge (*Euphorbia maculata*) was present, and it was separated as a different category due to its purple leaves and stems and unique appearance after close mowing in the sod fields.

Unmanned Aerial Systems Survey

Unmanned aerial system flights were conducted using DJI Phantom 4 Pro V2 (DJI, Shenzhen, China) equipped with a 20 megapixel red, green, and blue (RGB) camera. The image

TABLE 1 | Summary of the training images from six surveys conducted on Georgia sod farms in 2019 and 2020.

Survey number	Turf species	Status of the field	Dominant weed types	Number of images
1	Bermuda grass	Establishing	Broadleaf, Sedge	3,570
2	Bermuda grass	Established	Broadleaf, Sedge, Crabgrass	1,600
3	Bermuda grass	Established	Broadleaf, Sedge, Crabgrass	1,600
4	Bermuda grass	Established	Broadleaf, Sedge	1,600
5	Bermuda grass	Establishing	Broadleaf, Goosegrass	530
6	Bermuda grass and Zoysia grass	Established	Broadleaf, Sedge, Spotted spurge	1,280

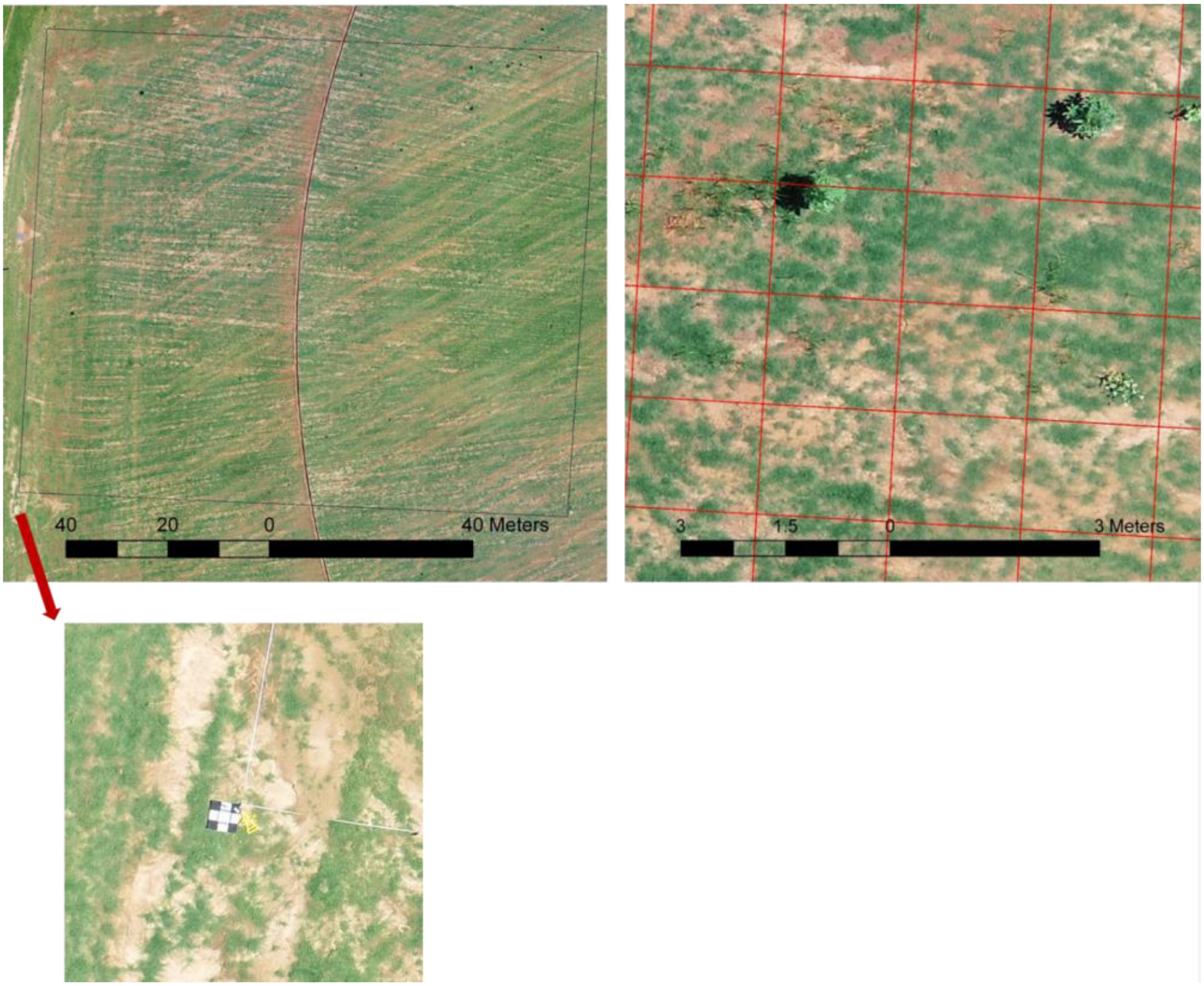


FIGURE 1 | Example of the survey area conducted on Georgia sod farm in 2019. Top left: overlook of the survey with the size of 91 m by 91 m outlined by the black box and the ground target placed on the southwest corner (bottom); top right: one small section of the survey with the grid overlaid.

resolution was 4,864 × 3,648. The flights were conducted at 75% side and front overlap, and the flight altitudes ranged from 20 to 40 m, resulting in ground sampling distances of 0.57 cm–1.28 cm pixel^{-1}. The flight times were between 10 a.m. and 4 p.m. with varying light conditions (clear, overcast, and partially cloudy). The flight plans were preprogrammed using DroneDeploy (DroneDeploy, Inc., San Francisco, CA, United States), which sets up the flight parameters such as path, altitude, and image overlap and sent out waypoints for autonomous flights.

Image Process and Labeling

Raw images were processed through Pix4DMapper (Pix4D SA, Lausanne, Switzerland), and orthomosaics were generated using a standard workflow template – "Ag RGB." The orthomosaic of each flight was further cropped into smaller images representing

1.5 m by 1.5 m cell size (**Figure 1**). Two main considerations were given to decide a proper cell size as follows: (1) 1.5 m by 1.5 m resulted in ∼200 pixels for each image which is needed to include important plant features and (2) the image size aligned with the ground survey cell size, which is practical for ground survey due to its intensiveness and time-consuming nature. The cropped images were labeled according to the ground survey results. Labels were divided into five classes including broadleaf, grass weeds, spotted spurge, sedge, and no weeds.

Training a Convolutional Neural Network

Fastai framework was chosen to train and validate the multi-label image classifier. Fastai is built on PyTorch and provides a high-level application programming interface (API), which implements many of the best practices from literature, allowing data practitioners to quickly create and train deep learning networks to achieve state-of-the-art results (Howard and Gugger, 2020). A multi-label method was used instead of an object detection method for the following reasons: sections of the field were targeted for treatment rather than individual plants, and the multi-label image classifier is lighter weight, less data, and process-intensive, and easier to train and implement because the drawing of bounding boxes is not required as in the object detection method.

More than 10,000 images from 6 surveys (**Table 1**) were used to train a CNN. The training was conducted under a Windows 10 operating system, and the graphics processing unit (GPU) card was NVIDIA Quadro P4000. The images were divided into training (80%) and validation (20%) datasets. The architecture used was ResNet 34. Another architecture ResNet 50 was also tested and yielded a deeper CNN, but no improvement of the performance on the validation dataset was found (**Supplementary Table 1**). The general workflow is illustrated in **Figure 2**, including data augmentation, image normalization, finding the learning rate (LR), and cycles of training from lower image size to higher image size. Through the process, approximately 20 epochs were trained with variable LRs. LR was determined using an LR finder (Learner.lr_find), which launches an LR range test to help the practitioner select a good LR. The test trains the model with exponentially growing LR, stops in case of divergence and then plots the losses vs. LR with a log scale. A good LR is when the slope is the steepest (Howard, 2020). The change of loss for training and validation datasets during four phases of training is included in **Supplementary Figure 1**.

Metrics, Thresholds, and Performance Comparison

The model output of the validation dataset is the number between zero and one indicating the confidence in the prediction for each class, the higher the number, the more probable the class. To assess precision vs. recall tradeoffs, a range of threshold values for accepting a positive result from the model between 0.2 and 0.5 was evaluated. The number of true positives (Tp), true negatives (Tn), false positives (Fp), and false negatives (Fn) were obtained. Metrics for accuracy, precision, and recall were further computed as follows (Sokolova and Lapalme, 2009):

Accuracy evaluates the average effectiveness of a classifier:

$$\text{Accuracy per class} = \frac{Tp + Tn}{Tp + Fp + Tn + Fn} \tag{1}$$

$$\text{Average Accuracy} = \frac{\sum_{i=1}^{l} Accuracy}{l} \tag{2}$$

Precision measures the number of correctly classified positive examples divided by the number of examples labeled by the system as positive:

$$\text{Precision per class} = \frac{Tp}{Tp + Fp} \tag{3}$$

$$\text{Average precision} = \frac{\sum_{i=1}^{l} Precision}{l} \tag{4}$$

Recall measures the number of correctly classified positive examples divided by the number of positive examples in the data:

$$\text{Recall per class} = \frac{Tp}{Tp + Fn} \tag{5}$$

$$\text{Average recall} = \frac{\sum_{i=1}^{l} Recall}{l} \tag{6}$$

In our use case, increasing precision will reduce the herbicide sprayed on non-weed areas, whereas increasing recall will ensure a more thorough control of the weeds (i.e., not missing any weeds). For sod growers using broadcast applications for weed control, emphasizing increased recall could enhance their confidence for early adoption of the technology. Thus, a metric Fbeta was used to evaluate the model by taking both the precision and recall into account using a single score (Sasaki, 2007):

$$\text{Fbeta} = \frac{\left(1 + beta^2\right) * Precision * Recall}{beta^2 * Precision + Recall} \tag{7}$$

$$\text{Average Fbeta} = \frac{\sum_{i=1}^{l} Fbeta}{l} \tag{8}$$

Beta = 2.0, referred to as the F2 score, is used to put more weight on recall than precision.

Metrics were computed separately for survey 1 and the other surveys. The field in survey 1 was under establishment, and the sod grower postponed herbicide application and mowing, resulting in relatively mature weeds. The larger weeds made for easier detection and better results. Metrics for survey 1 represent the case where weeds are relatively mature whereas the rest of

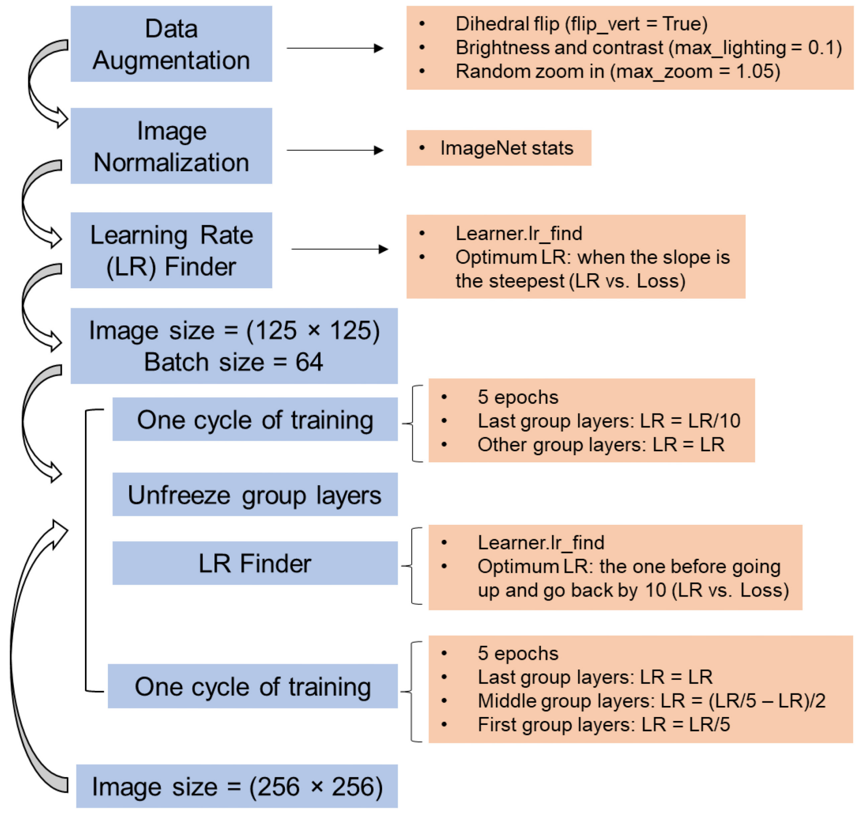

Data Augmentation
• Dihedral flip (flip_vert = True)
• Brightness and contrast (max_lighting = 0.1)
• Random zoom in (max_zoom = 1.05)
Image Normalization
• ImageNet stats
Learning Rate (LR) Finder
• Learner.lr_find
• Optimum LR: when the slope is the steepest (LR vs. Loss)
Image size = (125 × 125)
Batch size = 64
One cycle of training
• 5 epochs
• Last group layers: LR = LR/10
• Other group layers: LR = LR
Unfreeze group layers
LR Finder
• Learner.lr_find
• Optimum LR: the one before going up and go back by 10 (LR vs. Loss)
One cycle of training
• 5 epochs
• Last group layers: LR = LR
• Middle group layers: LR = (LR/5 – LR)/2
• First group layers: LR = LR/5
Image size = (256 × 256)

FIGURE 2 | Schematic workflow for training image classifier in Fastai.

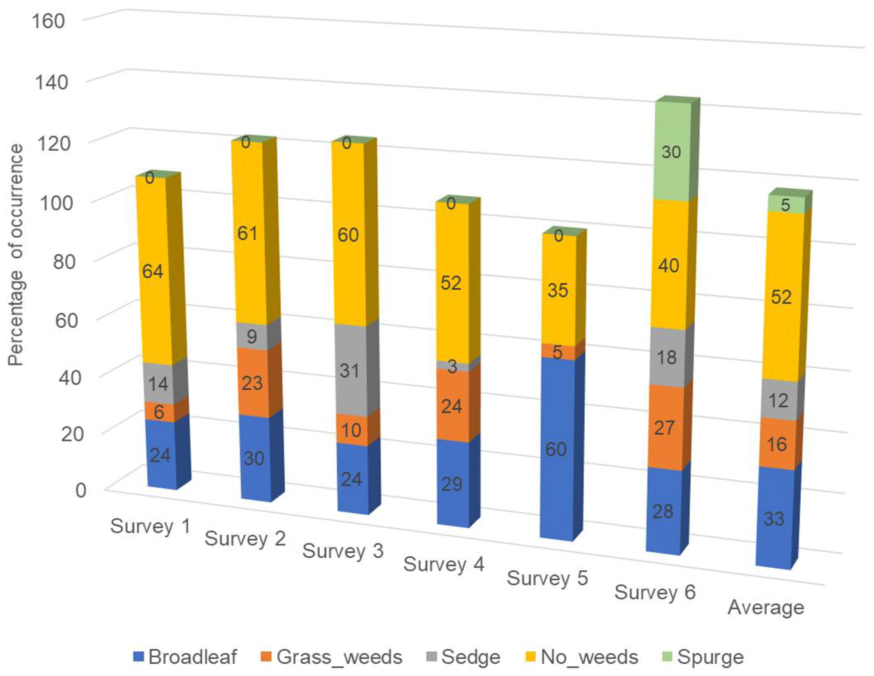


FIGURE 3 | Percentage of area (cells) with different weed types presented in six surveys corresponding to the surveys summarized in **Table 1**. All surveys were conducted on Georgia sod farms in 2019 and 2020.

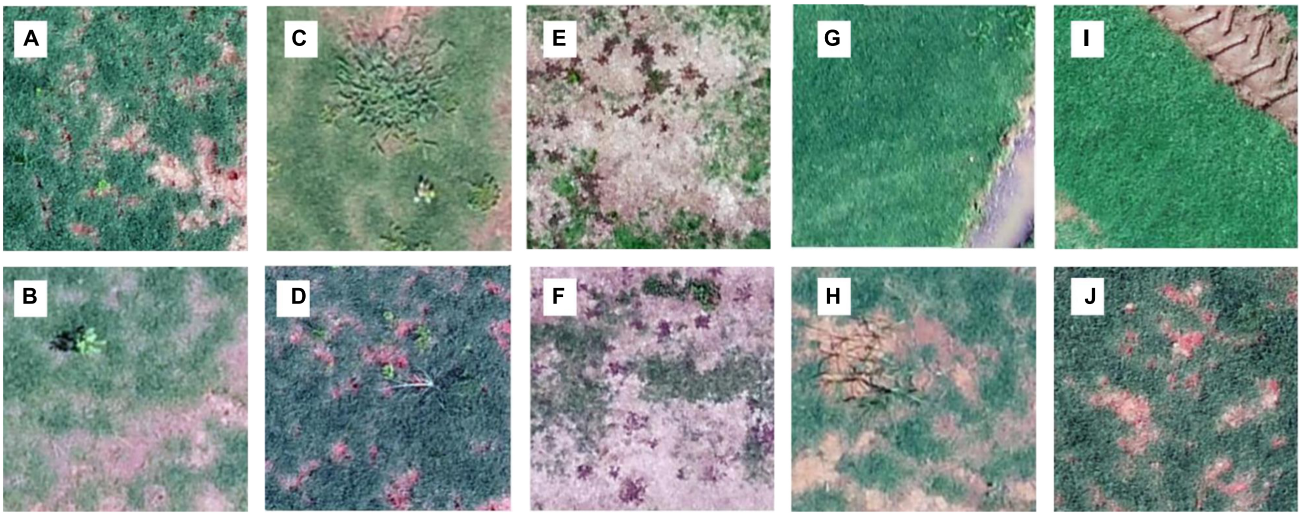


FIGURE 4 | Examples of training images for each class: broadleaf **(A,B)**, grass weeds **(C,D)**, spotted spurge **(E,F)**, sedge **(G,H)**, and no weeds **(I,J)**. The images were obtained through different surveys on Georgian sod farms in 2019 and 2020.

the surveys represent more typical conditions with smaller weeds and more challenging conditions for weed detection. Recall was also used to compare system performance against human performance in order to identify opportunities to improve based on the existing dataset. Three human evaluators visually labeled the validation dataset for each class, and their recall was recorded.

RESULTS

Ground Survey Results

A large portion (35–64%, 52% on average) of the 1.5 m by 1.5 m surveyed areas had no weeds present (**Figure 3**). Categories including broadleaf (**Figures 4A,B**), grass weed (**Figures 4C,D**), spotted spurge (**Figures 4E,F**), sedge (**Figures 4G,H**), and no weeds (**Figures 4I,J**) were recorded in the surveys. Areas of broadleaf and grass weeds accounted for 24–60% and 5%–27% of the total surveyed area, respectively. Sedge was only found in 3–31% of the total area. Spotted spurge was only found in survey 6 (30% of total area) where it could be detected by its purple leaves (**Figures 4E,F**).

Performance of Image Classifier (Convolutional Neural Network) for Weed Mapping

Validation results of the CNN are listed in **Tables 2**, **3**. Images from six surveys were collected under different sod growing stages including establishing, mature, and after harvest in order to train a more generalized model. When using a higher threshold value for the final decision, the precision of the CNN increased but its recall decreased. The CNN detected broadleaf, grass weeds, spurge, sedge, and no weeds at a precision of 0.68–0.87, 0.57–0.82, 0.68–0.83, 0.66–0.90, and 0.80–0.88, respectively, with varying threshold values from 0.5 to 0.2 (**Table 2**). Recall ranges for the five classes were 0.78–0.93, 0.65–0.87, 0.82–0.93, 0.52–0.79, and 0.94–0.99, respectively. Recall of detecting sedge was approximately 10–20% lower than when detecting other classes, indicating a higher number of false negatives. F2 scores were similar to recall due to its emphasis on the number of false negatives. Recall for sedge was elevated from 0.52 to 0.79 if the threshold value was set at 0.2, but the precision of detecting all classes decreased accordingly.

The CNN performed better in detecting validation images from survey 1 than from surveys 2–6 (**Table 3**). Precisions for detecting broadleaf, grass weeds, sedge, and no weeds in survey 1 were 0.87–0.93, 0.89–0.96, 0.87–0.97, and 0.93–0.96, respectively, with varying threshold values. Recall ranges for these four classes were 0.94–0.97, 1.00, 0.76–0.85, and 0.99–1.00, respectively. The metrics for validation images from surveys 2–6 were 10–40% lower in precision and 1–46% lower in recall than the metrics calculated from survey 1. It was noted that the CNN detected classes such as grass weeds and sedge in survey 1 at a much higher recall than in the other five surveys likely due to the larger and more mature weed size.

Performance of Image Classifier Against Human Performance

The model performance indicated by recall was compared against human performance (**Figure 5**). The model recall was similar to human recall in detecting grass weeds, sedge, and no weeds when its threshold value was set at 0.5, but the model recall was higher in detecting broadleaf and spurge than human recall at this threshold. The model was able to detect more weed targets than human eyes if the threshold value was set at 0.3. The lowest human recall was for detecting sedges at 0.54, indicating approximately that half of the sedge targets were not visually identifiable by human eyes. Some examples of images labeled in class broadleaf, grass weeds, and sedge, but not visible to human eyes are demonstrated in **Figure 6**.

TABLE 2 | Validation results of a multiple-class neural network trained on six surveys using architectures ResNet 34 for detection of weed types in sod production fields.

	Broadleaf	Grass weeds	Spurge	Sedge	No weeds	Avg.	Avg.T
Threshold = 0.5							
Precision	0.87	0.82	0.83	0.90	0.88	0.86	0.85
Recall	0.78	0.65	0.82	0.52	0.94	0.74	0.69
Accuracy	0.91	0.93	0.99	0.92	0.89	0.93	0.94
F2 score	0.80	0.68	0.82	0.57	0.93	0.76	0.72
Threshold = 0.4							
Precision	0.81	0.75	0.79	0.86	0.86	0.81	0.80
Recall	0.84	0.72	0.84	0.59	0.96	0.79	0.75
Accuracy	0.90	0.92	0.99	0.93	0.89	0.93	0.93
F2 score	0.84	0.72	0.83	0.63	0.94	0.80	0.76
Threshold = 0.3							
Precision	0.75	0.67	0.74	0.78	0.84	0.76	0.73
Recall	0.90	0.78	0.89	0.68	0.97	0.85	0.81
Accuracy	0.89	0.91	0.98	0.93	0.88	0.92	0.93
F2 score	0.86	0.76	0.86	0.70	0.94	0.83	0.80
Threshold = 0.2							
Precision	0.68	0.57	0.68	0.66	0.80	0.68	0.65
Recall	0.93	0.87	0.93	0.79	0.99	0.90	0.88
Accuracy	0.86	0.89	0.98	0.91	0.86	0.90	0.91
F2 score	0.87	0.78	0.87	0.76	0.94	0.85	0.82

Ave.T is the average metric of the targeted classes including broadleaf, grass weeds, spurge, and sedge.

DISCUSSION

To our knowledge, this is the first study investigating the use of UAS-based images and a deep learning model for weed mapping on sod farms. According to the ground survey result, on average, 52% of each field had no weeds present, which demonstrates the potential for reducing postemergence herbicide use if site-specific weed management can be properly adopted. These reductions can be economically and environmentally impactful. The advantages of using UAS with a simple RGB camera are manifold. Once the detection model and relevant software are available, UAS can cover large fields and generate weed maps in a relatively short time. By integrating this technology into a weed management program, sod growers will have the capability to quickly document problematic areas in the field and make sound treatment decisions. Typically, postemergence herbicides, such as 2,4-D, carfentrazone, dicamba, and simazine, are uniformly sprayed across Bermuda grass fields to provide control of various broadleaf weeds (McCalla et al., 2004; Yu et al., 2019b). By moving broadcast applications to targeted applications, growers will be more competitive with lower herbicide costs. Furthermore, this technology will reduce their environmental footprint by minimizing the pesticides used on sod farms, helping improve the sustainability of the industry.

The CNN trained using ResNet 34 demonstrated the capability to extract color, texture, and shape features (Deng et al., 2010; Grinblat et al., 2016) of different classes of weeds, achieving precision and recall of above 0.9 with the exception of sedges in an establishing field or the larger and mature weeds found in survey 1. The dominant broadleaf weed in survey 1 was pigweed (*Amaranthus* spp.) in varying sizes and growth stages. Precision for detecting broadleaf was 0.93 when the threshold *p*-value was set at 0.5, indicating that only 7% of the targets were misclassified. Results on recall exhibited that approximately 3–6% broadleaf targets were not detected. Yu et al. (2019a) reported a VGGNet model which detected three broadleaf types in Bermuda grass with precision ranging from 0.91 to 0.97 and recall ranging from 0.97 to 1.00. Their model detected almost all the targets, possibly due to the extremely high-resolution images (0.05 cm pixel^{-1}) used to train their model. The CNN in our study only yielded comparable results in survey 1, likely because the weeds were more mature, offsetting the 10–20 times lower (0.57 cm–1.28 cm pixel^{-1}) ground sampling distance than reported by Yu et al. (2019a). Metrics for detecting sedges indicate that 97% of the identified targets (precision) were accurate, and approximately 24% of the sedge targets in the surveyed area were not identified. Sedges are more challenging to detect than broadleaf due to their grass-like morphology: narrow leaf blades and broad ranges of types including nutsedges, annual and perennial sedges, and kyllinga (McCullough et al., 2015). Broadcast postemergence herbicides such as flazasulfuron, halosulfuron, imazaquin, sulfentrazone, sulfosulfuron, and trifloxysulfuron-sodium may still be needed to control sedges given the limitations of our CNN at this time (McElroy and Martins, 2013; McCullough et al., 2015).

Six surveys in our study were conducted in multiple sod fields with different surface conditions (establishing and mature fields) and weed types. It is not surprising that the practical effectiveness of the CNN was lower in the validation dataset of surveys 2–6 than for that of the first survey. Over the whole validation dataset, the CNN detected 78% of broadleaf, 65% grass weeds, 82% spurge, 52% sedge, and 94% no weeds when the threshold *p*-value was set at 0.5. The recall from human evaluators was generally lower than model recall, indicating that the limiting factor was the image resolution and a number of the smaller weeds were simply not visible in these cases. This also explained why a deeper architecture such as ResNet 50 did not improve the model performance. Nevertheless, by lowering the threshold to 0.3, 10% more targets can be identified at the expense of reduced precision. This might be a good option in practice, allowing the growers to balance the cost vs. control. During the surveys in this research, it was noted that weeds were either sporadically distributed across the field or followed a linear pattern, possibly resulting from spread from tractor tires or mowers. Another explanation could be skipped in previous preemergence and postemergence herbicide applications. In this study, the identification of seemingly randomly distributed weeds in a sod production field using the weed map generated by the CNN would be of great economic and environmental benefit. Diverse datasets are needed to train generalized models that perform well in different scenarios because the dynamics of weed pressure are fluid and ever-changing. Given that this study was one of the first attempts to generate a weed map using deep learning in

TABLE 3 | Validation results from survey 1 to the other 5 surveys of the multiple-class neural network trained using architectures ResNet 34 for detection of weed types in sod production fields.

	Broad-leaf	Grass weeds	Spurge	Sedge	No weeds	Broad-leaf	Grass weeds	Spurge	Sedge	No weeds
	Survey 1 validation images					Surveys 2, 3, 4, 5, and 6 validation images				
Threshold = 0.5										
Precision	0.93	0.96	na[z]	0.97	0.96	0.84	0.78	0.83	0.84	0.83
Recall	0.94	1.00	na	0.76	0.99	0.71	0.58	0.82	0.41	0.91
Accuracy	0.97	1.00	na	0.97	0.96	0.87	0.89	0.98	0.90	0.85
F2 score	0.94	0.99	na	0.79	0.98	0.74	0.61	0.82	0.46	0.89
Threshold = 0.4										
Precision	0.92	0.96	na	0.96	0.95	0.76	0.71	0.79	0.80	0.80
Recall	0.94	1.00	na	0.80	0.99	0.80	0.66	0.84	0.49	0.94
Accuracy	0.97	1.00	na	0.97	0.96	0.86	0.88	0.98	0.91	0.85
F2 score	0.94	0.99	na	0.83	0.98	0.79	0.67	0.83	0.53	0.91
Threshold = 0.3										
Precision	0.91	0.94	na	0.91	0.95	0.69	0.63	0.74	0.71	0.77
Recall	0.97	1.00	na	0.82	0.99	0.87	0.74	0.89	0.61	0.96
Accuracy	0.97	1.00	na	0.97	0.96	0.84	0.87	0.98	0.9	0.83
F2 score	0.96	0.99	na	0.84	0.98	0.82	0.72	0.86	0.63	0.91
Threshold = 0.2										
Precision	0.87	0.89	na	0.87	0.93	0.62	0.53	0.68	0.59	0.73
Recall	0.97	1.00	na	0.85	1.00	0.92	0.84	0.93	0.77	0.98
Accuracy	0.96	0.99	na	0.96	0.95	0.80	0.83	0.97	0.89	0.81
F2 score	0.95	0.98	na	0.85	0.98	0.84	0.75	0.87	0.72	0.92

[z] *na, not applicable. No spurge was present in survey 1.*

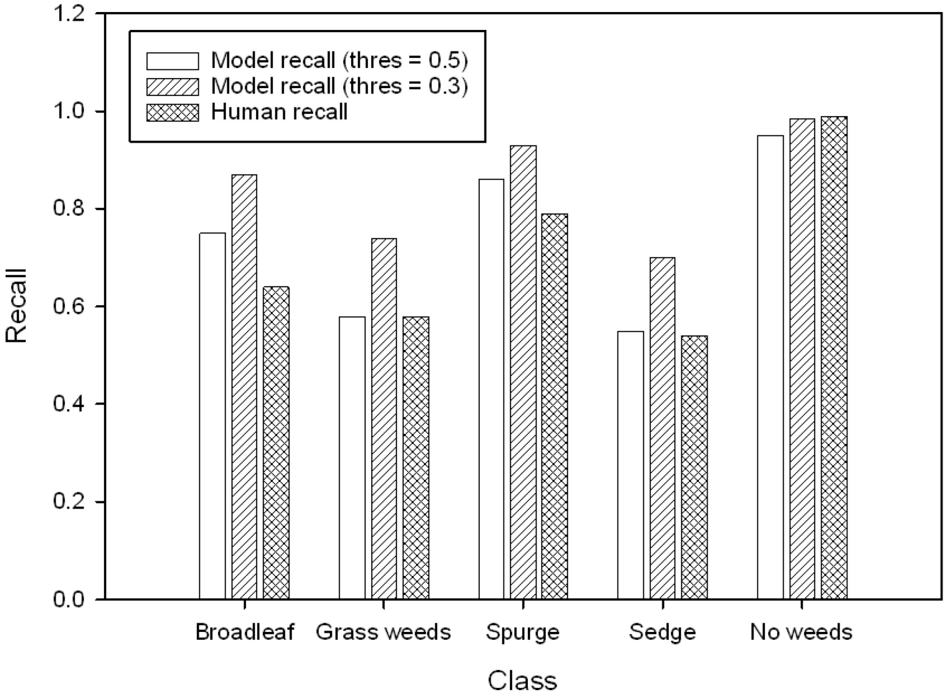


FIGURE 5 | The comparison of recall (threshold values = 0.5 and 0.3) in validation result and recall from human performance (averaged from three evaluators).

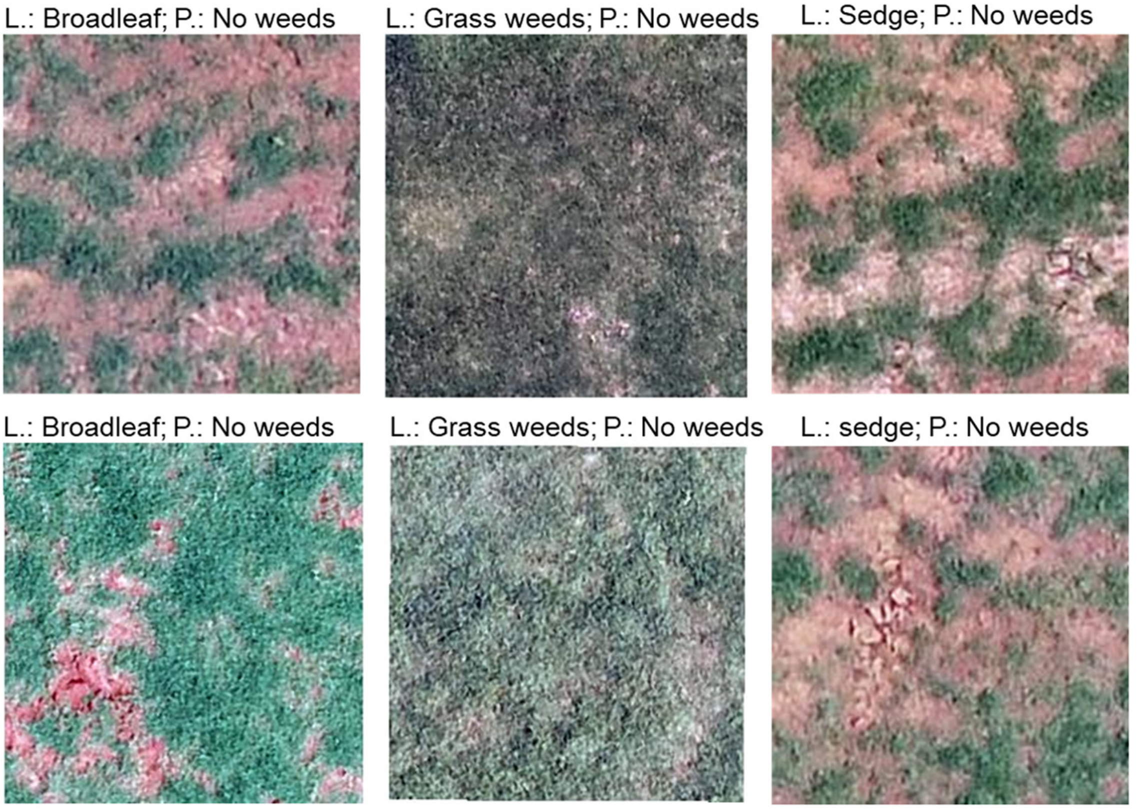

L.: Broadleaf; P.: No weeds
L.: Grass weeds; P.: No weeds
L.: Sedge; P.: No weeds
L.: Broadleaf; P.: No weeds
L.: Grass weeds; P.: No weeds
L.: sedge; P.: No weeds

FIGURE 6 | Examples of misclassified images by the convolutional neural network (CNN). L, true label; P, prediction.

turfgrass, there is less information to compare to at a similar scale and resolution.

The comparison between model recall and human recall suggested that the model performance is limited by image resolution. Higher image resolution would improve the performance of the model but requires greater computing power and either expensive cameras or lower and longer flight time. In some cases, it is a challenge to conduct low-altitude flights due to the close proximity of power lines or trees. Our results indicate that it is difficult to incorporate UAS-based weed mapping at a very early stage of weed treatment when the weeds are still immature or relatively small in size. In the future, technology continues to improve, and UAS-based weed mapping would be improved by higher resolution cameras or fully automated drone fleets flying close to the ground. In addition, a ground-based camera system on tractors or center pivot irrigation system could have much higher resolution and would be ideal for weed mapping if the images were collected in a consistent, timely, and automatic manner.

It remains uncertain whether a single model is sufficient to cover the whole spectrum of weed scenarios in sod production due to the complexity of the turfgrass-weed interactions during the entire growing season. During winter dormancy, however, a separate CNN will be needed to map winter weeds including *Poa annua* and ryegrass (*Lolium* spp.) along with some broadleaf weeds in production fields. Dormant turfgrass provides a brown background with more contrast for weed detection. Even after the CNN for weed mapping is available, there are several hurdles before the implementation of site-specific herbicide applications become routine, including the development of software to generate weed maps using the CNN, ensuring the location accuracy of the position of target weed, and the integration of the weed map into multiple sprayer systems.

SUMMARY

This study included the survey of several sod production fields for broadleaf, grass weeds, spurge, and sedge weed-type composition and areas of infestation, both from the ground level and using UAS, demonstrating the potential of herbicide savings if site-specific weed management is properly adopted. This study successfully trained a CNN for weed mapping using UAS-based imagery and high-level API implementation of a deep learning library. The performance of the CNN was overall similar to, and in some classes (broadleaf and spurge) better than, human identification as indicated by the metric recall. In general, the CNN detected different types of weeds at precision ranging from 0.57 to 0.90 and recall from 0.52 to 0.99 when using UAS images with similar resolution in this study (0.57 cm–1.28 cm pixel^{-1}). Furthermore, it was demonstrated that the CNN can achieve precision and recall above 0.9 for detecting different types of weeds under establishing field conditions when they are larger and more mature. Image resolution is currently the major limiting factor to further improvement of the CNN, with one possible solution being ground-level scouting. Due to the complex ecology and biology of the weeds typically found on sod farms, different models may be needed in practice to improve the overall performance of weed mapping and the eventual targeted, site-specific application of herbicides in these production systems.

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

JZ designed the study, conducted the survey, analyzed the images, and wrote the manuscript. JM conducted the survey, provided the critical technical support on model training, and wrote the manuscript. BS, GR, DJ, and FW provided constructive suggestions on the study design and manuscript. BS and DJ helped to conduct the survey and provided support in coordination activities. All authors contributed to the article and approved the submitted version.

FUNDING

This project was funded by the Georgia Department of Agriculture Specialty Crop Block Grant Program (Grant number: AM180100XXXXG014).

ACKNOWLEDGMENTS

The authors are grateful for the help from Michelle Cooper and Crystal Butler to evaluate the validation images, Amanda Webb, Jonathon Fox, Krishna Katuwal, and Somer Rowe to conduct weed survey, and Chis Roquemore to provide the production fields for data collection.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.702626/full#supplementary-material

REFERENCES

Colvin, D. L., Dickens, R., Everest, J. W., Hall, D., and McCarty, L. B. (2013). *Weeds of Southern Turfgrasses*, ed. T. R. Murphy (Athens, GA: CES/UGA CAES), 208.

de Castro, A. I., Torres-Sánchez, J., Peña, J. M., Jiménez-Brenes, F. M., Csillik, O., and ópez-Granados, F. L. (2018). An automatic random forest-OBIA algorithm for early weed mapping between and within crop rows using UAV imagery. *Remote Sens.* 10:285. doi: 10.3390/rs10020285

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2010). "ImageNet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL.

Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., and Vasilakoglou, I. (2020). Towards weeds identification assistance through transfer learning. *Comput. Electron. Agric.* 171:105306. doi: 10.1016/j.compag.2020.105306

Gašparović, M., Zrinjski, M., Barković, Đ, and Radočaj, D. (2020). An automatic method for weed mapping in oat fields based on UAV imagery. *Comput. Electron. Agric.* 173:105385. doi: 10.1016/j.compag.2020.105385

Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* 127, 418–424. doi: 10.1016/j.compag.2016.07.003

Howard, J. (2020). *Callbacks.lr_Finder*. Available online at: https://fastai1.fast.ai/callbacks.lr_finder.html (accessed November 12, 2021).

Howard, J., and Gugger, S. (2020). Fastai: a layered api for deep learning. *Information* 11:108. doi: 10.3390/info11020108

Hu, K., Coleman, G., Zeng, S., Wang, Z., and Walsh, M. (2020). Graph weeds net: a graph-based deep learning method for weed recognition. *Comput. Electron. Agric.* 174:105520. doi: 10.1016/j.compag.2020.105520

Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., and Zhang, L. (2018). A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. *PLoS One* 13:e0196302. doi: 10.1371/journal.pone.0196302

López-Granados, F., Torres-Sánchez, J., Serrano-Pérez, A., de Castro, A. I., Mesas-Carrascosa, F. J., and Peña, J. M. (2016). Early season weed mapping in sunflower using UAV technology: variability of herbicide treatment maps against weed thresholds. *Precis. Agric.* 17, 183–199. doi: 10.1007/s11119-015-9415-8

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

McCalla, J. H., Richardson, M. D., Karcher, D. E., and Boyd, J. W. (2004). Tolerance of seedling bermudagrass to postemergence herbicides. *Crop Sci.* 44, 1330–1336. doi: 10.2135/cropsci2004.1330

McCullough, P. E., Waltz, C., and Murphy, T. R. (2015). Weed control in home lawns. *UGA Ext. Bull.* 978, 1–11.

McElroy, J. S., and Martins, D. (2013). Use of herbicides on turfgrass. *Planta Daninha* 31, 455–467. doi: 10.1590/S0100-83582013000200024

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. (2019). DeepWeeds: a multiclass weed species image dataset for deep learning. *Sci. Rep.* 9:2058. doi: 10.1038/s41598-018-38343-3

Pérez-Ortiz, M., Peña, J. M., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C., and ópez-Granados, F. L. (2016). Selecting patterns and features for between- and within- crop-row weed mapping using UAV-imagery. *Expert Syst. Appl.* 47, 85–94. doi: 10.1016/j.eswa.2015.10.043

Sasaki, Y. (2007). The truth of the F-measure. *Teach. Tutor. Mater.* 1, 1–5.

Satterthwaite, L. N., Hodges, A. W., Haydu, J. J., and Cisar, J. L. (2009). *An Agronomic And Economic Profile Of Florida's Sod Industry in 2007*. Gainesville, FL: University of Florida, Institute of Food and Agricultural Sciences.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv [Preprint]*. arXiv:1409.1556

Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 45, 427–437. doi: 10.1016/j.ipm.2009.03.002

Stroppiana, D., Villa, P., Sona, G., Ronchetti, G., Candiani, G., Pepe, M., et al. (2018). Early season weed mapping in rice crops using multi-spectral UAV data. *Int. J. Remote Sens.* 39, 5432–5452. doi: 10.1080/01431161.2018.1441569

Torres-Sánchez, J., López-Granados, F., De Castro, A. I., and Peña-Barragán, J. M. (2013). Configuration and specifications of an Unmanned Aerial Vehicle (UAV) for early site specific weed management (D Abbott, Ed.). *PLoS One* 8:e58210. doi: 10.1371/journal.pone.0058210

USEPA (2009). *Amendment to Organic Arsenicals RED. EPA-HQ-OPP-2009-0191-0002.* Washington, DC: USEPA.

Wojciech, F. J., and Landry, G. W. Jr. (2009). *An Economic Profile Of The Professional Turfgrass And Landscape Industry In Georgia. Univ. Georg. Coop. Ext*. Available online at: https://athenaeum.libs.uga.edu/bitstream/handle/10724/12050/RR672.pdf?.1 (accessed November 12, 2021).

Yi, J. (2012). *Economic Analysis of Turfgrass-Sod Production in Alabama. Ph.D. Thesis*. Auburn, AL: Auburn University.

Yu, J., Schumann, A. W., Cao, Z., Sharpe, S. M., and Boyd, N. S. (2019a). Weed detection in perennial ryegrass with deep learning convolutional neural network. *Front. Plant Sci.* 10:1422. doi: 10.3389/fpls.2019.01422

Yu, J., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019b). Deep learning for image-based weed detection in turfgrass. *Eur. J. Agron.* 104, 78–84. doi: 10.1016/j.eja.2019.01.004

Yu, J., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019c). Detection of broadleaf weeds growing in turfgrass with convolutional neural networks. *Pest Manag. Sci.* 75, 2211–2218. doi: 10.1002/ps.5349

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

METHODS
published: 02 December 2021
doi: 10.3389/fpls.2021.780180

A Deep Learning Method for Fully Automatic Stomatal Morphometry and Maximal Conductance Estimation

Jonathon A. Gibbs[1]*, Lorna Mcausland[2], Carlos A. Robles-Zazueta[2], Erik H. Murchie[2] and Alexandra J. Burgess[2]

[1]School of Computer Science, University of Nottingham, Nottingham, United Kingdom, [2]School of Biosciences, University of Nottingham, Loughborough, United Kingdom

Stomata are integral to plant performance, enabling the exchange of gases between the atmosphere and the plant. The anatomy of stomata influences conductance properties with the maximal conductance rate, g_{smax}, calculated from density and size. However, current calculations of stomatal dimensions are performed manually, which are time-consuming and error prone. Here, we show how automated morphometry from leaf impressions can predict a functional property: the anatomical g_{smax}. A deep learning network was derived to preserve stomatal morphometry *via* semantic segmentation. This forms part of an automated pipeline to measure stomata traits for the estimation of anatomical g_{smax}. The proposed pipeline achieves accuracy of 100% for the distinction (wheat vs. poplar) and detection of stomata in both datasets. The automated deep learning-based method gave estimates for g_{smax} within 3.8 and 1.9% of those values manually calculated from an expert for a wheat and poplar dataset, respectively. Semantic segmentation provides a rapid and repeatable method for the estimation of anatomical g_{smax} from microscopic images of leaf impressions. This advanced method provides a step toward reducing the bottleneck associated with plant phenotyping approaches and will provide a rapid method to assess gas fluxes in plants based on stomata morphometry.

Keywords: deep learning, g_{smax} – maximum stomatal conductance, high-throughput phenotyping, semantic segmentation, stomata

OPEN ACCESS

Edited by:
Gregorio Egea,
University of Seville, Spain

Reviewed by:
Qingfeng Song,
Center for Excellence in Molecular Plant Sciences, Chinese Academy of Sciences (CAS), China
Shanwen Sun,
Northeast Forestry University, China

****Correspondence:***
Jonathon A. Gibbs
jonathon.gibbs1@nottingham.ac.uk

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *20 September 2021*
Accepted: *01 November 2021*
Published: *02 December 2021*

Citation:
Gibbs JA, Mcausland L, Robles-Zazueta CA, Murchie EH and Burgess AJ (2021) A Deep Learning Method for Fully Automatic Stomatal Morphometry and Maximal Conductance Estimation. Front. Plant Sci. 12:780180. doi: 10.3389/fpls.2021.780180

INTRODUCTION

Stomata are pores on a leaf that allow the exchange of gases between the atmosphere and the plant through their opening and closure (i.e., stomatal conductance – g_s). Carbon dioxide (CO_2) enters the plant in a trade-off against water vapour, which is simultaneously lost through transpiration. Stomata are found on almost all aerial plant organs and can be arranged in rows aligned with veins such as in monocotyledonous grasses or dispersed/clustered in dicotylodonous plants (Rudall et al., 2013). Their function is mediated by a pair of specialised cells, the guard cells, that control the aperture of the pore and determines the potential g_s. As such, stomata are key "gatekeepers" positioned between the atmosphere and the internal plant tissue and are key in influencing photosynthetic rate, water loss, and water use efficiency (WUE) (Buckley, 2005; Berry et al., 2010). Stomatal morphology is diverse, with patterning (such as clustering), size and density reflecting inter- and intra-specific differences (Franks and Farquhar, 2007; Dow et al., 2014; McAusland et al., 2016), growing conditions (Casson and Gray, 2007), and evolutionary selection pressures (Franks and Beerling, 2009; Mcelwain et al., 2016). These anatomical characteristics have been shown to translate into functional diversity with, for example, size and density partly determining the leaf conductance capacity whilst the rapidity of guard cell movement determines the speed of response, or sensitivity, to environmental factors such as fluctuating light and water availability (Franks et al., 2015; McAusland et al., 2016; Bertolino et al., 2019). Indirect agronomic selection has been shown to lead to altered stomatal conductance in wheat (Fischer et al., 1998).

A measurement of stomatal size allows a calculation of the potential maximal rate of g_s to water vapour, known as anatomical maximum stomatal conductance (g_{smax}; previously termed g_{max} or g_{wmax}; Equation 1).

$$g_{smax} = \left(d.D.a_{\max} \right) / \left(v.\left(l + \left(\frac{\pi}{2} \right).\sqrt{\frac{a_{\max}}{\pi}} \right) \right) \qquad \text{Eq. 1}$$

Where d is the diffusivity of water in air ($m^2 s^{-1}$, at 25°C), D is stomatal density for a single leaf surface (mm^{-2}), and l is pore depth (μm) and is estimated as half the mean guard cell width. For elliptical (i.e., graminaceous) guard cells, maximum stomatal pore area (a_{max}; μm^2) is estimated as an ellipsis with the major length estimated as pore length and minor length estimated as half the length of the peristomatal groove. For circular guard cells, a_{max} is calculated as the area of a circle with diameter corresponding to the pore length. Finally, v is the molar volume of air ($m^3\ mol^{-1}$ at 25°C), and π is the mathematical constant taken as 3.142 (Parlange and Waggoner, 1970; Weyers and Johansen, 1990; Franks and Beerling, 2009).

Anatomical g_{smax} often exceeds operational g_s by several fold (Sack and Buckley, 2016), but works in parallel with g_s at a spatial and temporal scale to optimise stomatal responses to the prevailing environmental conditions (Murray et al., 2020). High g_{smax} precludes high g_s under yield potential conditions and can be used to predict g_s under well-watered, light-saturated environments (Dow et al., 2014; Murray et al., 2020).

Improving the throughput and accuracy of measurements of stomatal size and density for the derivation of g_{smax} is essential, however, manual measurements of stomata are highly time consuming and small datasets are common when collecting images with few defects. Traditionally, stomatal density or index, the ratio of stomatal complexes to epidermal pavement cells, are collected through manual counting whereas measurements of pore and guard cell characteristics (morphometry) can be obtained through scaled dimensions using image processing software such as ImageJ (Schindelin et al., 2012). Whilst manual counts and morphometries are sufficient for smaller sample sets, they are untenable for screening larger populations – for example for genome-wide association studies (GWAS) – which often consist of 100 s of lines with multiple replicates. Moreover, further issues arise in that they are susceptible to intra-rater or inter-rater repeatability (the subjective differences in measurements between individuals, or from a single individual repeating the same task), consequently reducing accuracy. One such solution to the limitations of manual morphometry can come from the field of neural networks, namely, deep learning. In deep learning, a computer model learns to perform classification tasks from images, text, or sound with a high degree of accuracy, sometimes exceeding human-level performance. The training of a deep learning model requires a human annotated dataset, which the model learns from and once trained, can be applied to future predictions, namely the same classification tasks on unseen data.

As of late, deep learning has received an increased amount of attention for both plant and stomatal phenotyping and various deep learning models have been proposed. With respect to stomata literature, the most common application of deep learning is for the detection and counting of stomata in images. Fetter et al. (2019) use a deep convolutional neural network (DCNN), AlexNet (Hinton et al., 2012), to generate a likelihood map for each input image followed by a thresholding and peak detection to localise and count stomata and achieved an accuracy of 94.2%. Zhu et al. (2021) use a Faster R-CNN combined with a U-Net to automatically count stomata and epidermal cells for the calculation of stomatal index and achieve 98.03 and 95.03% accuracy for stomata and epidermal cells, respectively. In other instances smaller, shallower, networks are used for counting; a convolutional neural network (CNN), VGG (named after the Visual Geometry Group where the method was conceived), is commonly used to detect each stoma, encapsulating the detections in bounding boxes (Simonyan and Zisserman, 2015). Meeus et al. (2020) use VGG19 in which the number (19) corresponds to the number of layers. Casado-García et al. (2020) use an object detection network known as YOLO (Redmon and Farhadi, 2018), to detect the bounding boxes of stomata with accuracy of 91%. Whilst good results are reported for detecting stomata using the VGG and YOLO networks, a considerable amount of post-processing is required if morphological measurements are to be extracted, which is susceptible to error. Alternatively, deep learning approaches have been used for the classification of stomata types; Andayani et al. (2020) created a CNN that determines whether the input image contains stomata from turmeric (also known as kunyit; *Curcuma longa*) or temulawak (also known as Java ginger; *Curcuma zanthorrhiza*). Using a small dataset of only ~300 images, they achieve classification accuracy of 93.1%. More recently DeepImageJ, a deep learning framework to plugin for ImageJ was released (Gómez-de-Mariscal et al., 2021). DeepImageJ provides significant advances of traditional methods and improves the capabilities of ImageJ, incorporating support for deep learning networks. Outputting high quality, accurate, classification of data, however, the specific results depend upon user design and implementation.

Current methods to comprehensively calculate stomatal morphometry are lacking and the limited studies to do so using a combination of deep learning and image processing. These methods typically focus on stomata detection *via* bounding boxes followed by image processing algorithms to obtain limited morphological data. However, these methods often require specific fine tuning where a change in intensity or blur within the image set will significantly reduce the accuracy. Toda et al. (2018) detect stomata, the pore and whether it is open or closed using a three-stage approach; (1) the use of the histogram of gradients (HOG) to detect stomata in the images and extract bounding boxes, (2) a CNN to classify the HOG detections as open or closed stomata, and (3) Pore quantification using a series of image processing algorithms, reporting accuracy of 92%. Bhugra et al. (2019) propose a framework consisting of two neural networks; the first, a DCNN, is used detect stomata in images, the second is a fully convolutional neural network (FCNN) which accepts the detected bounding box as input and extracts the stoma from the bounding box. Ellipse fitting is applied to the resulting FCNN output to generate an estimate of pore shape. Whilst producing good results, accuracy of ~91% for detection, ellipse fitting can over- or under-fit the pore. Moreover, instances where the pore is not ellipse shaped will lead to significantly inaccurate results. (c) use AlexNet to detect stomata and estimate pore area using a series of image processing algorithms [such as Contrast Limited Adaptive Histogram Equalisation (CLAHE)], achieving up to 85% accuracy. To date, both guard cell and pore measurements have yet to be obtained from a single network.

Semantic segmentation, in which each pixel of an image is labelled with a corresponding class, allows the preservation of morphometry. Unlike bounding box algorithms, the output in semantic segmentation is the image mask; a high-resolution image (typically of the same size as input image) in which each pixel is classified. Previous applications of semantic segmentation include, but are not limited to, medical imaging analysis (Jiang et al., 2018), autonomous driving (Siam et al., 2018), and classification of terrain from satellite imagery (Wurm et al., 2019). Despite the ability for semantic segmentation to extract morphometric information, it has yet to be applied to stomatal phenotyping.

Here, we aim to reduce the bottleneck associated with manually measuring morphometric traits of stomata and provide a proof of concept study for the determination of anatomical g_{smax} by the development of a high-throughput phenotyping method using semantic segmentation. We incorporate aspects of existing deep learning models, such as the Attention U-Net architecture (Oktay et al., 2018) and Inception Network (Szegedy et al., 2016), discussed in the *methods* section, on a small dataset (<350 images), whilst computational costs are reduced by restricting the number of the trainable parameters when compared to many of the existing deep learning methods for stomata. Through this method, we: (1) automatically differentiate between distinctive stomatal types, the dumbbell shaped Poacaea and dicotyledonous stomata, (2) count stomata, (3) extract multiple morphological traits, (4) calculate density, and (5) calculate anatomical g_{smax} as circular or ellipse based on the type of stomata. We provide a substantial advance with the application of semantic segmentation to stomata and the first to show deep learning can produce high-throughput stomata phenotyping calculating anatomical g_{smax}. The tools developed here are freely available (See "*Data*" and "*Data availability*" sections).

MATERIALS AND METHODS

Data

In highly researched areas, such as object detection or handwriting recognition, existing datasets such as ImageNet (Deng et al., 2010), or MNIST (Deng, 2012), provide access to hundreds of thousands of annotated images. In the case of stomata, however, very few annotated datasets are freely available.

Two balanced datasets with distinctive stomata were chosen to evaluate our proposed model: a monocotyledonous Poaceae representative with dumbbell shaped stomata (wheat; *Triticum aestivum*) and dicotyledon with kidney shaped stomata (poplar; *Populus balsamifera*). For the wheat set, spring bread wheat cultivars were chosen from the Photosynthesis Respiration Tails (PS Tails) Panel and from the International Maize and Wheat Improvement Centre (CIMMYT); with eight genotypes selected for their contrasting plant architecture and aboveground biomass that were grown under yield potential conditions in a glasshouse. A subset of the data was used in this study, consisting of 348 images captured at a resolution of 2,592 × 1,944px with a 10 × 40 magnification. The stomatal impressions were collected using nail varnish and adhesive tape in the medium area of adaxial and abaxial sides of the main shoot flag leaf. Samples were left to dry for 10 min and then placed on a slide to be examined and photographed. Images were collected using a Leica DM 5000 B microscope (Wetzlar, Germany). The poplar dataset in this study was first published by Fetter et al. (2019) and is publicly available. A subset of the data was selected from an intraspecific collection of balsam poplar through random selection. A small subsample, totalling 114, images were annotated, which are of 2,048 × 2,048 px resolution with a 10 × 40 magnification. Note: the reduced poplar dataset used in this study, along with the corresponding annotations, has been made publicly available with links to the original source. The impression quality does not directly impact the quality of results unless the impressions used to train the network differ significantly from those used to test it. However, the quality should be good enough such that a human expert can manually annotate the images. Whilst the image set used for training was lower for the poplar, the increased density of stomata within each image led to a greater amount of stomata annotated overall (i.e., see **Table 1**).

An overview of the proposed method is given in **Figure 1**. For the annotation of both poplar and wheat datasets, a pixel level classification was performed where each pixel was labelled as guard cell, pore, or discard, to create the image mask using the Pixel Annotation Tool (Bréhéret, 2017). The discard refers to the background, noise (except that over the stoma), and subsidiary cells, which are not used in the calculation of g_{smax}. Similar annotation approaches could be used for other structures, such as epidermal cells, trichomes or, on the whole plant scale, yield components for example.

Data Augmentation

To increase the size and variation of the dataset, a series of augmentations were applied to manipulate the images prior to

TABLE 1 | Overview of image datasets and properties.

Dataset	# Images	Size (px)	# Stomata	Density (mm^2)	μm Per Pixel
Wheat	348	2,592 × 1,944	1,600	63	0.12547
Poplar	113	2,048 × 2,048	3,862	246	0.18181

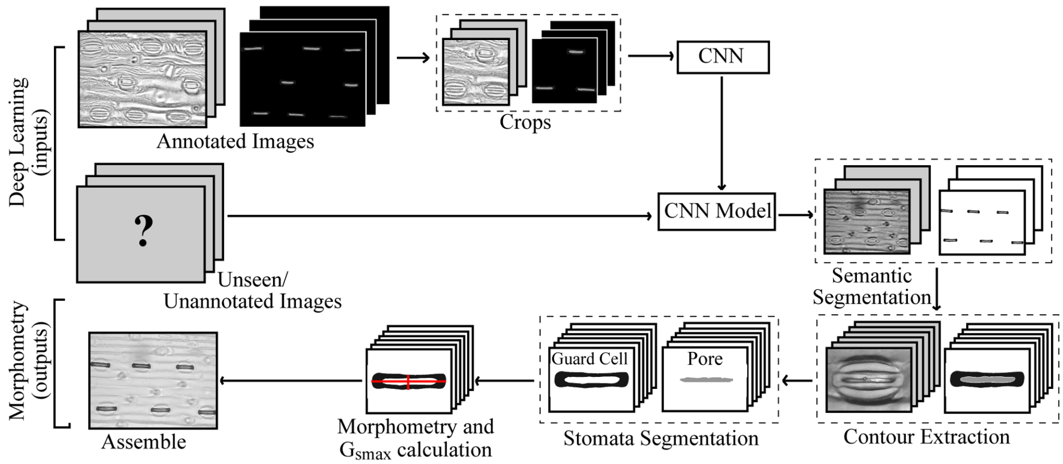


FIGURE 1 | Pipeline for the proposed method of extracted morphometric properties of stomata for the estimation of maximal stomatal conductance; anatomical g_{smax}.

and during deep learning. Each original image is cropped into four overlapping regions as (1) the original image resolution is too large and is computationally expensive to maintain and (2) due to the small size of the pore within images, scaling the original image results in a high loss of accuracy. Augmentations within the network are applied for each image every epoch (a full iteration of the dataset) and are performed as follows; A subsample of the image is taken at a resolution of 768×768px. The centre of the bounding box, i.e., the area of the subsample, is determined by a series of random variables; the first randomly selects whether to perform a *stomata crop*; using the centre of the stomata, or a *random crop*; a random position within the image, with an 4:1 probability, respectively. The stomata crop randomly selects a stoma in the image and applies random jitter to the position with upper bounds of 15% of the image size. The random crop is selected anywhere within the image bounds excluding half the crop size around the border of the image. For both crop methods, a random rotation is applied ranging between plus and minus 30°. There is a 20% probability that the image will be flipped vertically or horizontally and a 30% probability of blur, sharpness, or contrast manipulation. These augmentations increase the dataset size and help prevent overfitting (where a network learns only the data it is being trained on). The augmentation is applied to the training dataset.

Deep Learning

Within this project, all deep learning was performed using Python.

A brief overview of CNNs is provided for those who have no prior knowledge; for further reading, see (Maier et al., 2019). A CNN is a deep learning algorithm with a particular focus on imagery, for example, object detection or image classification within two-dimensional images. It is made up of a series of layers, each of which have a set of trainable parameters. The CNN takes as input an image and passes it through multiple layers and outputs a prediction that represents the class label of the input data, whether as an image as a whole or at pixel level. The three most common layers in a CNN are (1) *The convolution* layer which applies multiple filters, which aim to detect patterns such as edges, the input each of which have different parameters, so each filter is able to learn contrasting features whilst preserving the spatial relationship between pixels. The filters pass over the image, scanning a few pixels at a time, and creates a feature map. After a convolution, an *activation function* is performed to introduce nonlinearity calculating a weighted sum of its inputs and adding a bias. (2) The *Maxpooling* layer downsamples the feature map reducing its dimensionality, providing an abstracted form of the representation, and the associated computational costs. It allows for the CNN to be robust against minor displacements. (3) The final layer of a CNN is the *fully connected output* layer. After a sequence of multiple layers, it takes the outputs of these and classifies the pixels, computing scores for each class label applying an activation function such as *SoftMax*, which converts a set of numbers into a set of probabilities. Additional common steps can include *skip connections*, which allows the output of some layer to skip some other layers and be passed as input to layers further down the network.

The performance of a CNN, how well it has managed to learn these parameters and make predictions, can be evaluated in numerous ways. The *score* function or evaluation metric, evaluates the accuracy of the model during training, comparing the predicted outcome to the ground truth (i.e., the labels). The higher the score, the higher the degree of accuracy thus indicating that the model is correctly making predictions. The *loss* function is used as a method of evaluating how well the algorithm models the given data during training. If the predictions of a model deviate from the ground truth, a high loss value is returned. Too little data variation combined with a large network or high number of epochs can result in *overfitting*, where the model learns the training data and is unable to adapt to new or varying inputs.

The structure of CNNs vary depending on the data, application, or the size of the network and so multiple networks exist. In this study, we propose a CNN using features of both an Attention U-Net (Oktay et al., 2018) and Inception (Szegedy et al., 2016) to make pixel-level predictions of stomata for both guard cell and pore (**Figure 2**). The original U-Net model (Ronneberger et al., 2015), which was primarily developed for biomedical image segmentation, is a U-shaped network comprised of a series of encoder and decoder layers. The encoder layer is the downwards trajectory performing a series of convolutions and maxpooling, encoding the input sequence (**Figure 2**). The decoder performs the opposite, an upwards trajectory applying deconvolution to increase dimensionality, decoding the input sequence to an output sequence. Skip connections are added between encoder and decoder layers to combine spatial information. However, whilst skip connections offer many advantages, such as the ability to maintain feature information, they introduce many redundant low-level feature extractions, as feature representation is poor in the initial layers. Attention U-Net overcomes this, expanding on the original U-Net model, by adding attention gates which seek to highlight salient features. Skip connections combined with attention gates suppress activations in irrelevant regions, reducing the number of redundant features. The inception architecture employs multiple convolutions and pooling layers simultaneously in parallel within the same layer (inception layer) using the same input. The inception layer reduces the computational costs of the model and automatically selects the most useful features when training the network.

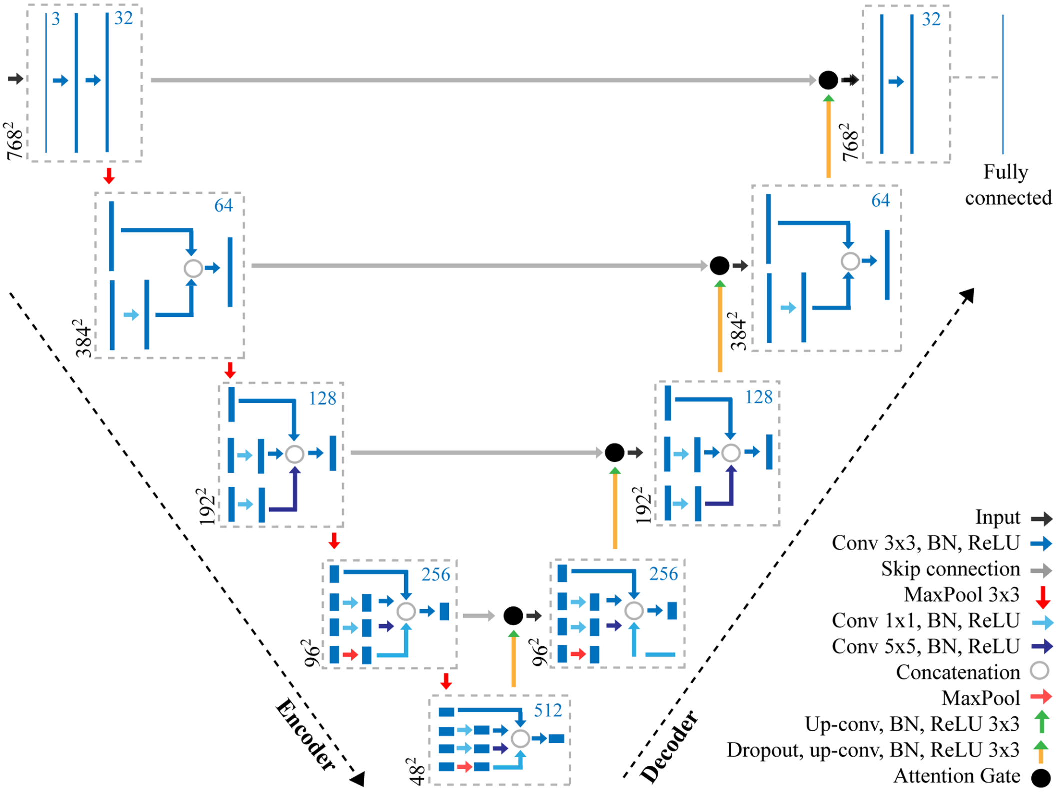


FIGURE 2 | Overview of the adapted CNN used for the extraction of stomata morphometry. The proposed CNN combines features of both an Attention U-Net (Oktay et al., 2018) and Inception (Szegedy et al., 2016) to make pixel-level predictions of stomata for both guard cell and pore. The CNN contains a number of layers including convolution (Conv) layers, Max pooling layers, and fully connected layers. The output of each convolution layer is a set of 2D images, known as feature maps, which are computed by convolving previous feature maps with a filter, the size of which is given in the key. Batch normalisation (BN) and Rectified Linear Units (ReLU) steps are added to normalise data and remove negative pixel values from features maps. Skip connections help to maintain spatial information whilst the Attention Gate removes redundant features. The number of filters at each step is given as the blue number, whilst the resolution is given in black.

Here, we present an incremental model (**Figure 2**), which increases the branches as the depth of the network increases, this works as follows.

- *Encoder layer 1*; The network takes, as input, an image with dimensions of 768 × 768 pixels. A Convolution (Conv) with a 3 × 3 filter, followed by Batch Normalisation (BN; normalises the input by re-scaling and re-centering the data, which increases the stability and speed of the network) and Rectified Linear Unit (ReLU; in which all negative pixel values in the feature map are converted to zero) is performed three times (we refer to this as Conv 3 × 3, BN, ReLU*x3*). Maxpooling is then applied with a kernel size of 3 × 3.
- *Encoder layer 2*; Receives input from the previous layer passing it through Conv, BN, ReLU*x2* followed by a maxpooling layer with a 3 × 3 kernel.
- *Encoder layer 3*; The input of the previous layer is copied into two branches, the first applies a Conv 3 × 3, BN, ReLU, whilst the second applies a Conv with a 1 × 1 filter, BN, ReLU followed by a Conv 3×3, BN, ReLU. The values are concatenated and a further Conv 3 × 3, BN, ReLU is applied. Maxpooling further reduces the dimensionality.
- *Encoder layer 4*; The input of the previous layer is passed to three branches, the first two are the same as the third encoder layer, whilst the additional branch performs Conv 1 × 1, BN, ReLU followed by a Conv 5 × 5, BN, ReLU. The values are concatenated and a further Conv 3 × 3, BN, ReLU is applied followed by maxpooling
- *Encoder layer 5*; Is the same as the previous encoder, but with an additional branch this time performing maxpooling with a 1 × 1 kernel followed by Conv 1 × 1, BN, ReLU.
- *Decoder layers 5–2*; Decoder layers 5–2 are the same as the encoder layers, though the maxpooling operation, which is used to down sample, is changed to a transpose convolution, which increases the dimensionality.
- *Decoder layer 1*; the final decoder, is responsible for the final output of the model and applies Conv 3 × 3, BN, ReLU*x3* followed by a fully connected layer to output predictions.

The parameters of the network were trained using Stochastic Gradient Descent (Kiefer and Wolfowitz, 1952) with a momentum of 0.9 and a learning rate of 0.1. The model was trained on an Nvidia Titan V GPU for 50 epochs using a batch size of 8. Whilst a GPU is not necessarily a requirement for deep learning, the speed of computations will be considerable using a CPU only. The Lovasz-Softmax (LS) loss function (Berman et al., 2018) is used; LS is a loss function for multi-class semantic segmentation incorporating SoftMax and supports direct optimisation of the mean intersection-over-union (IoU) loss in neural networks. IoU, also known as Jaccard index, is used to compute the area of overlap between the target mask (i.e., the annotated labels) and the predicted mask. The score function, or evaluation metric, evaluates the accuracy of the model during training. In this study, we use the IoU as a score function in two ways; (1) IoU is used to represent the percentage of overlap and (2) a confusion matrix summarises the performance of the model providing insight into the errors being made, returning an accuracy of the network. Moreover, the confusion matrix accounts for uneven number of samples for each class.

Once trained, the model allows new, unseen, images to be passed into the network producing, as output, a pixel-level annotation, the mask, of stomata within it. Unlike existing methods that use image processing methods to quantify the morphometry of stomata, in this study, the process is simplified by directly manipulating the mask. As a result, calculating morphometry becomes a relatively straightforward task, accomplished using a single network, and simple pixel counting.

Stomata Morphometry

Morphological traits such as length and width of pores can be segmented from the output of the CNN model proposed here by extracting information from the pixel-level labelled mask predicted by the CNN (**Figure 3A**). Contours in the mask are identified surrounding the guard cell (**Figure 3B**), and all pixels within each contour are selected and assigned to each individual stoma. A bounding box is fit around the contour and all background is removed (**Figure 3C**). Each individual stoma is rotated such that the principal axis is in line with the bounding box using the eigenvalues obtained

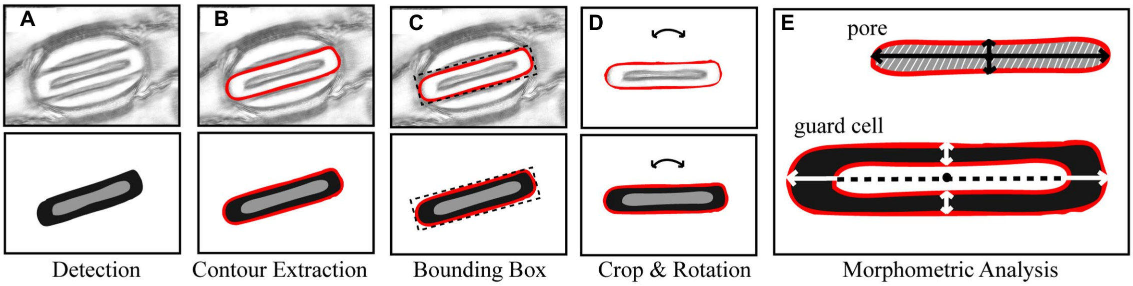


FIGURE 3 | Overview of the stages of stomata morphometry extraction. **(A)** each stoma is detected using the CNN model described in **Figure 1**; **(B)** the contour is extracted; **(C)** a bounding box is applied to the contour; **(D)** the bounding box is rotated using the primary eigen vector and the stoma contained within the contour is cropped; and **(E)** morphometric measurements of the guard cell and pore are automatically extracted including guard cell and pore length and widths plus peristomatal groove distance.

from principal component analysis (**Figure 3D**). The rotation step supports the trait extraction; allowing widths and heights to be easily obtained. The mask is then split based on the corresponding label thus enabling the extraction of the pore from the stoma leaving the guard cell for automated morphometry (**Figure 3E**).

To calculate the morphometry of each stoma within the image, it is represented as a two-dimensional matrix where the values correspond to pore, guard cell, or discard. From this the width, height, and area of both the guard cell and pore can be calculated as a sum of pixels multiplied by the μm to pixel conversion. To obtain the measurements relating to the guard cell, the centre point, along both *x* and *y*, is selected and the length and width are calculated as the average sum of pixels along 10 pixel transects surrounding this centre point. This averaging is used to account for artifacts in the data (i.e., asymmetry in guard cell shape). The same process is applied to the pore.

Stomatal density is automatically calculated from the dataset. For all images, the number of stomata is counted, excluding any detected stomata, which intersects the left or bottom border of the image. The area within each image (i.e., the field of view; FOV) is calculated using a pixel to mm conversion (Equation 2)

$$FOV\left(mm^2\right) = \left(\frac{\mathrm{w} * \mu m\,\mathrm{pixel}^{-1}}{1000}\right) * \left(\frac{\mathrm{h} * \mu m\,\mathrm{pixel}^{-1}}{1000}\right) \quad \text{Eq. 2}$$

Where *w* and *h* correspond to the width and height of the image in pixels. Density, *D*, is then calculated according to Equation 3:

$$D = \frac{\text{Total number of stomata}}{mm^2} \quad \text{Eq. 3}$$

Using these measurements, g_{smax} can be calculated using Equation 1.

RESULTS

The network was evaluated for its ability to accurately classify stomata type between wheat (Poaceae) and poplar in the datasets provided, detect features, obtain morphological traits, and predict g_{smax} compared to manual calculations.

Stomata Detection

An example test image is presented in **Figure 4** with the associated morphometric measurements.

The proposed network can be readily applied to both poplar and wheat, which have contrasting patterning (files vs. random spacing), thus making the method more universally applicable. The proposed model was evaluated against the U-Net (Ronneberger et al., 2015) and the Attention U-Net (Oktay et al., 2018) architectures. For each architecture, 25 epochs were performed using the same train and validation data. The results can be seen in **Table 1**; where *parameters* corresponds to the total number of trainable parameters in the network, *Time* is the total execution time in minutes, *IoU* is the intersection over union score; a value between 0.0 and 1.0 with 1.0 meaning that the prediction from the network is equivalent to the manual annotation, *Loss* is the result of the LS loss function, and *Acc.* is the accuracy of the model using a confusion matrix. As we can see from **Table 2**, the network proposed here has 50% fewer parameters than the related architectures, U-Net and Attention U-Net, and achieves at equal accuracy a higher IoU and a lower loss in a shorter amount of time.

The number of parameters can have a direct impact on the computational cost of training a network and the future predictions made on unseen images. In most instances, a smaller number of parameters is preferable, particularly when access to high-spec hardware is limited. For that reason, we have reduced the parameters of the well-known U-Net architecture. The network proposed here has a total of ~8 million parameters, which is considerably less than existing approaches used for stomata deep learning, for example, the VGG16 network has ~138 million trainable parameters and the YOLO network has ~63 million. Here, we show that the number of parameters can be reduced whilst obtaining a higher degree of accuracy with our proposed method achieving 100% accuracy for stomata counts across both datasets. Moreover, no false positives, the prediction that a stoma is present when it is not, were recorded. If false positives were to be detected in images, the contour detection stage, discussed in the previous section, would discard any small errors based on average size of the stomata in the image.

g_{smax}

Manual calculations of morphometry for 20 images of both the wheat and poplar dataset were obtained by an expert, and the measurements were used to calculate g_{smax} using Equation 1. The images chosen were of various quality and spanning a range of examples from each dataset. These values were compared to those obtained using the automated method proposed here. One further benefit of the proposed CNN is that the stomatal type has been detected, and so g_{smax} can be calculated based on the most appropriate stomatal shape: circular for poplar or elliptical for graminaceous wheat stomata. It is worth noting that the difference here, between the predicted and manually determined measurements, is not classified as an error as the manual process is susceptible to intra-rater or inter-rater repeatability. To determine g_{smax} a series of variables need to be extracted from the data.

Stomatal density, given as an average across all images in the set, is given in **Table 2**, calculated using Equations 2, 3. In general, stomatal density is the biggest driver of variation in g_{smax}, because the other two input variables (pore length and guard cell width) are averaged across many stomata and will differ less among samples. Within this proof of concept, the magnification required to calculate morphometry does not necessarily capture an accurate

Type	Poplar
Count	24
Density	246
Av. Pore Length	25.73
Av. GCW	5.62
Av. PSG	25.4
g_{smax}	7.47

	Pore Lgth (μm)	GCW1 (μm)	GCW2 (μm)	PSG (μm)
1	27.82	5.09	6.36	27.09
2	24.91	8.18	8.55	29.27
3	27.09	6.18	5.27	27.82
4	26.91	4.55	6.55	26.18
5	27.27	5.27	5.09	26.00
6	25.99	4.55	5.27	25.45
7	25.82	5.09	5.09	25.27
8	27.82	5.09	4.73	25.45
9	26.73	6.36	4.00	25.45
10	25.99	7.27	5.27	26.00
11	25.64	4.91	4.91	25.82
12	25.64	5.09	5.45	25.27
13	27.82	5.45	6.36	25.64
14	24.73	5.82	6.36	23.64
15	24.91	6.55	6.73	26.00
16	27.82	4.36	4.73	23.82
17	24.55	5.27	5.09	24.73
18	25.45	5.45	6.91	25.45
19	23.64	5.09	6.36	24.91
20	25.99	4.91	5.27	24.18
21	24.73	5.09	5.64	24.36
22	22.18	5.82	7.45	26.18
23	24.55	4.18	5.45	23.09
24	23.64	6.18	4.91	22.55

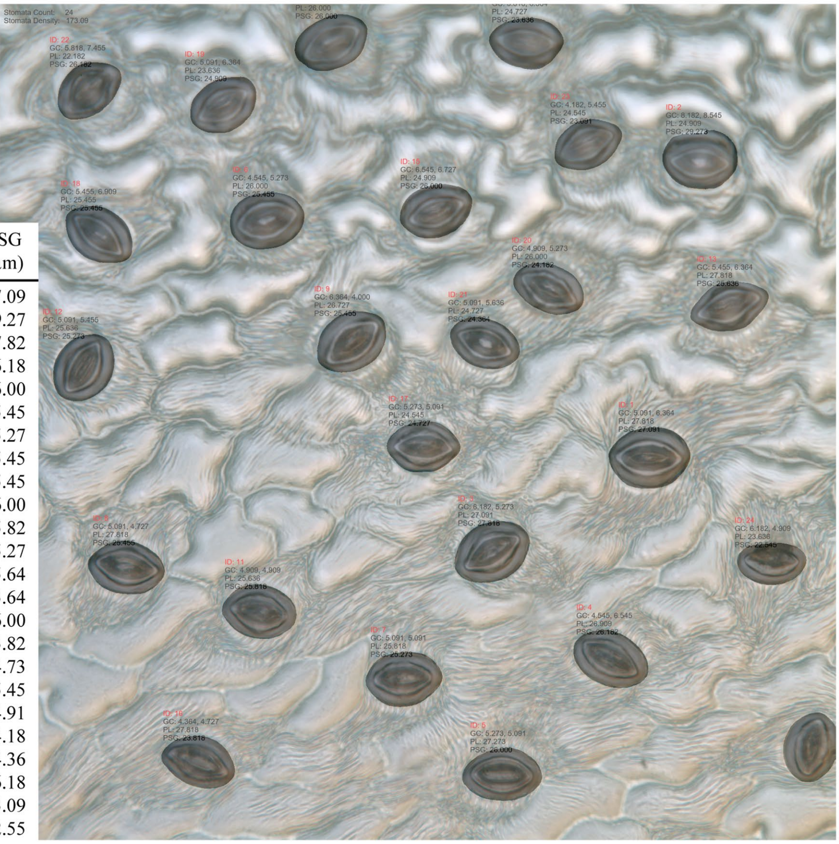


FIGURE 4 | Example output from the CNN model applied to an unseen poplar image. Summary results for the whole images are given in the top table, whilst the measurements for individual stomata are given in the bottom, where GCW refers to guard cell width and PSG refers to peristomatal groove distance.

TABLE 2 | Comparison of the proposed convolutional neural network (CNN) relative to two other common CNN architectures.

Method	Parameters	Time (m)	IoU	Loss	Acc.
U-Net	~16,482,000	200	0.78	0.18	0.98
Attention U-Net	~17,450,000	343	0.72	0.18	0.97
Proposed method	~8,114,000	176	0.84	0.16	0.98

stomatal density as it will not cover a wide enough range of samples, thus the g_{smax} results presented here may differ from those reported elsewhere in the literature (Lammertsma et al., 2011). This is particularly the case for the poplar dataset whereby the obtained sample images were of fixed magnification and were originally collected to test a stomata counting system, focused on relatively stomatal-dense samples (i.e., leaf sections lacking vein structures etc.; Fetter et al., 2019). In contrast, the g_{smax} values calculated for wheat are likely more accurate because wheat stomata are patterned in rows and thus calculating density at 10 × 40 has less spatial bias. This can be overcome through the addition of more samples at this same magnification or through an additional step to count stomata at a lower magnification.

For each image, the manually and automatically calculated g_{smax} is given in **Figure 5**. For the wheat dataset, the average difference between the manual and automated measurement was 3.8%, with a slope of 0.9373 and R^2 of 0.9661. For the poplar dataset, the average difference between manual and automatic calculated g_{smax} was 1.9%, with a slope of 0.9842 and R^2 of 0.9782.

This method also allows *operational* g_s to be calculated on a per image basis, or over a set of images. Replacing a_{max} in Equation 1 with the area of the pore allows such calculations to be made.

DISCUSSION

Here, we present a significant advancement in methodology, which permits both morphological (density, size, and area) and functional (anatomical g_{smax}) attributes to be predicted from purely image-based data that is easy to obtain and can be translated to high throughput systems. To our knowledge, this is the first time that the dumbbell – like Poaceae has been distinguished from dicotyledonous stomata and g_{smax} predicted using automated stomatal morphometry. Thus far g_{smax} has been used in disciplines where gas exchange measurements are inconvenient or simply not possible, for example, the recreation of conductance in palaeoclimates

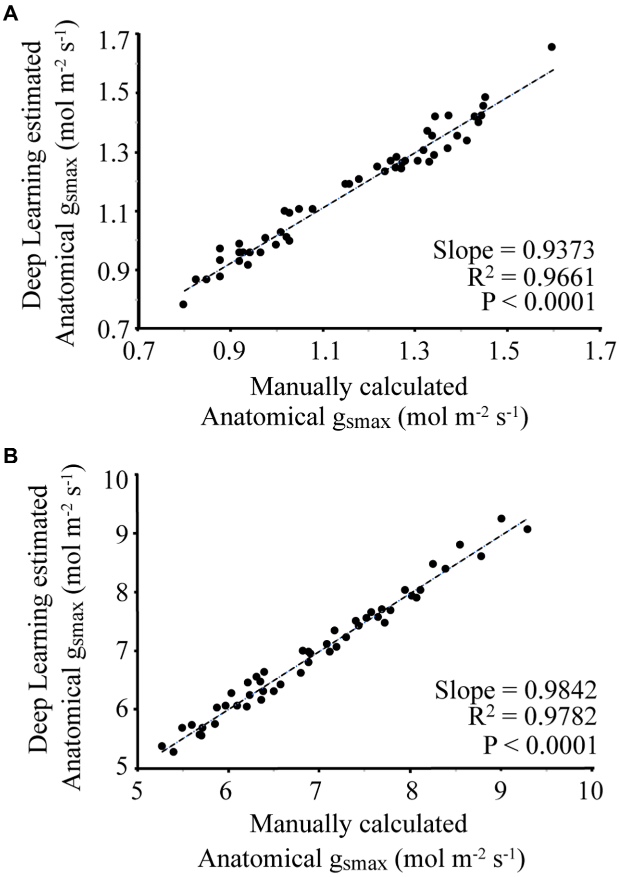


FIGURE 5 | Comparison of manual calculation of g_{smax} by an expert vs. automatic calculation using the proposed deep learning approach where **(A)** corresponds to the wheat dataset, **(B)** is the poplar dataset.

derived from fossils (Hetherington and Woodward, 2003; Franks and Beerling, 2009). This and similar approaches could prove useful in understanding and modelling future vegetation dynamics in climates with altered CO_2 and water vapour. In the case of crop phenotyping, individual leaf gas exchange on large numbers of lines is impractical, making a functional prediction from image-based data invaluable. However, g_{smax} does not always correlate well with measured leaf g_s due to variation in aperture. Despite this, measurement of the actual pore area, as opposed to maximal pore area, permits the calculation of operational g_s (Dow et al., 2014). However, such comparisons require careful consideration of both the conditions of measurement and the accuracy of the pore area estimation from a two-dimensional image made using light microscopy. The stomatal guard cell complex should really be considered in three dimensions relative to the surrounding cell structure, with the possibility of sunken or raised pores, whilst thickening of the guard cell wall may blur the calculation of the actual pore area. Finally, the means of taking the impression itself leads to uncertainty: after the resin or varnish has been applied there is a period of many minutes needed for drying (depending on temperature), which has unknown effects on stomatal aperture. Thus, the calculation of operational g_{smax} requires care. If such problems can be overcome, then this method provides opportunities to predict function from purely morphometric analysis and may be amenable to in-field instrumentation. By linking operational g_{smax} with mechanistic models of leaf gas exchange and environmental conditions, a prediction of photosynthetic rate would become possible.

There is a vast amount of literature relating to the extraction of stomata data from 2D images, the most recent and relevant of which are presented and compared to the current method in **Table 3**. Accuracy is not directly compared as each individual approach uses a different dataset and methods vary between papers. Dependent on the phenotyping task, each of these methods could be of use however none of the approaches explicitly output a g_{smax} calculation, which relies upon pixel segmentation, orientation of the stomata, and individual measurements of pore and guard cell. Also the method presented here, whilst limited to stomata, offers a solution that requires no tuning of parameters or user interaction to determine the optimal network.

The proposed method provides many advantages over manually obtaining morphological measurements, not least the time in which it takes to calculate g_{smax}. Unlike manual measurements, an automated approach allows for repeatability and a higher level of accuracy without bias, particularly beneficial for stomata phenotyping due to user-dependent variation in morphometric measurements. The time taken to calculate g_{smax} for a single image is less than a second regardless of the number of stomata present, substantially less than a manual approach. This may prove to be many hundreds of times faster with little manpower required. For example, it may take ~5–10 min per sample to count manually, with longer timespans required to measure dimensions. In a high-throughput phenotyping context with many thousands of samples this is difficult or impossible to achieve with limited human resources. We improve on existing works achieving 100% accuracy for stomata counting and obtain g_{smax} results that are within 4% of the manual measurements calculated by an expert. Furthermore, the pipeline can be applied to different species or varieties, currently applicable to the poplar and wheat but easily expandable with addition of an increasing number of datasets.

Historical trends in stomatal density using herbarium specimens have shown that rising CO_2 coincide with a reduction in stomatal density (Woodward, 1987; Hetherington and Woodward, 2003). Genetic manipulation has shown that changes in the size:density ratio can lead to changes in growth and WUE either through the improved uptake of CO_2 or *via* reducing water loss (Lawson and Blatt, 2014; Franks et al., 2015; Bertolino et al., 2019). Recently, it was discovered that reducing frequency in multiple crop plant species resulted in an enhancement of WUE with no cost to photosynthesis or yield (Nadal and Flexas, 2019). Therefore, understanding and manipulating this relationship are vital for sustaining or improving crop yields under global climate change, especially in regions

TABLE 3 | Comparison of the proposed method and output compared to other recently published methods.

Method	Overview	Output
Proposed method	A convolutional neural network based on semantic segmentation and image processing tool for morphometric calculations of stomata plus the automatic estimation of g_{smax} Applied to Poplar and Wheat	Pixelwise detection Count Density Pore measurements Guard cell measurements g_{smax} estimate
Toda et al., 2018 DeepStomata	Developed software comprising histogram of gradients (HOG) detection of stomata followed by region classification by a CNN. Used for stomatal pore quantification. Applied to Dayflower	Pixelwise detection Count Density Classification between open and closed stomata Pore measurements
Bhugra et al., 2019	Detects and quantifies stomata using a CNN and a series of image processing techniques Applied to Rice using scanning electron microscopy (SEM) images	Bounding box detection Count Density
Fetter et al., 2019 StomataCounter	A CNN for counting stomata, which detects bounding boxes that encapsulate the stomata Applied to Ginkgo and Poplar	Bounding box detection Stomata count Density
Andayani et al., 2020	Uses a CNN and image processing for classifying stomata into one of two groups belonging to either turmeric or ginger	Classification
Casado-García et al., 2020 LabelStoma	Use YOLO (Redmon and Farhadi, 2018) to detect bounding boxes Applied to Common Bean, Barley, and Soybean	Bounding box detection Stomata count Density
Kwong et al., 2021	A CNN applied specifically towards detecting stomata from Oil Palm	Bounding box detection Count Density
Toda et al., 2021	A platform that supports real time stomata detection when directly connected to a microscope Applied to Wheat- *N.B. measurements of bounding boxes allow morphometric calculations of stomata when orientated parallel or perpendicular to the field of view*	Bounding box detection Count Density Bounding box measures
Zhu et al., 2021	Applies R-CNN, U-Net, and image processing to calculate stomatal index Applied to Wheat	Bounding box detection Counts of stomata and epidermal cells Stomatal index calculation
Gómez-de-Mariscal et al., 2021 DeepImageJ	A plugin for the widely used ImageJ application. Brings a sophisticated method for integrating deep learning with ImageJ. A user friendly interface which supports a wide range of phenotyping tasks	Dependent on the network but also on the user for defining and selecting the best choice for their needs. Will give detection and possible measurements but no automatic calculation of indices without an additional step

dominated by heat and drought conditions and where precipitation patterns are shifting, and advanced methods to automatically calculate this will become increasingly important (Prasad et al., 2008; Asseng et al., 2015; Hughes et al., 2017; Caine et al., 2019; Dunn et al., 2019; Mohammed et al., 2019). This will allow for the rapid identification of anatomical traits for multiple applications including the acceleration and exploitation of variation in large-scale crop populations, for example in heat and drought dominated regions where higher WUE is essential to increase crop yields, analysis of stored specimens such as herbariums and palaeobotanical samples (Araus et al., 2002).

Application to General Research

The method presented here can be readily applied to new datasets. The key constraint, as with all deep learning methods, is the required annotated dataset; a network cannot find what it has not already "seen." This can be easily accomplished using the Pixel Annotation Tool used within this study to manually classify the guard cells and pore (Bréhéret, 2017). The network itself was generated for novice users, although access to a graphical processing unit (GPU) is required. Sample files and further instructions can be found on github.[1]

Whilst it is still quicker and more efficient to annotate a dataset to apply to future samples, the obvious next step would be to reduce the bottleneck associated with manual annotations. Future work could look at the use of Generative adversarial networks (GANs; Goodfellow et al., 2014), which generate

[1] http://github.com/drjonog

artificial annotations from a series of smaller datasets to reduce the overhead of training a network.

Previously stomatal conductance and related traits (i.e., transpiration, evapotranspiration, and photosynthesis) have been correlated in natural and crop ecosystems to remote sensing traits such as reflectance ratio R701/R820 as a response to photosynthesis and chlorophyll content in the leaves (Carter, 1998), Enhanced Vegetation Index (EVI), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Infrared Index (NDII) in water scarce regions (Carter, 1998; Glenn et al., 2008; Joiner et al., 2018) or infrared thermography and water indices (Gutierrez et al., 2010). However, none of these remote sensing methods, whilst allowing direct means of assessing canopy function, permit a means of selecting specifically for stomatal anatomy traits, which must require analysis at the cellular level. The rapid estimations of g_{smax} proposed in this study can facilitate breeding programs especially in arid and semi-arid countries were WUE is the most important trait for yield improvement.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: www.jonathongibbs.com/stomata2021 and www.github.com/drjonog.

AUTHOR CONTRIBUTIONS

JG, EM, and AB conceived the work. JG designed the model and computational framework, carried out implementation, performed the calculations and took the lead in writing the manuscript. CAR-Z collected the wheat dataset. JG and AB annotated the datasets. LM performed numerical calculations and assisted with stomatal biology. AB and JG designed and produced all figures for the paper. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Biotechnology and Biological Sciences Research Council (grant number BB/R004633/1). AB was supported by the Leverhulme Trust. CR-Z acknowledges financial support for his PhD scholarship to CONACYT, Mexico (CVU 626989). AB and JG wish to acknowledge support under the NVIDIA Academic Grant scheme.

ACKNOWLEDGMENTS

We wish to thank Dr. Matthew Reynolds and Dr. Gemma Molero (CIMMYT) for access to wheat plant material from the PS Tails panel.

REFERENCES

Andayani, U., Sumantri, I., Pahala, A., and Muchtar, M. (2020). The implementation of deep learning using convolutional neural network to classify based on stomata microscopic image of curcuma herbal plants. *IOP Conf. Ser. Mater. Sci. Eng.* 851:012035. doi: 10.1088/1757-899X/851/1/012035

Araus, J., Slafer, G., Reynolds, M., and Royo, C. (2002). Plant breeding and drought in C3 cereals: what should we breed for? *Ann. Bot.* 89, 925–940. doi: 10.1093/aob/mcf049

Asseng, S., Ewert, F., Martre, P., Rötter, R., Lobell, D., Cammarano, D., et al. (2015). Rising temperatures reduce global wheat production. *Nat. Clim. Chang.* 5, 143–147. doi: 10.1038/nclimate2470

Berman, M., Rannen, A., Matthew, T., and Blaschko, B. (2018). "The Lovász-Softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks." in *2018 IEEE/CVF Conference Computer Vision Pattern Recognition*; June 18–23, 2018; 4413–4421.

Berry, J., Beerling, D., and Franks, P. (2010). Stomata: key players in the earth system, past and present. *Curr. Opin. Plant Biol.* 13, 232–239. doi: 10.1016/j.pbi.2010.04.013

Bertolino, L., Caine, R., and Gray, J. (2019). Impact of stomatal density and morphology on water-use efficiency in a changing world. *Front. Plant Sci.* 10:225. doi: 10.3389/fpls.2019.00225

Bhugra, S., Mishra, D., Anupama, A., Chaudhury, S., Lall, B., Chugh, A., et al. (2019). "Deep convolutional neural networks based framework for estimation of stomata density and structure from microscopic images." in *Computer Vision- ECCV 2018 Workshop;* September 8–14, 2019; 412–423.

Bréhéret, A. (2017). Pixel Annotation Tool. Available at: https://github.com/abreheret/PixelAnnotationTool (Accessed April 18, 2021).

Buckley, T. (2005). The control of stomata by water balance. *New Phytol.* 168, 275–292. doi: 10.1111/j.1469-8137.2005.01543.x

Caine, R., Yin, X., Sloan, J., Harrison, E., Mohammed, U., Fulton, T., et al. (2019). Rice with reduced stomatal density conserves water and has improved drought tolerance under future climate conditions. *New Phytol.* 221, 371–384. doi: 10.1111/nph.15344

Carter, G. (1998). Reflectance wavebands and indices for remote estimation of photosynthesis and stomatal conductance in pine canopies. *Remote Sens. Environ.* 63, 61–72. doi: 10.1016/S0034-4257(97)00110-7

Casado-García, A., Del-Canto, A., Sanz-Saez, A., Pérez-López, U., Bilbao-Kareaga, A., Fritschi, F., et al. (2020). LabelStoma: A tool for stomata detection based on the YOLO algorithm. *Comput. Electron. Agric.* 178:105751. doi: 10.1016/j.compag.2020.105751

Casson, S., and Gray, J. (2007). Influence of environmental factors on stomatal development. *New Phytol.* 178, 9–23. doi: 10.1111/j.1469-8137.2007.02351.x

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

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai, L., and Li, F.-F. (2010). "ImageNet: a large-scale hierarchical image database." in *2009 IEEE Conference on Computer Vision and Pattern Recognition*; June 20–25, 2009; 248–255.

Dow, G., Bergmann, D., and Berry, J. (2014). An integrated model of stomatal development and leaf physiology. *New Phytol.* 201, 1218–1226. doi: 10.1111/nph.12608

Dunn, J., Hunt, L., Afsharinafar, M., Meselmani, M., Mitchell, A., Howells, R., et al. (2019). Reduced stomatal density in bread wheat leads to increased water-use efficiency. *J. Exp. Bot.* 70, 4737–4748. doi: 10.1093/jxb/erz248

Fetter, K., Eberhardt, S., Barclay, R., Wing, S., and Keller, S. (2019). StomataCounter: a neural network for automatic stomata identification and counting. *New Phytol.* 223, 1671–1681. doi: 10.1111/nph.15892

Fischer, R., Rees, D., Sayre, K., Lu, Z., Condon, A., and Larque Saavedra, A. (1998). Wheat yield progress associated with higher stomatal conductance and photosynthetic rate, and cooler canopies. *Crop Sci.* 38, 1467–1475. doi: 10.2135/cropsci1998.0011183X003800060011x

Franks, P., and Beerling, D. (2009). Maximum leaf conductance driven by CO2 effects on stomatal size and density over geologic time. *Proc. Natl. Acad. Sci. U. S. A.* 106, 10343–10347. doi: 10.1073/pnas.0904209106

Franks, P., Doheny-Adams, W. T., Britton-Harper, Z., and Gray, J. (2015). Increasing water-use efficiency directly through genetic manipulation of stomatal density. *New Phytol.* 207, 188–195. doi: 10.1111/nph.13347

Franks, P., and Farquhar, G. (2007). The mechanical diversity of stomata and its significance in gas-exchange control. *Plant Physiol.* 143, 78–87. doi: 10.1104/pp.106.089367

Glenn, E., Huete, A., Nagler, P., and Nelson, S. (2008). Relationship between remotely-sensed vegetation indices, canopy attributes and plant physiological processes: what vegetation indices can and cannot tell us about the landscape. *Sensors* 8, 2136–2160. doi: 10.3390/s8042136

Gómez-de-Mariscal, E., García-López-de-Haro, C., Donati, L., Unser, M., Muñoz-Barrutia, A., and Sage, D. (2021). DeepImageJ: a user-friendly plugin to run deep learning models in ImageJ. *Nat. Methods* 18, 1192–1195. doi: 10.1038/s41592-021-01262-9

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *Adv. Neural Inf. Proces. Syst.* 27, 2672–2680.

Gutierrez, M., Reynolds, M., and Klatt, A. (2010). Association of water spectral indices with plant and soil water relations in contrasting wheat genotypes. *J. Exp. Bot.* 61, 3291–3303. doi: 10.1093/jxb/erq156

Hetherington, A. M., and Woodward, F. I. (2003). The role of stomata in sensing and driving environmental change. *Nature* 424, 901–908. doi: 10.1038/nature01843

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv [Preprint].

Hughes, J., Hepworth, C., Dutton, C., Dunn, J., Hunt, L., Stephens, J., et al. (2017). Reducing stomatal density in barley improves drought tolerance without impacting on yield. *Plant Physiol.* 174, 776–787. doi: 10.1104/pp.16.01844

Jiang, F., Grigorev, A., Rho, S., Tian, Z., Fu, Y. S., Jifara, W., et al. (2018). Medical image semantic segmentation based on deep learning. *Neural Comput. Applic.* 29, 1257–1265. doi: 10.1007/s00521-017-3158-6

Joiner, J., Yoshida, Y., Anderson, M., Holmes, T., Hain, C., Reichle, R., et al. (2018). Global relationships among traditional reflectance vegetation indices (NDVI and NDII), evapotranspiration (ET), and soil moisture variability on weekly timescales. *Remote Sens. Environ.* 219, 339–352. doi: 10.1016/j.rse.2018.10.020

Kiefer, J., and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* 23, 462–466. doi: 10.1214/aoms/1177729392

Kwong, Q., Wong, Y., Lee, P., Sahaini, M., Kon, Y., Kulaveerasingam, H., et al. (2021). Automated stomata detection in oil palm with convolutional neural network. *Sci. Rep.* 11:15210. doi: 10.1038/s41598-021-94705-4

Lammertsma, E., De Boer, H., Dekker, S., Dilcher, D., Lotter, A., and Wagner-Cremer, F. (2011). Global CO2 rise leads to reduced maximum stomatal conductance in Florida vegetation. *Proc. Natl. Acad. Sci. U. S. A.* 108, 4035–4040. doi: 10.1073/pnas.1100371108

Lawson, T., and Blatt, M. (2014). Stomatal size, speed, and responsiveness impact on photosynthesis and water use efficiency. *Plant Physiol.* 164, 1556–1570. doi: 10.1104/pp.114.237107

Maier, A., Syben, C., Lasser, T., and Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Z. Med. Phys.* 29, 86–101. doi: 10.1016/j.zemedi.2018.12.003

McAusland, L., Vialet-Chabrand, S., Davey, P., Baker, N., Brendel, O., and Lawson, T. (2016). Effects of kinetics of light-induced stomatal responses on photosynthesis and water-use efficiency. *New Phytol.* 211, 1209–1220. doi: 10.1111/nph.14000

Mcelwain, J., Yiotis, C., and Lawson, T. (2016). Using modern plant trait relationships between observed and theoretical maximum stomatal conductance and vein density to examine patterns of plant macroevolution. *New Phytol.* 209, 94–103. doi: 10.1111/nph.13579

Meeus, S., Van den Bulcke, J., and Wyffels, F. (2020). From leaf to label: a robust automated workflow for stomata detection. *Ecol. Evol.* 10, 9178–9191. doi: 10.1002/ece3.6571

Mohammed, U., Caine, R., Atkinson, J., Harrison, E., Wells, D., Chater, C. C., et al. (2019). Rice plants overexpressing OsEPF1 show reduced stomatal density and increased root cortical aerenchyma formation. *Sci. Rep.* 9:5584. doi: 10.1038/s41598-019-41922-7

Murray, M., Soh, W., Yiotis, C., Spicer, R., Lawson, T., and McElwain, J. (2020). Consistent relationship between field-measured stomatal conductance and theoretical maximum stomatal conductance in C3 woody angiosperms in four major biomes. *Int. J. Plant Sci.* 181, 142–154. doi: 10.1086/706260

Nadal, M., and Flexas, J. (2019). Variation in photosynthetic characteristics with growth form in a water-limited scenario: implications for assimilation rates and water use efficiency in crops. *Agric. Water Manag.* 216, 457–472. doi: 10.1016/j.agwat.2018.09.024

Oktay, O., Schlemper, J., Folgoc, L., Lee, M., Heinrich, M., Misawa, K., et al. (2018). Attention U-Net: Learning Where to Look for the Pancreas. arXiv [Preprint].

Parlange, J.-Y., and Waggoner, P. (1970). Stomatal dimensions and resistance to diffusion. *Plant Physiol.* 46, 337–342. doi: 10.1104/pp.46.2.337

Prasad, P., Staggenborg, S., and Ristic, Z. (2008). "Impacts of drought and/or heat stress on physiological, developmental, growth, and yield processes of crop plants," in *Response of Crops to Limited Water: Understanding and Modelling Water Stress Effects on Plant Growth Process. Vol. 1.* eds. L. Ahuja and American Society of Agronomy (ASA-CSSA-SSSA), 301–355.

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

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation." in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics);* October 5-9, 2015; Springer Verlag; 234–241.

Rudall, P., Hilton, J., and Bateman, R. (2013). Several developmental and morphogenetic factors govern the evolution of stomatal patterning in land plants. *New Phytol.* 200, 598–614. doi: 10.1111/nph.12406

Sack, L., and Buckley, T. (2016). The developmental basis of stomatal density and flux. *Plant Physiol.* 171, 2358–2363. doi: 10.1104/pp.16.00476

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* 97, 676–682. doi: 10.1038/nmeth.2019

Siam, M., Gamal, M., Abdel-Razek, M., Yogamani, S., Jagersand, M., and Zhang, H. (2018). "A comparative study of real-time semantic segmentation for autonomous driving." in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW);* June 18-22, 2018; 700–710.

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition." in *3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings*; May 7–9, 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision." in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR);* Las Vegas, NV, USA, 2818–2826.

Toda, Y., Tameshige, T., Tomiyama, M., Kinoshita, T., and Shimizu, K. (2021). An affordable image-analysis platform to accelerate stomatal phenotyping During microscopic observation. *Front. Plant Sci.* 12:715309. doi: 10.3389/fpls.2021.715309

Toda, Y., Toh, S., Bourdais, G., Robatzek, S., Maclean, D., and Kinoshita, T. (2018). DeepStomata: facial recognition technology for automated stomatal aperture measurement. bioRxiv [Preprint]. doi: 10.1101/365098

Weyers, J., and Johansen, L. (1990). *Methods of Stomatal Research*. Harlow, Essex, England: Longman Scientific & Technical.

Woodward, F. (1987). Stomatal numbers are sensitive to increases in CO2 from pre-industrial levels. *Nature* 327, 617–618. doi: 10.1038/327617a0

Wurm, M., Stark, T., Zhu, X. X., Weigand, M., and Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* 150, 59–69. doi: 10.1016/j.isprsjprs.2019.02.006

Zhu, C., Hu, Y., Mao, H., Li, S., Li, F., Zhao, C., et al. (2021). A deep learning-based method for automatic assessment of stomatal index in wheat microscopic images of leaf epidermis. *Front. Plant Sci.* 12:1895. doi: 10.3389/fpls.2021.716784

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

METHODS
published: 14 December 2021
doi: 10.3389/fpls.2021.770916

Check for updates

Automatic and Accurate Calculation of Rice Seed Setting Rate Based on Image Segmentation and Deep Learning

Yixin Guo[1], Shuai Li[1], Zhanguo Zhang[2], Yang Li[2], Zhenbang Hu[3], Dawei Xin[3], Qingshan Chen[3]*, Jingguo Wang[3]* and Rongsheng Zhu[2]*

[1] College of Engineering, Northeast Agricultural University, Harbin, China, [2] College of Arts and Sciences, Northeast Agricultural University, Harbin, China, [3] Agricultural College, Northeast Agricultural University, Harbin, China

The rice seed setting rate (RSSR) is an important component in calculating rice yields and a key phenotype for its genetic analysis. Automatic calculations of RSSR through computer vision technology have great significance for rice yield predictions. The basic premise for calculating RSSR is having an accurate and high throughput identification of rice grains. In this study, we propose a method based on image segmentation and deep learning to automatically identify rice grains and calculate RSSR. By collecting information on the rice panicle, our proposed image automatic segmentation method can detect the full grain and empty grain, after which the RSSR can be calculated by our proposed rice seed setting rate optimization algorithm (RSSROA). Finally, the proposed method was used to predict the RSSR during which process, the average identification accuracy reached 99.43%. This method has therefore been proven as an effective, non-invasive method for high throughput identification and calculation of RSSR. It is also applicable to soybean yields, as well as wheat and other crops with similar characteristics.

Keywords: rice grain identification, computer vision, deep learning, rice seed setting rate, image segmentation

OPEN ACCESS

Edited by:
Wanneng Yang,
Huazhong Agricultural University,
China

Reviewed by:
Marcin Wozniak,
Silesian University of Technology,
Poland
Michael Gomez Selvaraj,
Consultative Group on International
Agricultural Research (CGIAR),
United States
Lejun Yu,
Hainan University, China

Correspondence:
Qingshan Chen
qshchen@126.com
Jingguo Wang
wangjg@neau.edu.cn
Rongsheng Zhu
rshzhu@126.com

Specialty section:
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

Received: *05 September 2021*
Accepted: *23 November 2021*
Published: *14 December 2021*

Citation:
Guo Y, Li S, Zhang Z, Li Y, Hu Z,
Xin D, Chen Q, Wang J and Zhu R
(2021) Automatic and Accurate
Calculation of Rice Seed Setting Rate
Based on Image Segmentation
and Deep Learning.
Front. Plant Sci. 12:770916.
doi: 10.3389/fpls.2021.770916

INTRODUCTION

Rice (Oryza sativa) is a cereal grain and the most widely consumed staple food for a large part of the world's human population, especially in Asia (Ghadirnezhad and Fallah, 2014). The number of rice grains per panicle is a key trait that effects grain cultivation, management, and subsequent yield (Wu et al., 2019). The grains per panicle are usually divided into two categories, one is full grain and the other is empty grain. Among them, full grain is the real measure of the number of grains per panicle, and the ratio of full grain to the total number of grains per panicle is called the seed setting rate. The number of grains per panicle and the seed setting rate are considered to be the two most important traits directly reflecting rice yield (Oosterom and Hammer, 2008; Gong et al., 2018).

Generally, grain weight, grain number, panicle number, and RSSR are considered to be the main factors affecting rice yield. However, research into RSSR is improving with the advancements in science and technology. Li et al. (2013) have shown that the domestication-related POLLEN TUBE BLOCKED 1 (PTB1), a RING-type E3 ubiquitin ligase, positively regulates the rice seed setting rate by promoting pollen tube growth. Xu et al. (2017) proposed that OsCNGC13 acts as a novel maternal sporophytic factor required for stylar $[Ca^2]_{cyt}$ accumulation, ECM components modification, and STT cell death, and thus facilitates the penetration of the pollen tube for successful double fertilization and seed setting in rice. Xiang et al. (2019) reported on a novel rice gene, LOW SEED SETTING RATE1 (LSSR1), which regulates the seed setting rate by facilitating rice fertilization. Through these studies and their achievements, improving the RSSR has become an expected thing. However, a new issue has arisen with them, a problem posed by the automatic high-throughput calculation of the RSSR.

With developments in deep learning and plant phenotypic science, efficient and accurate research on rice through information technology (IT) has become very anticipated. Desai et al. (2019) proposed a simple pipeline which uses ground level RGB images of paddy rice to detect which regions contain flowering panicles, and then uses the flowering panicle region count to estimate the heading date of the crop. Hong Son and Thai-Nghe (2019) proposed an approach for rice quality classification. In their approach, image processing algorithms, the convolutional neural network (CNN), and machine learning methods are used to recognize and classify two different categories of rice (whole rice and broken rice), based on rice sizes according to the national standard of rice quality evaluation. Lin et al. (2018) proposed a machine vision system based on the deep convolutional neural network (DCNN) architecture to improve, compared with traditional approaches, the accuracy with which three distinct groups of rice kernel images are classified. Xu et al. (2020) proposed a simple, yet effective method termed the Multi-Scale Hybrid Window Panicle Detect (MHW-PD), which focuses on enhancing the panicle features to then detect and count the large number of small-sized rice panicles in the in-field scene. Chatnuntawech et al. (2018) developed a non-destructive rice variety classification system that benefits from the synergy between hyperspectral imaging and the deep CNN. The rice varieties are then determined from the acquired spatio-spectral data using a deep CNN. Zhou et al. (2019) developed and implemented a panicle detection and counting system based on improved region-based fully convolutional networks, and used the system to automate rice-phenotype measurements. Lu et al. (2017) proposed an innovative technique to enhance the deep learning ability of CNNs. The proposed CNN-based model can effectively classify 10 common rice diseases through image recognition technology. Chu and Yu (2020) constructed a novel end-to-end model based on deep learning fusion to accurately predict the rice yields for 81 counties in the Guangxi Zhuang Autonomous Region, China, using a combination of time-series meteorology data and area data. Xiong et al. (2017) proposed a rice panicle segmentation algorithm called Panicle-SEG, which is based on the generation of simple linear iterative clustering super pixel regions, CNN classification, and entropy rate super pixel optimization. Kundu et al. (2021) develop the "Automatic and Intelligent Data Collector and Classifier" framework by integrating IoT and deep learning. The framework automatically collects the imagery and parametric data and automatically sends the collected data to the cloud server and the Raspberry Pi. It collaborates with the Raspberry Pi to precisely predict the blast and rust diseases in pearl millet. Dhaka et al. (2021) present a survey of the existing literature in applying deep CNNs to predict plant diseases from leaf images. This manuscript presents an exemplary comparison of the pre-processing techniques, CNN models, frameworks, and optimization techniques applied to detect and classify plant diseases using leaf images as a data set.

RSSR was initially calculated manually. However, Kong and Chen (2021) proposed a method based on a mask region convolutional neural network (Mask R-CNN) for feature extraction and three- dimensional (3-D) recognition in CT images of rice panicles, and then calculated the seed setting rate through the obtained three-dimensional image. However, due to the difficulty and high cost of CT image acquisition, this method lacks practicality.

In our research, we closely link deep learning with RSSR, making it a portable tool for the automatic and high-throughput study of RSSR. Through experimental verification, we have found that the correlation between our proposed RSSROA and the results from manual RSSR calculations is as high as 93.21%. In addition, through the verification of 10 randomly selected rice panicle images, our proposed method has been shown to be able to correctly distinguish between two kinds of rice grains. The average accuracy of the number of full grains per panicle is 97.69% and the average accuracy of the number of empty grains per panicle is 93.20%. Therefore, our proposed method can effectively detect two different grains in rice panicles and can accurately calculate RSSR. It can thus become an effective method for low-cost, high-throughput calculations of RSSR.

MATERIALS AND METHODS

An overview of the proposed method can be seen in **Figure 1**. The input to our system consists of a sequence of images (across different days and times) of different rice varieties taken in a particular environment (**Supplementary Table 1**). The collected images were first cropped to give them the best possible resolution for the network input, and then they were input into the deep learning network we adopted for training after calibration. The training results from each network were compared, and the best network was adopted as the method to calculate the RSSR.

Image Acquisition and Processing

Rice planting was carried out in both 2018 and 2019 at Northeast Agricultural University's experimental practice and demonstration base in Acheng, which is located at an east longitude of 127°22′∼127°50′ and north latitude of 45°34′∼45°46′. The test soil was black soil, and there were protection and isolation rows around each 20 m^2 plot area. The seeds were sown on April 20, 2018 (April 17 for the 2019 crop) and transplanted on May 20, 2018 (May 24 for the 2019 crop). The transplanting size was 30 cm × 10 cm and the field management was the same as for the production field (Zhao et al., 2020).

In order to improve the generalization ability of the experiment and reduce the time required for the artificial labeling of rice grains, 56 varieties of rice were randomly selected from the experimental field and the rice panicle information was collected

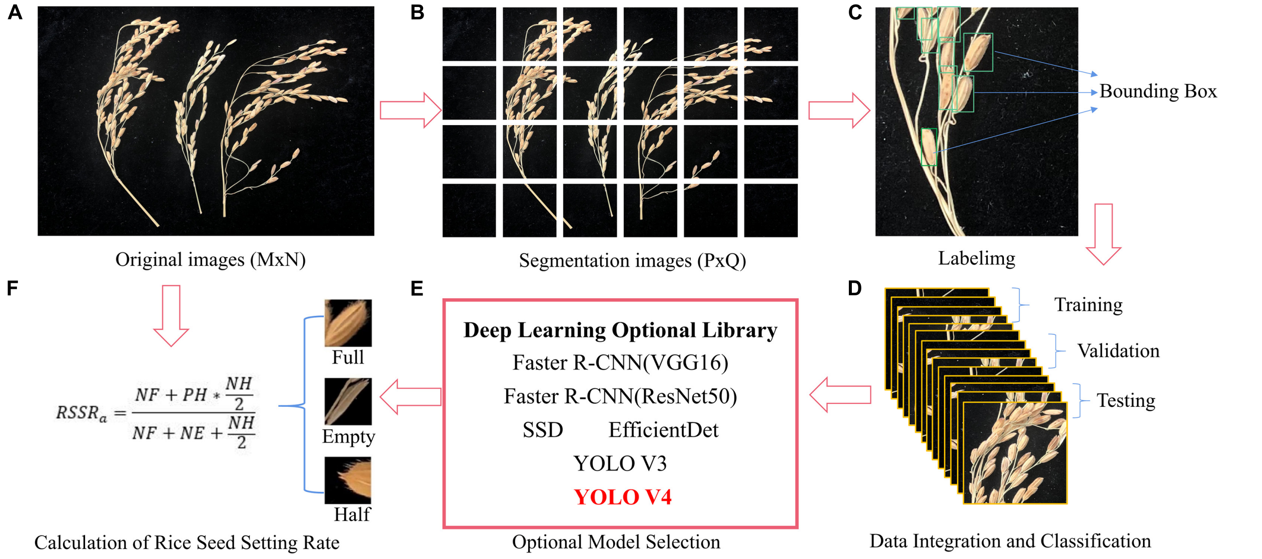

A
Original images (MxN)
B
Segmentation images (PxQ)
C
Bounding Box
Labelimg
D
Training
Validation
Testing
Data Integration and Classification
E
Deep Learning Optional Library
Faster R-CNN(VGG16)
Faster R-CNN(ResNet50)
SSD
EfficientDet
YOLO V3
YOLO V4
Optional Model Selection
F
$RSSR_a = \frac{NF + PH * \frac{NH}{2}}{NF + NE + \frac{NH}{2}}$
Full
Empty
Half
Calculation of Rice Seed Setting Rate

FIGURE 1 | Research flow diagram. **(A)** Original images **(B)** Segmentation images **(C)** Labelimg **(D)** Data integration and classification **(E)** Optional model selection **(F)** Calculation of rice seed setting rate.

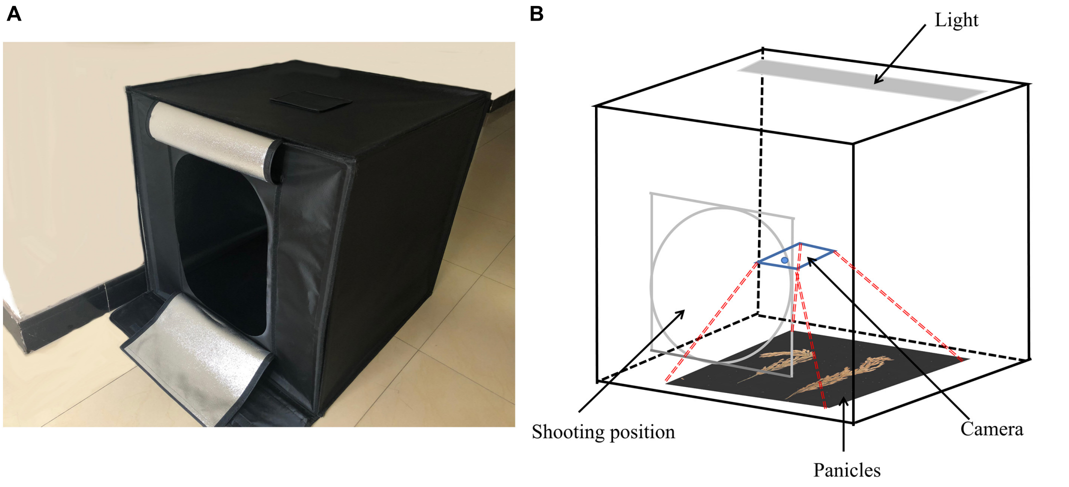


FIGURE 2 | Rice panicle image collection cubed darkroom. **(A)** Real map and **(B)** structural diagram.

using a smartphone iPhone X. The image collection environment consisted in a cubed darkroom with a length, width, and height all measuring 80 cm. The top of the darkroom environment possessed a unique light source, while the other directions were all covered by all-black light-absorbing cloth. The shooting method was to artificially push the keys on the mobile phone from the oval entrance on the front of the cubed darkroom (a rectangle measuring 55 cm in length and 40 cm in width). The shooting equipment was kept about 30 cm from the top of the rice panicles (The shooting equipment is not fixed, it only needs to be maintained manually). The image collection cubed darkroom for the rice panicles is shown in **Figure 2**.

A total of 263 rice panicles and 298 images were obtained. Each panicle of rice is shot in both natural and artificially shaped states. Each image contains a different panicle of rice, at least one panicle of rice and at most four panicles of rice. The panicles of each rice variety ranged from 2 to 11. Among them, 60 images were used as the data to calculate the RSSR, while the remaining images were divided into a training verification set and a test set by a ratio of 8:2.

We calibrated the obtained images by labeling with a target detection marking tool, and then used these images for training and prediction purposes. **Figure 3A** shows the calibration difference between different data sets, and **Figure 3B** shows the detailed differences between various categories in the image cutting process, where "full" represents a full rice grain, "empty" represents an empty rice grain, "half" represents a half rice grain, "H-full" and "H-empty" represent

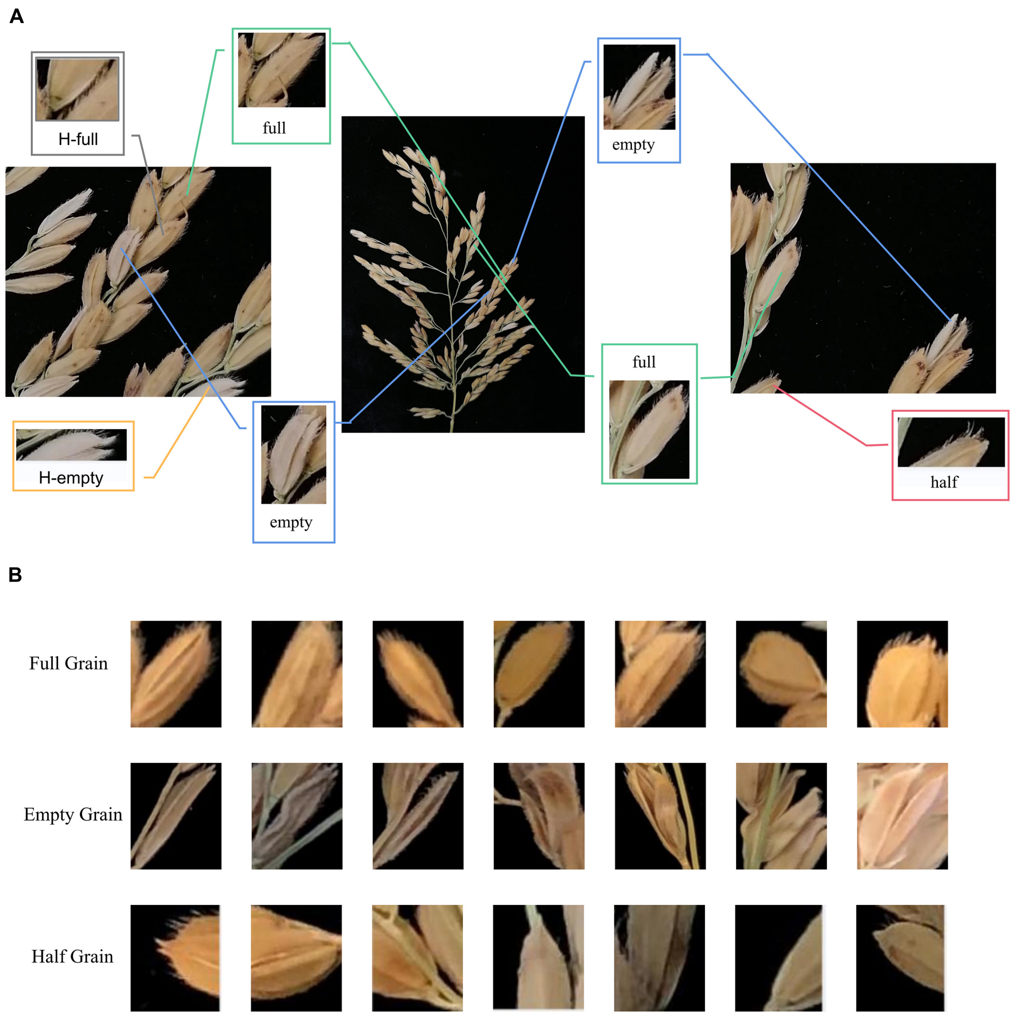


FIGURE 3 | Feature image for depth learning. **(A)** Comparison of local characteristics of rice grains, **(B)** comparison of grain characteristics of different rice varieties.

the full and empty grains detected in in the half grain count after cropping.

Convolutional Neural Network

The CNN consists of several layers of neurons and computes a multidimensional function with several variables (Chen et al., 2014; Schmidhuber, 2015). The neurons in each layer, other than from the first layer, are connected with the neurons from the preceding layer. The first layer is called the input layer (Zhang et al., 2015; Dong et al., 2016), which is then followed by hidden layers, and the concluding layer. Each neuron connection has a weight that is adjusted during the learning process. Initially, the weights are taken at random. All neurons receive input values, which they then process and send out as output values. The input layer neurons' input and output values are the values from the variables of the function. In the other layers meanwhile, a neuron receives at its input the weighted sum of the output values from the neurons with which the neuron in question is connected. The weights of the connections are used as the weights for the weighing process. Each neuron gives its function to an input value and these functions are called activation functions (LeCun et al., 2015; Mitra et al., 2017).

The motivation of building an Object Detection model is to provide solutions in the field of computer vision. The primary essence of object detection can be broken down into two parts: to locate objects in a scene (by drawing a bounding box around the object) and later to classify the objects (based on the classes it was trained on). There are two deep learning based approaches for object detection: one-stage methods (YOLO–You Only Look Once, SSD–Single Shot Detection) and two-stage approaches (Faster R-CNN) (Rajeshwari et al., 2019). In addition, we have added a newer one-stage object detector-EfficientDet. These will be our main research methods.

Faster Region Convolutional Neural Network

As a typical two-stage object detection algorithm, the faster region convolutional neural network (Faster R-CNN) has been widely applied in many fields since its proposal (Ren et al., 2016).

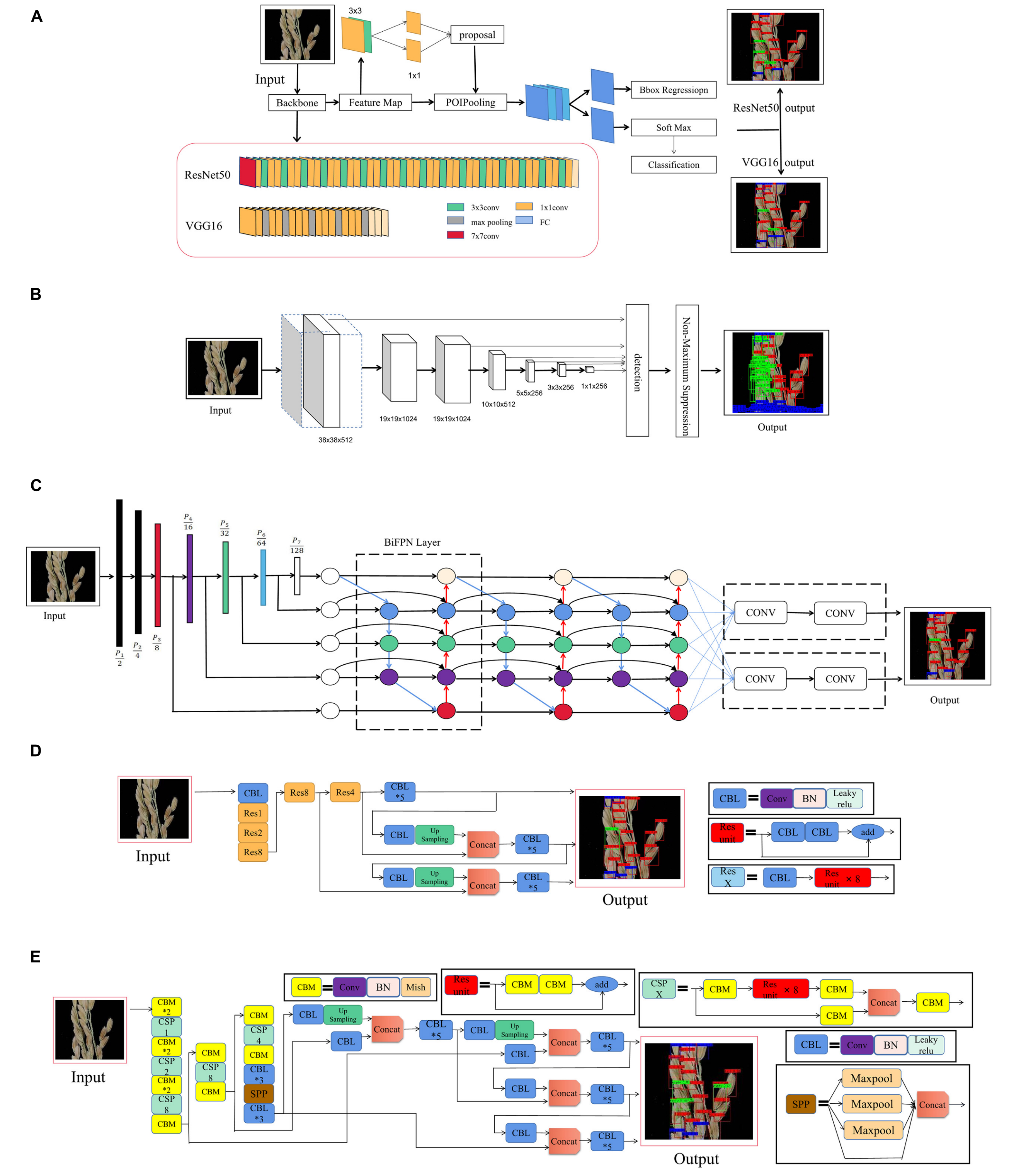


FIGURE 4 | Convolutional neural network. **(A)** Faster R-CNN, **(B)** SSD, **(C)** EfficientDet, **(D)** YOLO V3, and **(E)** YOLO V4.

As shown in **Figure 4A**, a region proposal network (RPN) is constructed to generate confident proposal for multi-classification and bounding box refinement. More precisely, RPN first generates a dense grid of anchor regions (candidate bounding boxes) with specified sizes and aspect ratios over each spatial location of the feature maps. According to intersection over union (IOU) ratio with the ground truth object bounding boxes, an anchor will be assigned with a positive or negative label on top of the feature maps, a shallow CNN is built to judge whether an anchor contains an object and predict an offset for each anchor. Then anchors with high confidence are rectified by the offset predicted in RPN. Then the corresponding features of each anchor will go through a RoI pooling layer, a convolution layer and a fully connected layer to predict a specific class as well as refined bounding boxes (Zou et al., 2020). In addition, it is worth noting that we use ResNet50 and VGG16 as the backbone networks for training.

Single Shot Detector

The single shot detector (SSD) (Liu et al., 2016) discretizes the bounding boxes' output space into a set of default boxes over different aspect ratios and scales per feature mAP location. At the predicted time, the network awards scores to the situation of each object category in each default box, after which, it makes the according adjustments to the box to better match the object shape. Additionally, in order to naturally handle objects of various sizes, the network combines predictions from multiple feature mAPs with different resolutions. SSD is simple compared to methods that require object proposals, because it completely eliminates the need for proposal generations and the subsequent pixel or feature resampling stages, and encapsulates all the necessary computations in a single network. This makes SSD easily trainable and straightforward to integrate into systems requiring a detection component (see **Figure 4B**).

EfficientDet

EfficientDet proposes a weighted bi-directional feature pyramid network (BiFPN) and then uses it as the feature network. It

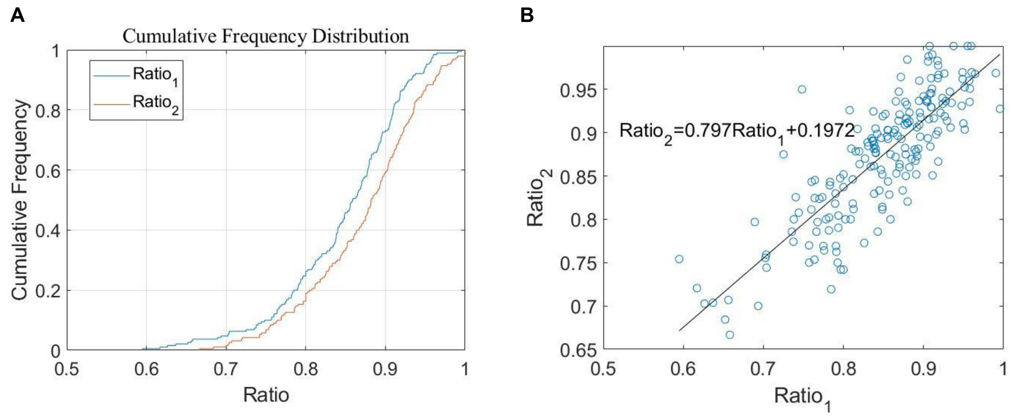


FIGURE 5 | Research on the relationship of Ratio. **(A)** The proportion of cumulative frequency according to the change of ratio **(B)** relationship between *Ratio*$_1$ and *Ratio*$_2$.

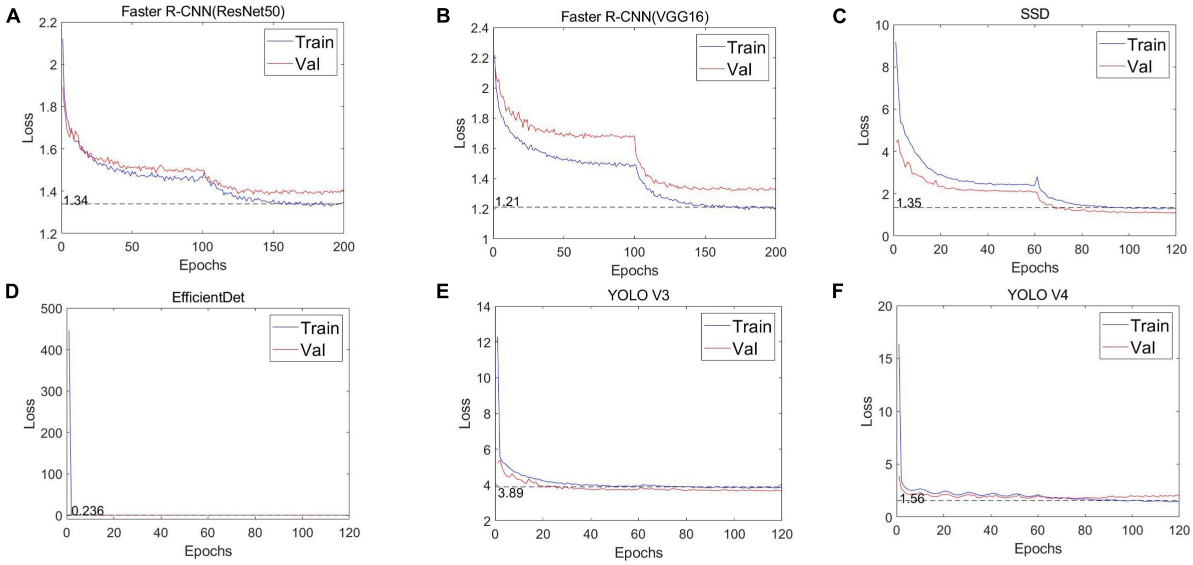


FIGURE 6 | Loss curves of the different CNNs. **(A)** Faster R-CNN (ResNet50), **(B)** Faster R-CNN (VGG16), **(C)** SSD, **(D)** EfficientDet, **(E)** YOLO V3, and **(F)** YOLO V4.

takes level 3–7 features (P3, P4, P5, P6, P7) from the backbone network and repeats the top-down and bottom-up bi-directional feature fusion. These fused features are fed to the class and box networks to generate object class and boundary box predictions, respectively. A composite scaling extension method is also proposed, which is able to uniformly scale the resolution, depth and width of all the backbone networks, feature networks and prediction networks. The network structure of EfficientDet is shown in **Figure 4C** (Tan et al., 2020).

You Only Look Once

YOLO V3 adopts a network structure called Darknet53. It draws on the practice of residual network, and sets up fast links between some layers to form a deeper network level and multi-scale detection, which improves the detection effect of mAP and small objects (Redmon and Farhadi, 2018). Its basic network structure is shown in **Figure 4D**.

The real-time and high-precision target detection model, YOLO V4, allows anyone training and testing with a conventional GPU to achieve real-time, high quality and convincing object detection results. As an improved version of YOLO V3, YOLO V4 combines many of the techniques from YOLO V3. Among them, the feature extraction network, Darknet53, which was the backbone network for YOLO V3, has been changed to CSPDarknet53, the feature pyramid has become SPP and PAN, while the classification regression layer remains the same as in YOLO V3. In order to achieve better target detection accuracy without increasing inference costs, a method is used that either only changes the training strategy or only increases the training cost. This method is called the "bag of freebies." A common method for target detection that meets the requirements of being a "free bag" in the "bag of freebies" method, is data enhancement. The purpose of data augmentation is to increase the variability of the input images, meaning that the designed object detection model will have higher robustness to images obtained in different environments. Another addition to this method, is known as the "bag of specials." This bag consists of plugin modules and a post-processing method that can significantly improve the accuracy of object detection and only increase the inference cost by a small amount. Generally speaking, these plugin modules are used to enhance certain attributes in a model, such as enlarging the receptive field, introducing an attention mechanism, or strengthening feature integration capability. Post-processing meanwhile, consists in a method used for screening model prediction results. Its basic network structure is shown in **Figure 4E** (Bochkovskiy et al., 2020).

Hardware and Software

The CNNs were trained on the rice image dataset using a hardware solution from our computer. This was a personal desktop computer with Intel core i9-9900k CPU, NVIDIA Titan XP (12G) GPU, and 64G RAM. We used the desktop to train the six networks in Python language under a Windows operating system with a Pytorch framework.

Rice Seed Setting Rate Optimization Algorithm

Obtaining the RSSR is the ultimate goal of this research. According to the traditional RSSR calculation formula used in agriculture, the following formula was offered for adaption to our research results:

$$RSSR_t = \frac{NF_t}{NF_t + NE_t} \tag{1}$$

We put forward a novel method to calculate the RSSR, which is to segment the original rice images to form the third category "half grain," and calculate the RSSR by finding the correlation among them. This method is called the rice seed setting rate optimization algorithm (RSSROA), the formula is as follows:

$$RSSR_a = \frac{NF + PH \times \frac{NH}{2}}{NF + NE + \frac{NH}{2}} \tag{2}$$

$$Ratio_1 = \frac{NF}{NF + NE} \tag{3}$$

$$Ratio_2 = \frac{NFH}{NFH + NEH} \tag{4}$$

where $RSSR_t$ is a traditional measurement method used for calculating the RSSR in agronomy, NF_t is the number of full grains obtained by traditional methods, NE_t is the number of empty grains obtained by traditional methods, $RSSR_a$ is the RSSR result calculated by our rice seed setting rate optimization algorithm (RSSROA), $NF(NUMBER\ OF\ FULL\ GRAIN)$ is the number of full rice grains obtained by RSSROA, $NE(NUMBER\ OF\ EMPTY\ GRAIN)$ is the number of empty

TABLE 1 | Detection performance of different models in the test set during the clipping stage.

Network name	Category	Precision	Recall	F1	AP	mAP
Faster R-CNN (ResNet50)	Full grain	74.24%	87.80%	0.80	84.10%	50.65%
	Empty grain	56.28%	56.21%	0.56	44.70%	
	Half grain	50.20%	32.95%	0.40	23.15%	
Faster R-CNN (VGG16)	Full grain	82.32%	88.43%	0.85	86.55%	59.70%
	Empty grain	61.07%	51.77%	0.56	46.10%	
	Half grain	69.35%	50.16%	0.58	46.45%	
SSD	Full grain	36.43%	71.47%	0.48	66.09%	31.01%
	Empty grain	10.24%	60.05%	0.18	17.87%	
	Half grain	3.18%	56.91%	0.06	9.08%	
EfficientDet	Full grain	79.43%	84.45%	0.82	86.99%	54.54%
	Empty grain	100.00%	0.02%	0.00	15.84%	
	Half grain	92.54%	27.26%	0.42	60.78%	
YOLO V3	Full grain	81.00%	84.07%	0.83	88.29%	62.62%
	Empty grain	60.12%	35.19%	0.44	40.84%	
	Half grain	83.94%	44.54%	0.58	58.72%	
YOLO V4	Full grain	89.79%	92.79%	0.91	94.78%	83.98%
	Empty grain	77.66%	74.68%	0.76	73.92%	
	Half grain	87.79%	75.83%	0.81	83.24%	

grains obtained by RSSROA, $NH(NUMBER\ OF\ HALF\ GRAIN)$ is the number of half grains obtained by RSSROA, $PH(PROBABILITY\ OF\ FULL\ HALF\ SEED)$ is the prior probability of there being full grains of rice in the half grain count, $NFH(NUMBER\ OF\ FULL\ GRAIN\ IN\ HALF\ GRAIN)$ is the number of full grains in the half grain count, and $NEH(NUMBER\ OF\ EMPTY\ GRAIN\ IN\ HALF\ GRAIN)$ is the number of empty grains in the half grain count.

Through our simulation study, it was found that there is a certain linear relationship between $Ratio_1$ and $Ratio_2$. This can be seen in **Figure 5A**, which shows the distribution density curves of $Ratio_1$ and $Ratio_2$, where both curves belong to normal distribution and have 99.89% probability of consistency by the Kolmogorov-Smirnov test (Frank, 1951). Therefore, we further explored and obtained the scatter diagram with $Ratio_1$ as the X-axis and $Ratio_2$ as the Y-axis, as shown in **Figure 5B**. Through a correlation analysis, we then obtained the correlation coefficient of 0.8327 and the linear equation of $PH = Ratio_2 = 0.797Ratio_1 + 0.1972$. The result of this current method can be used as our PH coefficient.

Evaluation Standard

We evaluated the results from the different networks used on our data set. For the evaluation, a detected instance was considered a true positive if it had a Jaccard Index similarity coefficient, also known as an intersection-over-union (IOU) (He and Garcia, 2009; Csurka et al., 2013) of 0.5 or more, with a ground truth instance. The IOU is defined as the ratio of pixel number in the intersection to pixel number in the union. The instances of ground truth which did not overlap with any detected instance were considered false negatives. From these measures, the precision, recall, F1 score, AP, and mAP were calculated (Afonso et al., 2020):

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2Precision \times Recall}{Precision + Recall} \tag{7}$$

$$AP = \sum_{k=1}^{N} Precision(k) \triangle Recall(k) \tag{8}$$

$$mAP = \frac{\sum_i^M AP_i}{M} \tag{9}$$

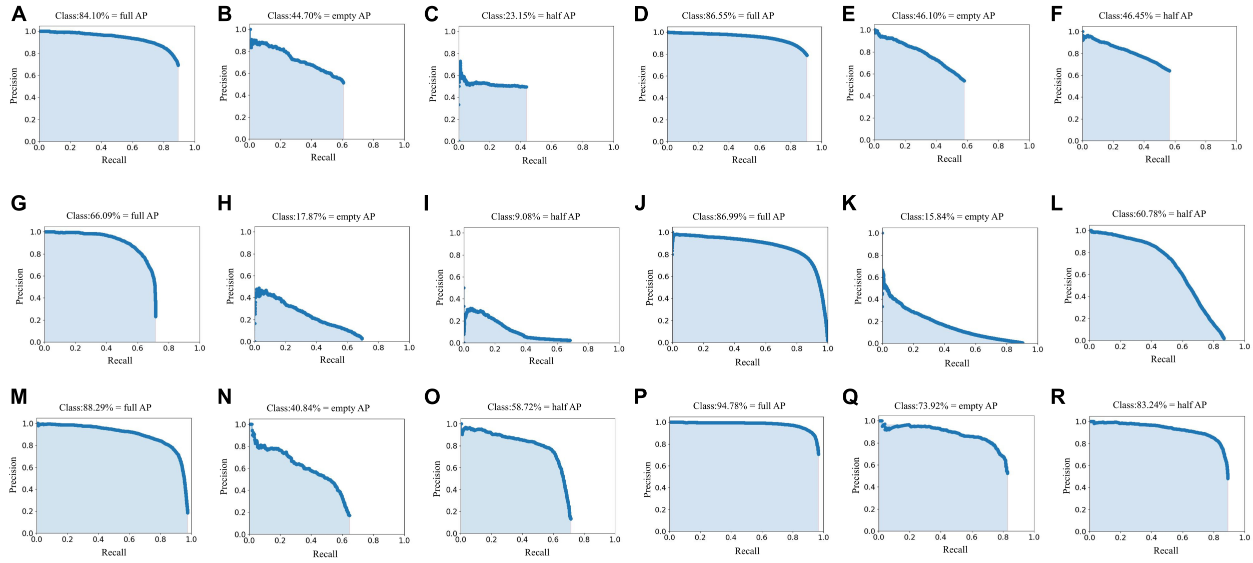


FIGURE 7 | Precision-recall curves of the different convolutional neural networks in test set. **(A–C)** Are the Faster R-CNN (ResNet50) network Precision-Recall curves, where **(A)** is the full grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network, **(B)** is the empty grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network, and **(C)** is the half grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network. **(D–F)** Are the Faster R-CNN (VGG16) network Precision-Recall curves, where **(D)** is the full grain precision-recall curve obtained by the Faster R-CNN (VGG16) network, **(E)** is the empty grain precision-recall curve obtained by the Faster R-CNN (VGG16) network, and **(F)** is the half grain precision-recall curve obtained by the Faster R-CNN (VGG16) network. **(G–I)** Are the SSD network precision-recall curves, where **(G)** is the full grain precision-recall curve obtained by the SSD network, **(H)** is the empty grain precision-recall curve obtained by the SSD network, and **(I)** is the half grain precision-recall curve obtained by the SSD network. **(J–L)** Are the EfficientDet network precision-recall curves, where **(J)** is the full grain precision-recall curve obtained by the EfficientDet network, **(K)** is the empty grain precision-recall curve obtained by the EfficientDet network, and **(L)** is the half grain precision-recall curve obtained by the EfficientDet network. **(M–O)** Are the YOLO V3 network precision-recall curves, where **(M)** is the full grain precision-recall curve obtained by the YOLO V3 network, **(N)** is the empty grain precision-recall curve obtained by the YOLO V3 network, and **(O)** is the half grain precision-recall curve obtained by the YOLO V3 network. **(P–R)** Are the YOLO V4 network precision-recall curves, where **(P)** is the full grain precision-recall curve obtained by the YOLO V4 network, **(Q)** is the empty grain precision-recall curve obtained by the YOLO V4 network, and **(R)** is the half grain precision-recall curve obtained by the YOLO V4 network.

where *TP* = the number of true positives, *FP* = the number of false positives, and *FN* = the number of false negatives. Where *N* is the total number of images in the test dataset, *M* is the number of classes, *Precision*(*k*) is the precision value at *k* images, and $\triangle Recall\left(k\right)$ is the recall change between the *k* and *k* − 1 images.

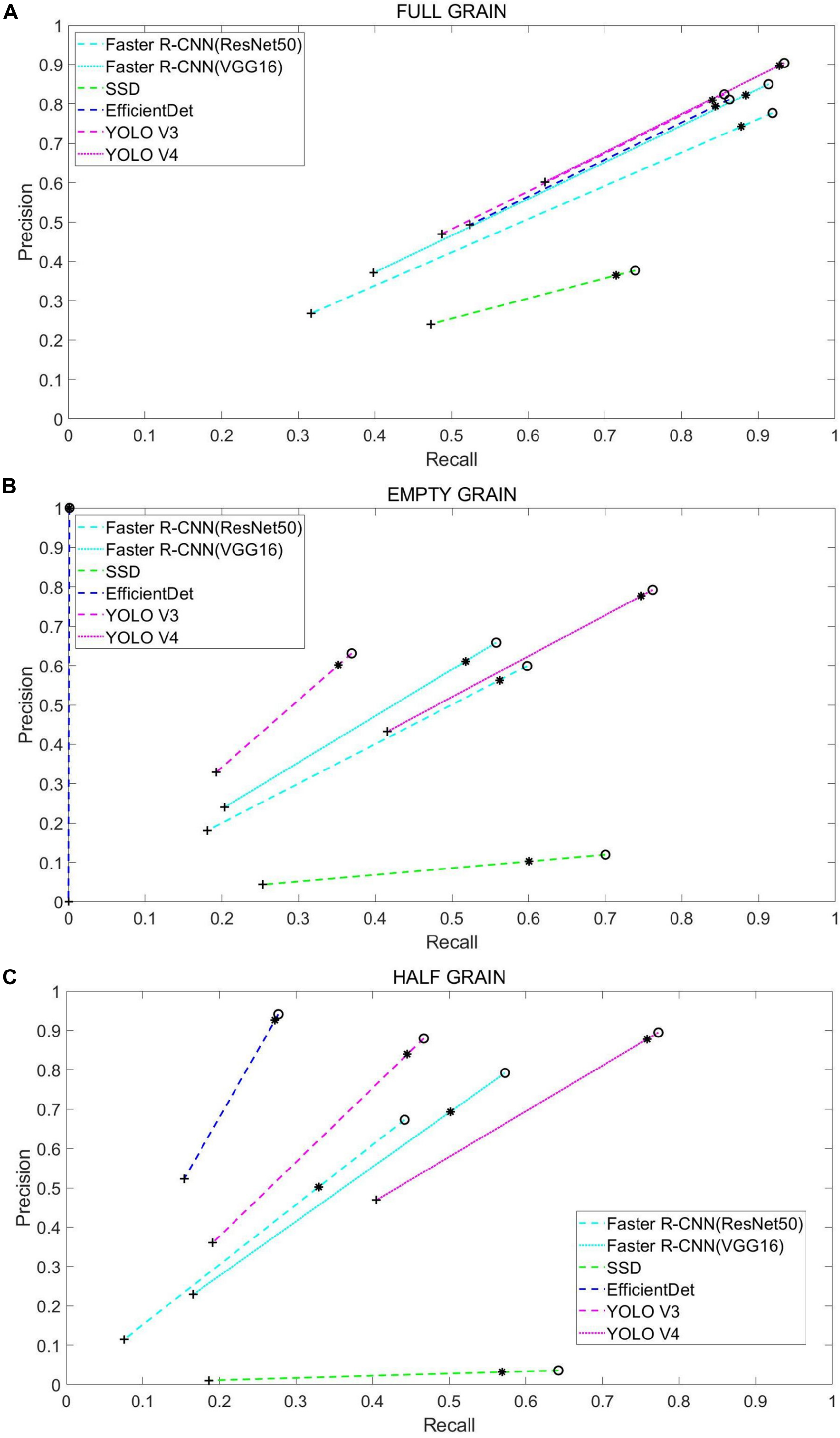


FIGURE 8 | Each color corresponds to the test results from a different network model, while the symbols "°," "∗, " and "" correspond to a 0.25, 0.5, and 0.75 overlap IOU, respectively. The results from each method and their use of these IOU thresholds are connected by dashed lines: **(A)** Test results in full grain, **(B)** test results in empty grain, and **(C)** test results in half grain.

In addition, the mean absolute error (*MAE*), the mean squared error (*MSE*), the root mean squared error (*RMSE*), and the correlation coefficient (*R*), were used as the evaluation metrics to assess the counting performance. They take the forms:

$$MAE = \frac{1}{N}\sum_{1}^{N}|t_i - c_i| \tag{10}$$

$$MSE = \frac{1}{N}\sum_{1}^{N}(t_i - c_i)^2 \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(t_i - c_i)^2} \tag{12}$$

$$R = \sqrt{1 - \frac{\sum_{i=1}^{N}(t_i - c_i)^2}{\sum_{i=1}^{N}\left(t_i - \bar{t}\right)^2}} \tag{13}$$

where N denotes the number of test images, t_i is the ground truth count for the $i - th$ image, c_i is the inferred count for the $i - th$ image, and $\bar{t}$ is the arithmetic mean of t_i.

RESULTS

Rice Grain Detection

First, we evaluated the convergence between the YOLO series model (YOLO V3, YOLO V4) and its four alternatives [Faster R-CNN (ResNet50), Faster R-CNN (VGG16), SSD, and EfficientDet], as well as the number of iterations. The loss curves of the training and verification processes from the adopted six deep neural networks are shown in **Figure 6**. For the full six networks, the uniform batch size is 4 and the learning rate starts from 0.0001. In terms of iterations, 200 are used for Faster R-CNN (ResNet50) and Faster R-CNN (VGG16), while SSD, EfficientDet, YOLO V3 and YOLO V4 use 120. It can be seen that at the beginning of the training phase, the training loss drops sharply, and then after a certain number of iterations, the loss value slowly converges around an accurate value.

Liu et al. (2021) proposes a self-attention negative feedback network (SRAFBN) for realizing the real-time image super-resolution (SR). The network model constrains the image mapping space and selects the key information of the image through the self-attention negative feedback model, so that higher quality images can be generated to meet human visual perception. There are good processing methods for the mapping from low resolution image to high resolution image, but there is still a lack of processing method from high resolution to low resolution. Therefore, we propose the following idea: We cut the 190 images into 4,560 images, re-tagged them, and added the “half” category. Among these newly cut images, 2,705 were marked as foreground images and 1,855 were not marked as background images. We input the 2,705 foreground images into the six networks that we proposed as a data set, and obtained the precision-recall curve (**Supplementary Figure 1**). This greatly improved the recognition effect of all the networks (**Supplementary Table 2**). Among them, the mAP of the proposed YOLO V4 model in the training set reached 90.13%, which is the most effective.

The features of the full grains are that they are full and the middle of the grain presents a raised state (We believe that partially filled grains caused by abiotic stress are also full grains), empty grains meanwhile, are flat and the whole grain presents a plane effect. The three-dimensional sense in an empty grain is weaker than in a full grain, and part of the empty grain is reflected by cracks and openings in its center. The fact that these differences are small results in a poor detection effect by the alternative models we proposed. The proposed YOLO V4 model uses a Mosaic data enhancing method to reduce training costs and CSPDarknet53 to reduce the number of parameters and FLOPS of the model, which not only ensures the speed and accuracy of reasoning, but also reduces the model size. At the same time, DropBlock regularization and class label smoothing are employed to avoid any overfitting due to small differences. Thus, this means that our proposed YOLO V4 model performs much better than the other alternative models.

Following this, we tested the performance of different networks on the test set (**Table 1** and **Figure 7**), where we plotted the precision and recall index graphs for full grain, empty grain, and half grain, with the *X*-axis corresponding to recall and the *Y*-axis corresponding to precision (**Figure 8**). Each

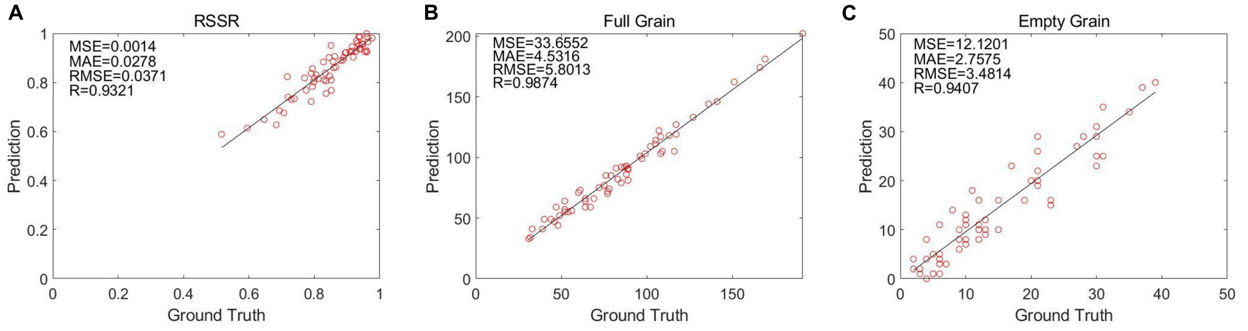


FIGURE 9 | The results calculated by the algorithm are in the form of a linear regression: **(A)** Linear regression of full grains in the optimization algorithm, **(B)** linear regression of empty grains in the optimization algorithm, and **(C)** linear regression of half grains in the optimization algorithm.

TABLE 2 | Comparison of the proposed method's results and those obtained manually.

Sample label	1	2	3	4	5	6	7	8	9	10
No. of full grains per panicle determined manually	64	88	117	83	97	141	54	64	52	89
No. of full grains per panicle determined using proposed algorithm	64	86	119	82	99	146	55	66	55	91
No. of empty grains per panicle determined manually	35	39	27	21	15	9	20	5	3	12
No. of empty grains per panicle determined using proposed algorithm	34	40	27	20	16	10	20	5	2	11
RSSR determined manually, %	64.65	69.29	81.25	79.81	86.61	94.00	72.97	92.75	94.55	88.12
RSSR determined using proposed algorithm, %	64.89	68.53	81.55	80.23	86.18	93.65	73.08	92.69	95.79	88.98
Accuracy of the full grain number per panicle, %	100	97.73	98.32	98.80	97.98	96.58	98.18	96.97	94.55	97.80
Accuracy of the empty grain number per panicle, %	97.14	97.50	100	95.24	93.75	90.00	100	100	66.67	91.67
Accuracy of the seed setting rate, %	99.63	98.90	99.63	99.48	99.50	99.63	99.85	99.94	98.71	99.03

color corresponds to the test results of a network structure. For each color, the symbols "°," "*," and "″" represent the respective overlapping IoU thresholds of 0.25, 0.50, and 0.75. Since in an ideal situation, both indicators will be close to 1, the best approach will be shown as close to the upper right corner as possible. It is clear from **Figure 8** that the results from the YOLO V4 model were significantly better than those from the other networks, regardless of their category. For all methods, we noted that both accuracy and recall measures were lower when the overlap threshold was 0.75, and highest when the overlap threshold was 0.25. This means that in the case of more stringent matching criteria (higher IoU thresholds), fewer detected rice grains were matched with instances from the ground truth, which resulted in lower indices for both. The network closest to the top right was YOLO V4, with an overlap threshold of 0.25 and 0.50, respectively.

Calculation of Rice Seed Setting Rate

Through an analysis and comparison, YOLO V4 was finally selected as the main network to be used for RSSR predictions, due to its good partitioning effect on the rice grains. For the calculation of RSSR, the rice images were first input for automatic cropping, with the number of full grain, empty grain, and half grain in each cropped image predicted by the YOLO V4 network. Following this, all sub-images belonging to an image were automatically synthesized, and the RSSR was calculated according to the algorithm we provided.

The linear regression between the manual calculation result and the optimization algorithm's calculation result of 60 rice images is shown through (**Figures 9A–C**). It can be observed that YOLO V4 is the most efficient at identifying rice grains, and that its correlation coefficient *R* surpasses 90%.

Table 2 is a comparison of the results from the proposed method and those that were obtained manually. From **Table 2**, it can be seen that the proposed method's average accuracy for calculating the full grain number per panicle was 97.69%, for the empty grain number per panicle it was 93.20%, and for the RSSR

TABLE 3 | Detection performance of the different models during the training data set's untrimmed state.

Network name	Category	Precision	Recall	F1	AP	mAP
Faster R-CNN (ResNet50)	Full grain	14.43%	3.01%	0.05	0.55%	0.30%
	Empty grain	6.61%	0.26%	0	0.05%	
Faster R-CNN (VGG16)	Full grain	12.47%	2.40%	0.04	0.37%	0.21%
	Empty grain	7.63%	0.22%	0	0.04%	
SSD	Full grain	9.37%	9.95%	0.1	1.11%	0.67%
	Empty grain	2.14%	0.14%	0	0.22%	
EfficientDet	Full grain	0.01%	0.01%	0	0.26%	0.14%
	Empty grain	0.01%	0.01%	0	0.01%	
YOLO V3	Full grain	45.53%	45.77%	0.46	29.82%	16.65%
	Empty grain	37.21%	4.39%	0.08	3.48%	
YOLO V4	Full grain	49.54%	40.30%	0.44	24.51%	17.97%
	Empty grain	43.69%	17.60%	0.25	11.43%	

TABLE 4 | Detection performance of various networks under precise division.

Network name	Category	Precision	Recall	F1	AP	mAP
Faster R-CNN (ResNet50)	Full grain	73.85%	86.68%	0.80	80.82%	37.04%
	Empty grain	59.84%	43.10%	0.50	36.48%	
	H-full grain	51.31%	31.87%	0.39	25.12%	
	H-empty grain	51.54%	4.35%	0.08	5.73%	
Faster R-CNN (VGG16)	Full grain	77.89%	90.01%	0.84	86.53%	43.91%
	Empty grain	59.51%	51.42%	0.55	43.66%	
	H-full grain	75.34%	30.13%	0.43	36.66%	
	H-empty grain	73.08%	3.70%	0.07	8.77%	
SSD	Full grain	70.67%	75.72%	0.73	71.24%	37.75%
	Empty grain	38.80%	50.25%	0.44	38.99%	
	H-full grain	16.15%	55.43%	0.25	28.89%	
	H-empty grain	34.02%	10.64%	0.16	11.87%	
EfficientDet	Full grain	80.89%	80.01%	0.80	86.01%	44.38%
	Empty grain	80.14%	1.80%	0.04	32.36%	
	H-full grain	83.19%	25.71%	0.39	58.46%	
	H-empty grain	0.00%	0.00%	0.00	0.69%	
YOLO V3	Full grain	82.93%	83.06%	0.83	87.72%	46.78%
	Empty grain	65.59%	27.47%	0.39	35.51%	
	H-full grain	80.04%	39.53%	0.53	56.16%	
	H-empty grain	80.00%	1.16%	0.02	7.74%	
YOLO V4	Full grain	86.87%	93.17%	0.9	94.27%	66.57%
	Empty grain	79.30%	76.37%	0.78	78.44%	
	H-full grain	86.73%	51.07%	0.64	64.38%	
	H-empty grain	79.93%	14.99%	0.25	29.19%	

it was 99.43%. This indicates that the proposed method offers high accuracy and stability. The deviations in a few cases can be attributed to identification errors for some small empty grains and half grains during the YOLO V4 model's testing process. The characteristics of some empty grains are not obvious, appearing highly similar to the full grains. Some half grains have a relatively complete shape, which is similar to the shape of full grains with their shielding, resulting in recognition difficulties.

DISCUSSION

Detection Effect of Different Data Sets

To better understand the performance of our proposed methods, we studied the network detection effects during different image states. First, however, it must be noted that the rice identification process is carried out using the initial image, which has 4,032 × 3,024 pixels.

Table 3 shows the detection performances of the six deep learning networks, all of which are clear as the high input images undergo the necessary resizing before going through the networks. However, in spite of the preservation of various network category characteristics, the minor differences between full and empty grains are still easily ignored. Therefore, although we adopted a variety of networks to train the data set, we were still unable to find a network with an accuracy as high as our own experimental results. Our proposed model, the YOLO V4 network, achieved the best accuracy among the six networks, with an mAP value of 17.97%, however, this is still far below our target expectations.

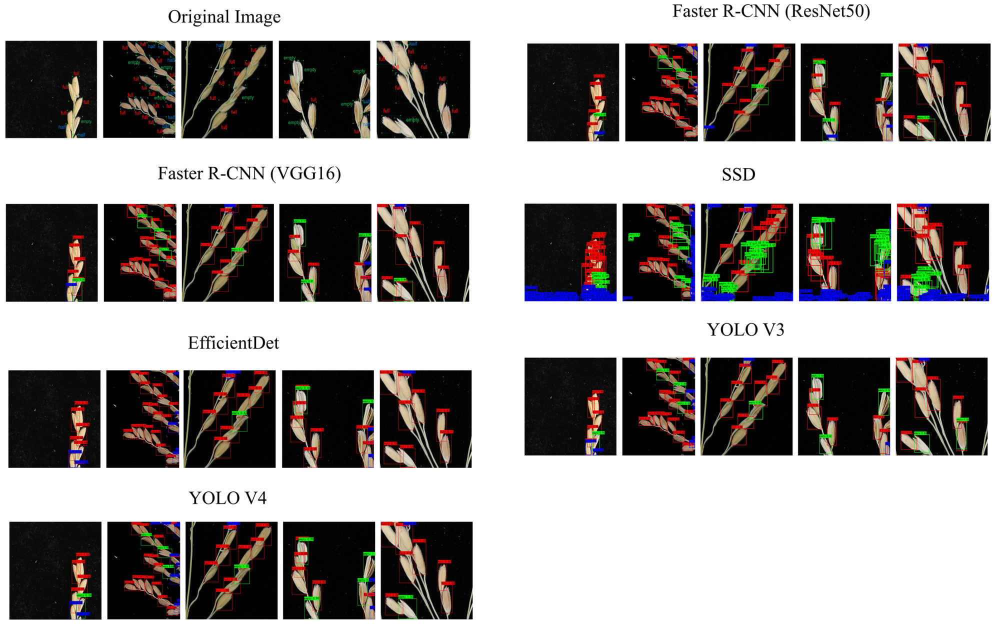


FIGURE 10 | Comparison between the prediction results and the actual results from the different networks.

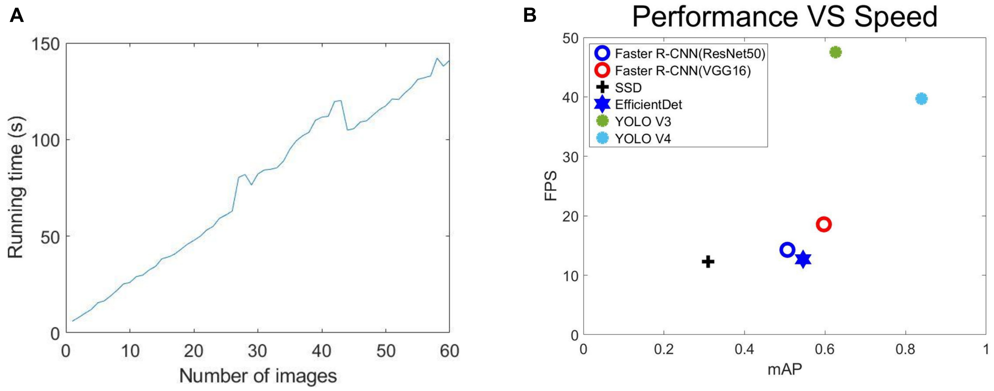


FIGURE 11 | Performance: **(A)** Relationship between the number of different prediction images and prediction time, **(B)** the error in term of mAP vs. Speed (FPS) on test set.

Table 4 shows the detection effect under precise division. 4,560 images were obtained by cropping 190 images, whereupon these were used as the data set. The cropping principle is that the size of the cropped images be as close as possible to the input size of each network, and that the categories of half-full grain and half-empty grain are added. H-full and H-empty represent the full and empty grains detected in in the half grain count after cropping. It can be observed that the accuracy of all the networks and the recognition accuracy of some of the categories have been improved. These results accorded with our hypothesis and proved the effectiveness of the proposed method. However, the overall performance remains unsatisfactory.

Prediction Effect of Different Convolution Neural Networks

Figure 10 shows the predictive effects of our six network architectures: Faster R-CNN (ResNet50), Faster R-CNN (VGG16), SSD, EfficientDet, YOLO V3, and YOLO V4. Through this, it can be seen that most of the target detection methods greatly improve the detection effect once image segmentation has been completed. Faster R-CNN (ResNet50), Faster R-CNN (VGG16), EfficientDet, and YOLO V3 in particular, showed significant improvements when working with the proposed method, and performed well when detecting full grain. Almost all the full grain samples were detected, but empty and half grain samples were not detected as efficiently. YOLO V4 on the other hand, was not only the best at detecting full grains, but also at detecting the empty and half grains, as well as many categories that the other networks were unable to detect.

Performance vs. Speed

Figure 11A shows that as the number of predicted images increased, so did the prediction time, with a roughly linear increase. We calculated that one image's average running time is about 2.65 s, which is much less than that achieved with a manual counting time.

We also considered the reasoning speed of various networks. **Figure 11B** shows the error terms for mAP and speed (FPS) on the test data set. Faster R-CNN (ResNet50), Faster R-CNN (VGG16), SSD, EfficientDet, YOLO V3, YOLO V4 were all implemented using the same Pytorch framework and used the same input image size. We measured the speed of all the methods on a single Nvidia GeForce GTX TITAN XP GPU (12G) computer. According to **Figure 11B**, YOLO V4 is superior to the other five methods except YOLO V3 in both its speed (FPS) and mAP (the higher the better). YOLO V4 is significantly better than YOLO V3 in mAP, but the detection speed (FPS) is slightly inferior. Considering the overall situation, we think that the importance of mAP is higher than the detection speed (FPS). Therefore, we think that the performance of YOLO V4 is stronger. Faster R-CNN (ResNet50), Faster R-CNN (VGG16), and EfficientDet meanwhile, show less of a difference in their performance and speed. The SSD's speed was similar to Faster R-CNN (ResNet50), Faster R-CNN (VGG16), and EfficientDet, but its performance was far below that of the other networks, with a poor detection of small features being the main issue.

Error Analysis

Through the identification of the grains of 60 rice images, we detected that the average error number of full grains was 5.78 grains, and the average error number of empty grains was 2.76 grains, and the final RSSR error was 2.84%. In addition, the results of MAE, MSE, RMSE for solid grains, shrunken grains, and seed setting rates can be obtained from **Figures 9A–C**, which shows that although our results have certain errors, they are acceptable.

In future work, we plan to continue improving the detection accuracy of full rice grains and empty grains, and to eliminate the impact of full half grains on RSSR as much as possible. Considering the high efficiency of the program, we will also improve the RSSR calculation speed.

CONCLUSION

In this paper, a RSSR calculation method based on deep learning for high-resolution images of rice panicles is proposed for the realization of the automatic calculation of RSSR. The calculation method is composed of both deep learning and RSSROA. Deep learning is used to identify the grain category characteristics of rice, and the RSSROA is used to calculate the RSSR.

In this study, a rice panicle data set composed of 4560 cut images was established. These images were taken from multiple rice varieties which had been grown under the same environment and had been processed based on image segmentation. Through the identification and comparison of data sets, we choose YOLO V4 with the best comprehensive performance as our network for calculating RSSR. In addition, the detection accuracy for full grain, empty grain, and RSSR in 10 randomly selected rice images, were 97.69, 93.20, and 99.43%, respectively. The calculation time for the RSSR in each image was 2.65 s, which meets the needs for automatic calculation. In cooperation with rice research institutions, because this method is a non-destructive operation when collecting rice panicles information, it is more convenient for rice researchers to reserve seeds, and the simple operation method enables rice researchers to obtain RSSR information more efficiently and accurately, which will be a reliable method for further estimating rice yield.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://www.kaggle.com/soberguo/riceseedsettingrate.

AUTHOR CONTRIBUTIONS

YG: formal analysis, investigation, methodology, visualization, and writing—original draft. SL: supervision and validation. YL, ZH, and ZZ: project administration and resources. DX: writing—review and editing and funding acquisition. QC: writing—review and editing, funding acquisition, and resources. JW: writing—review and editing and resources. RZ: designed the research the article, conceptualization, data curation, funding acquisition, resources, and writing—review and editing. All authors agreed to be accountable for all aspects of their work to ensure that the questions related to the accuracy or integrity of any part is appropriately investigated and resolved, and approved for the final version to be published.

FUNDING

This work was supported by the National Natural Science Foundation of China (Grant nos. 31400074, 31471516, 31271747, and 30971809), the Natural Science Foundation of Heilongjiang Province of China (LH2021C021), and the Heilongjiang Postdoctoral Science Foundation (LBH-Q18025).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.770916/full#supplementary-material

Supplementary Figure 1 | Precision-recall curves of the different convolutional neural networks in training set. **(A–C)** Are the Faster R-CNN (ResNet50) network Precision-Recall curves, where **(A)** is the full grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network, **(B)** is the empty grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network, and **(C)** is the half grain precision-recall curve obtained by the Faster R-CNN (ResNet50) network. **(D–F)** Are the Faster R-CNN (VGG16) network Precision-Recall curves, where **(D)** is the full grain precision-recall curve obtained by the Faster R-CNN (VGG16) network, **(E)** is the empty grain precision-recall curve obtained by the Faster R-CNN (VGG16) network, and **(F)** is the half grain precision-recall curve obtained by the Faster R-CNN (VGG16) network. **(G–I)** Are the SSD network precision-recall curves, where **(G)** is the full grain precision-recall curve obtained by the SSD network, **(H)** is the empty grain precision-recall curve obtained by the SSD network, and **(I)** is the half grain precision-recall curve obtained by the SSD network. **(J–L)** Are the EfficientDet network precision-recall curves, where **(J)** is the full grain precision-recall curve obtained by the EfficientDet network, **(K)** is the empty grain precision-recall curve obtained by the EfficientDet network, and **(L)** is the half grain precision-recall curve obtained by the EfficientDet network. **(M–O)** Are the YOLO V3 network precision-recall curves, where **(M)** is the full grain precision-recall curve obtained by the YOLO V3 network, **(N)** is the empty grain precision-recall curve obtained by the YOLO V3 network, and **(O)** is the half grain precision-recall curve obtained by the YOLO V3 network. **(P–R)** Are the YOLO V4 network precision-recall curves, where **(P)** is the full grain precision-recall curve obtained by the YOLO V4 network, **(Q)** is the empty grain precision-recall curve obtained by the YOLO V4 network, and **(R)** is the half grain precision-recall curve obtained by the YOLO V4 network.

REFERENCES

Afonso, M., Fonteijn, H., Fiorentin, F., Lensink, D., Mooij, M., and Faber, N. (2020). Tomato fruit detection and counting in greenhouses using deep learning. *Front. Plant Sci.* 11:571299. doi: 10.3389/fpls.2020.571299

Bochkovskiy, A., Wang, C., and Mark Liao, H. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv* [Preprint]. arXiv:2004.10934.

Chatnuntawech, I., Tantisantisom, K., Khanchaitit, P., Boonkoom, T., Bilgic, B., and Chuangsuwanich, E. (2018). Rice classification using spatio-spectral deep convolutional neural network. *arXiv* [Preprint] arXiv:1805.11491,

Chen, X., Xiang, S., Liu, C., and Pan, C. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* 11, 1797–1801. doi: 10.1109/ACPR.2013.33

Chu, Z., and Yu, J. (2020). An end-to-end model for rice yield prediction using deep learning fusion. *Comput. Electron. Agric.* 174:105471. doi: 10.1016/j.compag.2020.105471

Csurka, G., Larlus, D., and Perronnin, F. (2013). "What is a good evaluationmeasure for semantic segmentation?," in *Proceedings of the British Machine Vision Conference*, (Bristol: BMV Press).

Desai, S. V., Balasubramanian, V. N., Fukatsu, T., Ninomiya, S., and Guo, W. (2019). Automatic estimation of heading date of paddy rice using deep learning. *Plant Methods.* 15:76. doi: 10.1186/s13007-019-0457-1

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

Dong, C., Loy, C., He, K., and Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 295–307. doi: 10.1109/TPAMI.2015.2439281

Frank, J. (1951). The kolmogorov-smirnov test for goodness of fit. *Am. Stat. Assoc.* 46, 68–78. doi: 10.1080/01621459.1951.10500769

Ghadirnezhad, R., and Fallah, A. (2014). Temperature effect on yield and yield components of different rice cultivars in flowering stage. *Int. J. Agron.* 2014:846707. doi: 10.1155/2014/846707

Gong, L., Lin, K., Wang, T., Liu, C., Yuan, Z., Zhang, D., et al. (2018). Image-based on- panicle Rice [*Oryza sativa L.*] grain counting with a prior edge wavelet correction model. *Agronomy* 8:91. doi: 10.3390/agronomy8060091

He, H., and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21, 1263–1284. doi: 10.1109/TKDE.2008.239

Hong Son, N., and Thai-Nghe, N. (2019). "Deep learning for rice quality classification," in *Proceedings of the International Conference on Advanced Computing and Applications (ACOMP)*, (Nha Trang: Institute of Electrical and Electronics Engineers), 92–96.

Kong, H., and Chen, P. (2021). Mask R-CNN-based feature extraction and three-dimensional recognition of rice panicle CT images. *Plant Direct.* 5:e00323. doi: 10.1002/pld3.323

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., and Verma, S. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21:5386. doi: 10.3390/s21165386

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

Li, S., Li, W., Huang, B., Cao, X., Zhou, X., Ye, S., et al. (2013). Natural variation in PTB1 regulates rice seed setting rate by controlling pollen tube growth. *Nat. Commun.* 4:2793. doi: 10.1038/ncomms3793

Lin, P., Li, X., Chen, Y., and He, Y. (2018). A deep convolutional neural network architecture for boosting image discrimination accuracy of rice species. *Food Bioprocess Technol.* 11, 765–773. doi: 10.1007/s11947-017-2050-9

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., et al. (2016). SSD: single shot multibox detector. *arXiv* [Preprint]. arXiv:1512.02325v5.

Liu, X., Chen, S., Song, L., Woźniak, M., and Liu, S. (2021). Self-attention negative feedback network for real-time image super-resolution. *J. King Saud Univ. Comput. Inf. Sci.* doi: 10.1016/j.jksuci.2021.07.014

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

Mitra, V., Sivaraman, G., Nam, H., Espy-Wilson, C., Saltzman, E., and Tiede, M. (2017). Hybrid convolutional neural networks for articulatory and acoustic information based speech recognition. *Speech Commun.* 89, 103–112. doi: 10.1016/j.specom.2017.03.003

Oosterom, E. J. V., and Hammer, G. L. (2008). Determination of grain num-ber in sorghum. *Field Crops Res.* 108, 259–268. doi: 10.1016/j.fcr.2008.06.001

Rajeshwari, P., Abhishek, P., Srikanth, P., and Vinod, T. (2019). Object detection: an overview. *Int. J. Trend Sci. Res. Dev* 3, 1663–1665.

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

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *arXiv* [Preprint]. arXiv:1506.01497v3.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Netw.* 2015, 85–117. doi: 10.1016/j.neunet.2014.09.003

Tan, M., Pang, R., and Le, V. Q. (2020). EfficientDet: scalable and efficient object detection. *arXiv* [Preprint] arXiv:1911.09070v7,

Wu, W., Liu, T., Zhou, P., Yang, T., Li, C., Zhong, X., et al. (2019). Image analysis-based recognition and quantification of grain number per panicle in rice. *Plant Methods* 15:122. doi: 10.1186/s13007-019-0510-0

Xiang, X., Zhang, P., Yu, P., Zhang, Z., Sun, L., Wu, W., et al. (2019). LSSR1 facilitates seed setting rate by promoting fertilization in rice. *Rice* 12:31. doi: 10.1186/s12284-019-0280-3

Xiong, X., Duan, L., Liu, L., Tu, H., Yang, P., Wu, D., et al. (2017). Panicle-SEG: a robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization. *Plant Methods* 13:104. doi: 10.1186/s13007-017-0254-7

Xu, C., Jiang, H., Yuen, P., Zaki Ahmad, K., and Chen, Y. (2020). MHW-PD: a robust rice panicles counting algorithm based on deep learning and multi-scale hybrid window. *Comput. Electron. Agric.* 173:105375. doi: 10.1016/j.compag.2020.105375

Xu, Y., Yang, J., Wang, Y., Wang, J., Yu, Y., Long, Y., et al. (2017). OsCNGC13 promotes seed-setting rate by facilitating pollen tube growth in stylar tissues. *PLoS Genet.* 13:e1006906. doi: 10.1371/journal.pgen.1006906

Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S., et al. (2015). Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *Neuroimage* 108, 214–224. doi: 10.1016/j.neuroimage.2014.12.061

Zhao, H., Sun, L., Jia, Y., Yu, C., Fu, J., Zhao, J., et al. (2020). Effect of nitrogen,phosphorus and potassium fertilizer combined application on japonica rice growth and yield in cold areas. *J. Northeast Agric. Univ.* 51, 1–13. doi: 10.19720/j.cnki.issn.1005-9369.2020.12.001

Zhou, C., Ye, H., Hu, J., Shi, X., Hua, S., Yue, J., et al. (2019). Automated counting of rice panicle by applying deep learning model to images from unmanned aerial vehicle platform. *Sensors* 19:3106. doi: 10.3390/s19143106

Zou, H., Lu, H., Li, Y., Liu, L., and Cao, Z. (2020). Maize tassels detection: a benchmark of the state of the art. *Plant Methods* 16:108. doi: 10.1186/s13007-020-00651-z

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: 04 January 2022
doi: 10.3389/fpls.2021.774068

Outdoor Plant Segmentation With Deep Learning for High-Throughput Field Phenotyping on a Diverse Wheat Dataset

Radek Zenkl[1]*, *Radu Timofte*[2], *Norbert Kirchgessner*[1], *Lukas Roth*[1], *Andreas Hund*[1], *Luc Van Gool*[2], *Achim Walter*[1] and *Helge Aasen*[3]

[1] Group of Crop Science, Department of Environmental Systems Science, Institute of Agricultural Sciences, ETH Zurich, Zurich, Switzerland, [2] Computer Vision Lab, Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland, [3] Remote Sensing Team, Division of Agroecology and Environment, Agroscope, Zurich, Switzerland

Robust and automated segmentation of leaves and other backgrounds is a core prerequisite of most approaches in high-throughput field phenotyping. So far, the possibilities of deep learning approaches for this purpose have not been explored adequately, partly due to a lack of publicly available, appropriate datasets. This study presents a workflow based on DeepLab v3+ and on a diverse annotated dataset of 190 RGB (350 x 350 pixels) images. Images of winter wheat plants of 76 different genotypes and developmental stages have been acquired throughout multiple years at high resolution in outdoor conditions using nadir view, encompassing a wide range of imaging conditions. Inconsistencies of human annotators in complex images have been quantified, and metadata information of camera settings has been included. The proposed approach achieves an intersection over union (IoU) of 0.77 and 0.90 for plants and soil, respectively. This outperforms the benchmarked machine learning methods which use Support Vector Classifier and/or Random Forrest. The results show that a small but carefully chosen and annotated set of images can provide a good basis for a powerful segmentation pipeline. Compared to earlier methods based on machine learning, the proposed method achieves better performance on the selected dataset in spite of using a deep learning approach with limited data. Increasing the amount of publicly available data with high human agreement on annotations and further development of deep neural network architectures will provide high potential for robust field-based plant segmentation in the near future. This, in turn, will be a cornerstone of data-driven improvement in crop breeding and agricultural practices of global benefit.

Keywords: deep learning, breeding, machine learning, remote sensing, random forrest, support vector classification, high resolution image analysis, benchmark

OPEN ACCESS

Edited by:
Gregorio Egea,
University of Seville, Spain

Reviewed by:
Orly Enrique,
University of Seville, Spain
Nguyen Quoc Khanh Le,
Taipei Medical University, Taiwan

****Correspondence:***
Radek Zenkl
radek.zenkl@usys.ethz.ch

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *10 September 2021*
Accepted: *05 November 2021*
Published: *04 January 2022*

Citation:
Zenkl R, Timofte R, Kirchgessner N, Roth L, Hund A, Van Gool L, Walter A and Aasen H (2022) Outdoor Plant Segmentation With Deep Learning for High-Throughput Field Phenotyping on a Diverse Wheat Dataset. Front. Plant Sci. 12:774068. doi: 10.3389/fpls.2021.774068

1. INTRODUCTION

The growth of the human population, global climate change, and detrimental effects of agriculture on the environment exerts an increasing pressure to address challenges in crop production and breeding (Pretty et al., 2010; Reynolds and Langridge, 2016). Wheat is one of the most important staple crops and, therefore, methods assessing its performance in various management conditions and methods improving breeding pathways are urgently required. Phenotyping and thereby the quantification of plant properties from images is a core bottleneck to achieving this (Fiorani and Schurr, 2013; Walter et al., 2015).

Reliable and automated segmentation of wheat canopy under field conditions is a premise to quantify canopy cover and to derive traits, such as crop emergence, leaf growth, tillering, and other traits subsequently (Roth et al., 2018, 2020). Also, the classification between crops and weeds and the distinction between healthy and diseased plant tissue are based on this essential first step: how to detect the crop organ of interest in any given image?

A reliable organ detection is a challenging task due to diverse and dynamic lighting conditions, changing optical properties of the soil due to wetting and drying, and diverse spatial patterns which result in highly complex and constantly changing scenes (refer to **Figure 1** for a collection of random samples from the same field).

With current methods, a significant amount of manual work is still required to run and evaluate the experiments. Thus, research groups are limited in the number and size of experiments that they can operate. This has led to the emerging trend of high-throughput phenotyping, increasing analysis throughput by focusing on scalable experiments with a high level of automation which should improve the genetic gain of breeding programs (Araus and Cairns, 2014). In recent years, different platforms for automatic data acquisition have been developed in hope of attaining all the relevant information for the crop assessment (Hund et al., 2019). This includes Unmanned Aerial Vehicle (UAV) (Candiago et al., 2015; Aasen et al., 2018; Burkart et al., 2018), moving platforms (Andrade-Sanchez et al., 2013; Bai et al., 2016), autonomous rovers (Ruckelshausen et al., 2009, Agerris[1]), and large-scale, fixed platforms (Kirchgessner et al., 2017; Virlet et al., 2017).

The evaluation of the acquired data is a delicate task due to the variance in the scene as described above. There is an enormous amount of possible scenarios, in which classical approaches such as manual and/or automatic visual indices thresholding often achieve their limits since they need to be tuned for every individual scenario. This makes their deployment for outdoor canopy segmentation tedious and offers limited generalization capabilities. Thus, the data evaluation of these experiments has experienced penetration of data-driven approaches through machine learning and deep learning techniques. The use of data driven approaches for phenotyping is very promising as it enables higher analysis throughput and removes potential human error, theoretically leading to better results as the evaluation is data-driven and not hand-engineered (Kamilaris and Prenafeta-Boldú, 2018).

1.1. Related Work

The challenge to be addressed can be abstracted to a semantic segmentation task. Most prominent approaches for semantic segmentation, such as Encoder-Decoder Networks, Pyramid Networks, R-CNN based models and Dilated CNN models (Ronneberger et al., 2015; Yu and Koltun, 2015; He et al., 2017; Chen et al., 2018; Wang et al., 2020) incorporate the concept of fully convolutional networks. These networks do not contain any dense, fully connected layers but leverage the notion of stacking convolutional layers with up- and downsampling. This concept preserves the spatial information throughout the network as the data is being propagated. Besides the improvement in performance, one practical benefit is that the networks can operate on varying image sizes. Typically, the used encoders are slightly adjusted standalone deep convolutional neural networks (CNNs) that have been pretrained on classification tasks in order to leverage large scale datasets for additional generalization performance. Prominent examples of such networks are He et al. (2016), Huang et al. (2017), and Xie et al. (2017).

Segmentation for agricultural applications is receiving more attention over the years as it repeatedly appears as a challenge in major computer vision conferences such as Chiu et al. (2020) or CVPPA21[2]. In our experience, under uncontrolled outdoor conditions, outdoor plant segmentation is currently an unsolved problem that appears to be a bottleneck for increasing the degree of automation in agriculture. Research has been conducted in the scope of enabling robots to distinguish different plants in order to apply precise local treatments (Milioto et al., 2018) or to detect diseases for further analysis and adjusted mitigation strategies (Singh and Misra, 2017). In addition, segmentation has also found its use in phenotyping, as it is used for leaf counting (Aich and Stavness, 2017), ears counting David et al. (2020), and plant-soil segmentation on multiple scales which are ultimately leveraged for growth tracking. Furthermore, segmentation can be leveraged for roots analysis (Smith et al., 2020) and post harvest quality control (Wu et al., 2020). Sensor carriers range from satellites imagery that allows segmenting on a field-scale (Ulmas and Liiv, 2020) to drones (Torres-Sánchez et al., 2015; Fuentes-Pacheco et al., 2019), ground vehicles (Liu et al., 2017) and stationary facilities (Sadeghi-Tehran et al., 2017) that allow segmenting individual plants. The paradigm of segmentation in agriculture is moving from empirical threshold based models (Carlson and Ripley, 1997; Zheng et al., 2009; Bai et al., 2014) and decision-tree based approaches (Guo et al., 2013) toward machine learning (Sadeghi-Tehran et al., 2017; Yu et al., 2017; Rico-Fernández et al., 2019) and deep learning (Milioto et al., 2018; Abdalla et al., 2019). The most significant change in general is that deep learning approaches are implicitly utilizing spatial context information in addition to color information.

The trend of moving toward data driven models requires an increasing amount of labeled data. Unfortunately, the number and size of publicly available agricultural datasets are very limited. These datasets are often designed for niche applications, such as detection of specific diseases. This complicates the creation of standard benchmarks and hinders the collaboration of different research groups which results in small dataset sizes. The most similar plant segmentation datasets to the Eschikon wheat segmentation (EWS) dataset are the Leaf Segmentation Challenge[3] and Sugar Beets 2016 dataset[4].

[1] https://agerris.com/

[2] https://cvppa2021.github.io/

[3] https://www.plant-phenotyping.org/CVPPP2017-challenge

[4] https://www.ipb.uni-bonn.de/data/sugarbeets2016/

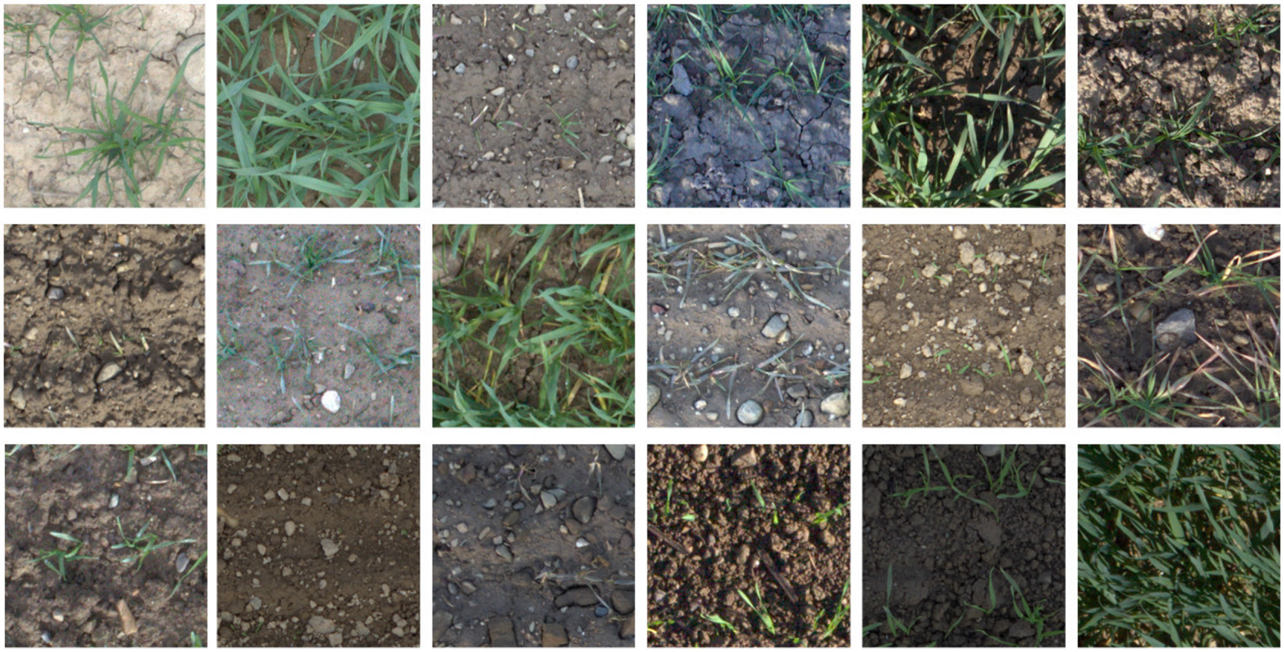


FIGURE 1 | Overview of variance in images from the Eschikon wheat segmentation (EWS) dataset of images taken between 2017 and 2020 with a Canon 5D Mark II full-frame RGB camera integrated into the sensor head of the field phenotyping platform of ETH Zurich (Kirchgessner et al., 2017).

However, both datasets have controlled diffuse lighting, and the Leaf Segmentation Challenge data originates from an indoor experiment. It is worth noting the Global Wheat Head Detection dataset (David et al., 2020) which is taken under the same conditions but offers only bounding boxes for wheat ears and not pixel-wise labels for plants.

1.2. Focus of This Work

This work focuses on establishing an analysis pipeline for plant and soil segmentation in RGB images. Images and metadata were taken in the Field Phenotyping Platform (FIP) [5] at the Research Station for Plant Sciences in Eschikon, Switzerland (Kirchgessner et al., 2017), and used to create a manually labeled segmentation dataset. Images were captured with a nadir-oriented DSLR camera that photographs different winter wheat genotypes. The annotation process was distributed and coordinated amongst two annotators which resulted in a feasible, subsampled, and stratified dataset. The experience gained by creating this novel annotated dataset will be used in future dataset extensions.

Methods to mitigate the limited dataset size were tested and their influence on the performance was quantified. Possibilities of using some of the provided metadata of the dataset were explored. The results of the algorithm were compared with respect to the quality of the annotations. The annotations' quality was assessed in form of agreement evaluation of multiple annotations attempts of same and different annotators.

2. MATERIALS AND METHODS

2.1. EWS Dataset

Within the scope of this work, a new dataset for the segmentation of plants and soil was created. It consists of 190 manually chosen and hand annotated image patches of 350×350 pixels. The images were selected from a large unlabeled dataset that consists of approximately 100,000 20 Mpx RGB images of different winter wheat genotypes. These images were collected between 2017 and 2020 with a Canon 5D Mark II (Canon Inc., Japan) - 35 mm set to autofocus and mounted on the FIP in Eschikon (47°27'01.9"N 8°40'57.5"E). Distance to the ground was approximately 3 m, resulting in a ground sampling distance of 0.3 $\frac{\text{mm}}{\text{pixel}}$. ISO, aperture and shutter speed were adapted to illumination conditions based on aperture priority in 2017 and 2018 and shutter speed priority in 2019 and 2020. The image set within each year covers the whole growing period from emergence to harvest. As the images are taken in the field, they show situations with widely varying illumination and soil moisture conditions (refer to **Figure 1**).

To generate a training set, images of the wheat canopies between emergence and stem elongation were selected. In order to ensure a balanced sampling of the different imaging situations, the following subsampling strategy was used: the first major criterion for the selection of images was the growth stage. On the one hand, only images starring recognizable seedlings were selected. On the other hand, only the images until stem elongation were considered. These growth stage restrictions were chosen as they correspond to the critical phase of early canopy development of winter wheat where yield components are formed (Simmons, 1987). Different growth stages with

[5] https://kp.ethz.ch/infrastructure/FIP.html

TABLE 1 | Eschikon wheat segmentation (EWS) dataset overview of the distribution of direct and diffuse light with respect to the number of different days.

Year	Images direct light	Images diffuse light	Different dates	Images total
2017	32	16	12	48 (25%)
2018	25	27	13	52 (27%)
2019	35	29	16	64 (34%)
2020	11	15	7	26 (14%)

respect to plant pixel ratios with respect to soil can be seen in **Figure 1**.

After this preselection, the images were grouped according to the illumination conditions direct and diffuse light folds. However, this was done on the image date level which is a simplification of the lighting dynamics. Since the complete data acquisition cycle can take multiple hours, the lighting can change within one measurement campaign. The goal was to produce a balanced set of lighting conditions and growth stages. However, the direct light scenario is over-represented in the data, which means that not enough samples for perfectly stratified lighting and growth stage subset can be established. This lead to approximately 55% of images being in the direct light category. The wheat genotypes were selected as follows, one-half of the genotypes was sampled at random, whilst the other half consists of one planophile and one erectophile genotype. **Table 1** shows the resulting general partitioning of the EWS dataset.

The resulting subset of 190 RGB images was cropped into patches of 350×350 pixels and then manually annotated in form of binary masks for plants and soil, respectively. The crop size of 350 × 350 pixels was determined so that atleast two wheat rows are visible in the image. In this way, no matter the image rotation or cropping at least one wheat row will be clearly visible after augmenting the image. The labeling process took place in GIMP[6], executed by two annotators. The protocol was to segment vegetative active material. Pixels, where the annotator was certain that they belong to vegetative active material from a wheat plant, should be labeled as such. Everything else (soil, rocks, dead plants, etc.) belongs to the class vegetative inactive material. The segmented masks were then exported as lossless 8-bit monochromatic PNG images. The resulting 190 images required approximately 80 h of combined annotation work.

Besides the images, multiple additional metadata is provided. This contains the timestamps of the images, camera settings (ISO, F-number, exposure), and measurements from a weather station that logs temperature, soil moisture, and light flux. Based on the temperature measurements, GDD metric is calculated and provided as well (see Growing Degree Days in **Appendix 2.2**). The distribution of data acquisition dates is bi-modal with a main focus on spring and a secondary focus in late fall. This distribution corresponds to the winter wheat growth cycle. Winter wheat is sown in fall where weather conditions allow for phenotyping and plant growth is significant. During winter, insignificant changes in plant canopies occur, and measurement conditions are unfavorable particularly due to very short, dim days or snow cover. In spring, growth is restarted, and measurement conditions improve and allow for phenotyping again. The images were taken during different times of the day. The acquisition times cover a great portion of a day, except for late and early hours. The challenges with lighting conditions can be seen in the different camera settings that should compensate for the changes in the scene. The camera's sensor gain (ISO) was kept low when possible for achieving a maximal signal to noise ratio. The movement of the camera platform and plants due to the wind had to be taken into consideration when selecting exposure time while the F-number had to be tuned based on the growth stage of the plants, so that the depth of field is sufficient. The histograms of date, time, ISO and combinations of exposure time with respect to F-number can be seen in **Figure 2**.

2.2. Plant Segmentation With Deep Learning

The basis of this work relies on CNN. The core principle of CNNs is to piece-wise multiply of the convolutional kernel with input. This simple operation is repeated and stacked into layers, which form a network. With this operation, the spatial information is incorporated into the computational algorithm as a combination of multiple neighboring values from the input. The research in this area has contributed to many different variations of the convolution itself and also of the ways how to combine the operations (for example, see He et al., 2016; Chollet, 2017; Chen et al., 2018). As the input is passed through multiple layers of the network, complex combinations of input are created. Based on the application, the architectures have varying forms. Fully convolutional image segmentation consists of two major steps. First, a smaller set of high-level features is extracted from the image input. Afterward, the extracted features are used to make predictions with the original resolution for every individual pixel. One of the approaches for this problem is to use an encoder-decoder architecture. By its design, the encoder is forced to compress the data into some high-level representation while still preserving a link to the position in the original images which is usually realized in a form of low-level feature and/or spatial information propagation. In contrast, the decoder is forced to restore the original resolution of the image from high-level features.

For the encoder module, ResNet (He et al., 2016) has been selected. It is a widely used deep learning architecture that has been proved in a broad range of different scenarios (Jung et al., 2017; Lin et al., 2018; Reddy and Juliet, 2019; Wang et al., 2019). The key elements are the residual blocks where the output consists of a sum of input passed through convolutional layers and the original input. This approach helps with the problem of vanishing gradient for deep networks as it yields a more direct way of propagating information deeper through the network. Based on the number and sizes of the underlying convolutions multiple ResNet variants with different degrees of complexity have been introduced. The choice of ResNet depth directly influences the expressivity of the network and thus its

[6] https://www.gimp.org/

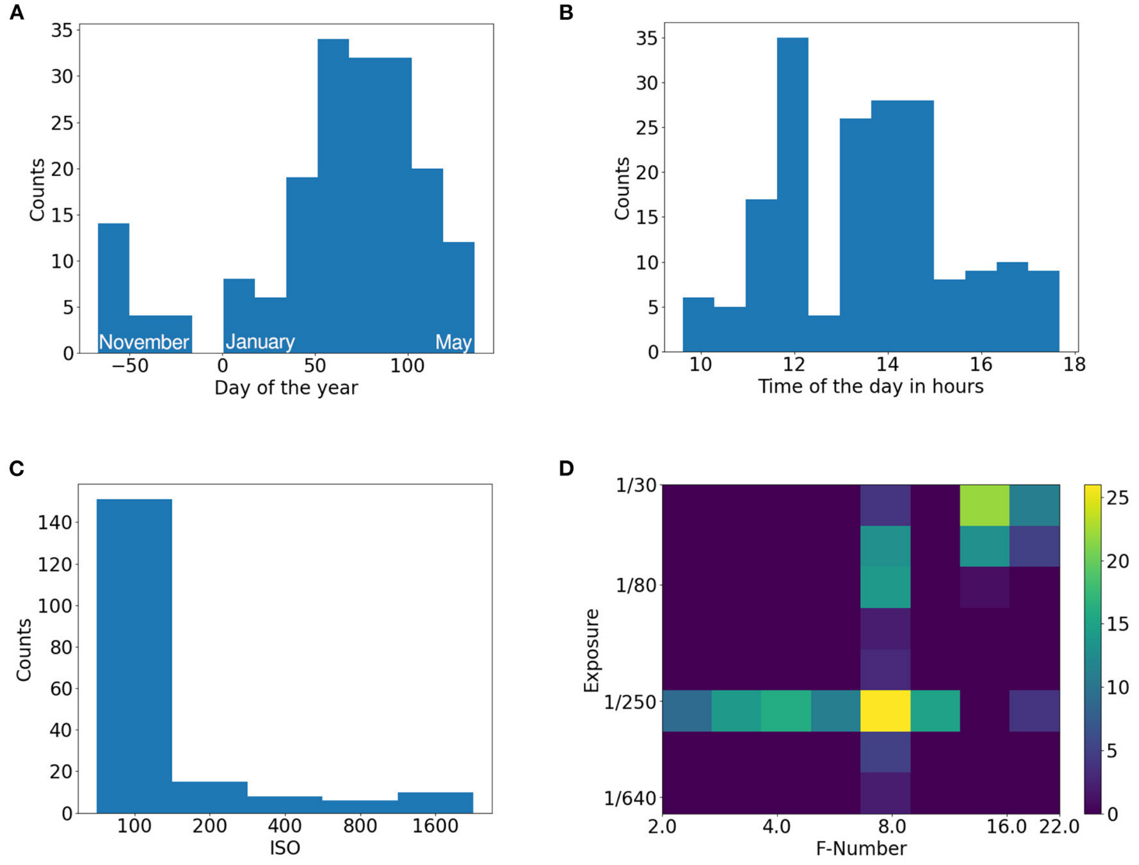


FIGURE 2 | Histograms of EWS dataset. **(A)** Day of the year, **(B)** Time of the day, **(C)** ISO settings, **(D)** Exposure and F-number settings pair.

performance (He et al., 2016). Deeper networks are able to learn more complex relations at the cost of increasing the total number of parameters. Usually, this leads to a trade-off between performance and speed. However, for small datasets, deeper networks tend to overfit the data due to their larger amount of parameters.

Deeplab v3+ (Chen et al., 2018) was selected as a segmentation framework. It is a variation of the encoder-decoder architecture. It uses Atrous Spatial Pyramid Pooling (ASPP) to extract features at multiple scales at the same time. Additionally, it leverages depthwise separable convolution which decomposes a depth-wise convolution from a 3D convolution applied on the spatial dimension and on the channels at the same time into a 2D spatial convolution followed by a channel-wise 1×1 point convolution. This approach greatly reduces the number of parameters required. The atrous convolution (also referred to as dilated convolution) introduces a spacing for the convolution kernel so that it is not necessarily applied to neighboring values only, but with the same amount of parameters, it can be spread out to a larger field. This offers a direct way to control the resolution and receptive field of the features in the network. This concept is leveraged in the ASPP module where features are extracted at multiple scales by using multiple different rates for the atrous convolution at the same time. The extracted low-level and high-level features are then combined in the decoder module where the original resolution and pixel-wise predictions as achieved in multiple steps including bilinear upscaling two times. Deeplab v3+ is a well-proven and extensively used architecture for semantic segmentation. It has demonstrated state-of-the-art performance on multiple datasets with diverse applications. This has led to the high availability of the model with pretrained weights. Since the target domain of this work offers a limited dataset size only, the availability of pretrained models needs to be considered during the selection.

2.3. Implementation Details

The proposed method was implemented in Pytorch Framework[7] and trained on Nvidia RTX3070 with 8GB GPU memory, 16GB RAM, 4 cores of AMD Threadripper 3960X. It is based on DeepLab v3+ architecture with ResNet50 (He et al., 2016) backbone pretrained on Imagenet (Krizhevsky et al., 2012)[8]. The network was trained on data from the years 2018–2020 using the crossentropy loss while reporting on 2017. This results in 154 images (75.0%) used for training and 24 images (12.5%) for validation and 24 images (12.5%) for testing. Images for

[7] https://pytorch.org/

[8] https://download.pytorch.org/models/resnet50-19c8e357.pth

validation and testing were split at random. SGD optimizer with learning rate 0.1, the momentum of 0.9, and batch size of 16 was used to train with mixed precision for 150 epochs. The architecture incorporates feature injection of additional inputs and freezing of network parts (shown in **Figure 3**).

Random flipping, rotation by 20 degrees, and cropping were used during image loading to generate images with the size of 224 × 224 px. Additionally, jittering of saturation by 25%, contrast by 10%, and brightness by 1% were applied. Next, the images are normalized to $[0, 1]$ and standardized with the mean of $[0.485, 0.456, 0.406]$ and standard deviation of $[0.229, 0.224, 0.225]$ which were used during the pretraining on the Imagenet. Finally, the images were upscaled to 448 × 448 px using the bilinear transform and gaussian noise with an SD of 0.001 was applied. The F-Number and exposure were represented as a decimal number while ISO was first transformed with $log_2(\mathrm{ISO}/100)$.

2.4. Experiment Overview and Evaluation Methodology

The very basis of metrics used in this case is to interpret the vegetative active plant pixels as positives and the remaining pixels (soil, vegetative inactive material, etc.) as negatives. Based on this a confusion matrix and derived scores F1 score and Intersection over Union (IoU) were calculated. As the plant pixel ratio varies from image to image and the metrics are nonlinear, the calculations were done with respect to individual images and then averaged over the dataset. The dataset was split to training, validation and testing fold, where 1 year is intentionally left out for validation and testing in order to mitigate the potential bias. The validation and testing splits have equal size and were sampled at random.

In order to explore possible improvements of plant segmentation, different extensions and variations to the classical deep learning approach were analyzed. These cover the data augmentation pipeline, transfer learning with finetuning for additional generalization, changes to the architecture, and weighting of samples. These methods try to mitigate the challenges of varying lighting conditions and external influences which are typical to applications for outdoor plants.

2.4.1. EWS Dataset Benchmark

In order to quantify the difficulty of the dataset, the following paragraph describes multiple methods used to acquire a performance benchmark. The first reported method is the unsupervised pre segmentation (refer to **Appendix 2.3**) performance.

Next, a selection of different methods used for segmentation in the scope of phenotyping is reported. This starts with Yu et al. (2017) who used a decision tree with preliminary weather state classification with Support Vector Classifier (SVC) (Platt, 1999) followed by another SVC for pixel classifications. This method is trained on 5% of all available pixels selected at random, as it did not converge when trained on more data. This is followed by Sadeghi-Tehran et al. (2017) which used Random Forest Classifier (Breiman, 2001) with 21 different color space features as input. Furthermore, Rico-Fernández et al. (2019) involved spatial context in a form of a 5 × 5 window around the individual pixels transformed into CIE-Luv color space which is fed into an SVC. This method was trained on 200 pixels per image as proposed in the publication. However, in this case, these 200 pixels were selected as random and not around plant centers. Please note that none of the methods described above included code for reproduction. Therefore, the methods had to be reverse engineered and the reported results need to be taken with caution.

Next, an out-of-the-box DeepLab v3+ with ResNet50 encoder trained from scratch using the Stochastic Gradient Descent (SGD) with a tuned learning rate of 0.1, the momentum of 0.9, batch size of 16, and crossentropy loss. This corresponds to a straightforward strategy with basic hyperparameters optimization which is then followed by its Imagenet-pretrained twin. Finally, the proposed method consists of DeepLab v3+ with ResNet50 Encoder. The encoder is pretrained on Imagenet and contains additional pathways for injection of ISO, F-number and exposure time as supplementary inputs. Another important element of the method is a tuned data augmentation pipeline (refer to section 3.7). In addition, middle blocks of the ResNet encoder were frozen during training. For implementation details, refer to section 2.3 and particularly **Figure 3**.

2.4.2. Human Annotations in Perspective

In order to properly evaluate an algorithm on the proposed EWS dataset, the subjectivity and consistency of human annotations need to be taken into account. Since this dataset is dealing with a large amount of different visual scenarios (see **Figure 1**), the performance of human annotators and tested algorithms varies with the different cases.

An overlay of the 4 annotation attempts can be seen in **Figure 4**. The first image represents the easy case with diffuse light and medium sized plants. The second image shows a similar scene as in the first image but under direct light. The next image shows a low contrast scenario of small plants.

In order to quantify the consistency of the annotators, four images were selected and annotated by two different annotators two times. This resulted in four annotation, attempts for the selected images. Based on these annotations different sets of metrics can be computed by taking one set as ground truth and the remaining three as performance benchmarks. This process can be repeated for every annotation which results in 12 benchmark permutations.

In addition, the algorithm benchmarks can be computed with respect to every annotation attempt. This leads to four benchmarks per image for each tested method. Since the benchmark metrics are non-linear, the benchmark results' variations based on the selected annotation set are not trivial. By comparing the performance of annotators and the segmentation method, the theoretical buffer for improvement can be quantified. Without having a perfect ground truth, the theoretical performance is bounded by the quality of the labels.

Annotators' agreement with respect to one another and to the proposed method's performance is reported in section 3.2.

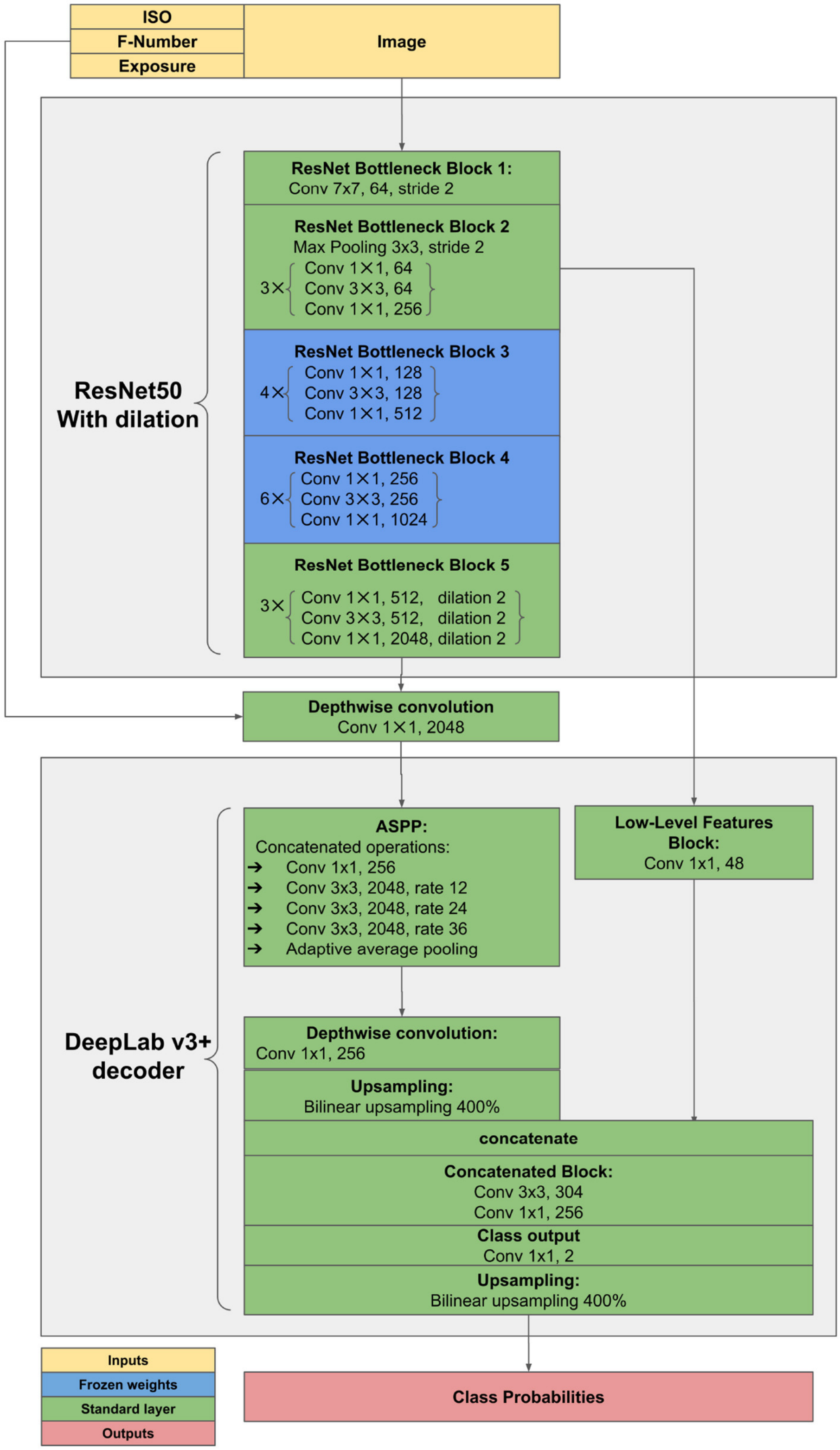


FIGURE 3 | Adjustments to ResNet encoder. Diagram denotes feature injection pathways and frozen layers during training. Blocks correspond to the blocks of convolutions in the original ResNet architecture.

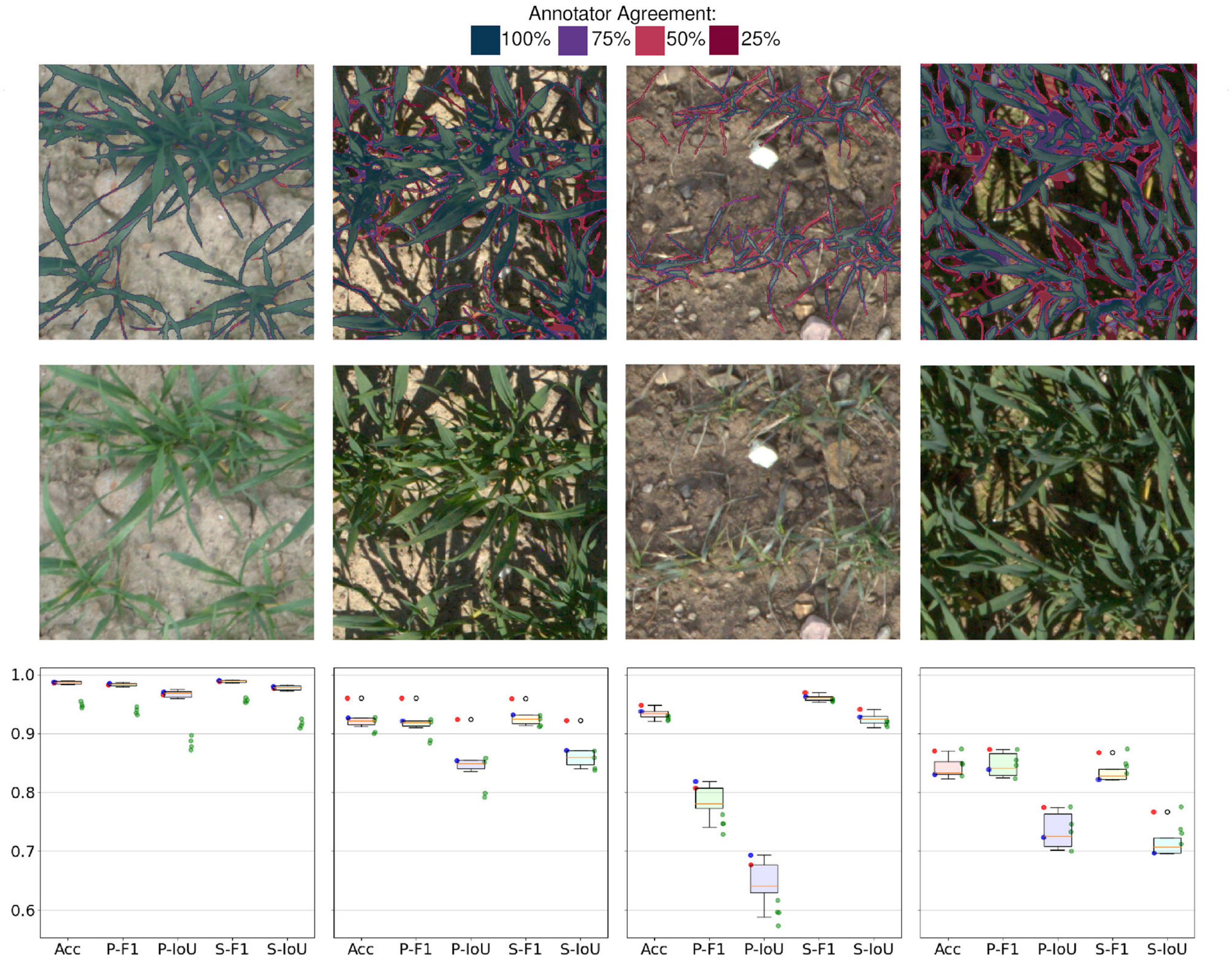


FIGURE 4 | The first row depicts the agreement of four different annotation attempts. The second row shows the original images. Please zoom in for details at the pixel level. Last row shows statistics of human annotations: Boxplot refers to 12 permutations of metrics resulting from 4 annotation attempts. Red and blue points refer to annotator 1 and annotator 2, respectively. Green dots show the performance of the proposed method. Acc: Accuracy, F1: F1-Score, IoU: Plants Intersection over union, prefix P stands for plants and prefix S for soil.

2.4.3. Architecture and Finetuning

In this experiment segment, elementary architecture concepts should be tested. This covers the ResNet encoder depth performance evaluated on two training sets with different sizes and the influence of finetuning the middle encoder blocks (refer to **Figure 3**). This experiment should provide some insights into the appropriate architecture choice and allow for observations with respect to the data volume used for training.

2.4.4. Feature Injection

Typically, only images are used as an input for image segmentation. However, deep neural networks can utilize additional information during training and/or predicting. The design of neural networks creates increasingly high-level information as the input passes through the network. In a typical case, color or brightness gradients are detected at first. Deeper in the network, edges will be recognized, and toward the end, whole objects, such as leaves, in our case will be identified. If there is some additional information available and this information is correlated with the objective, it should theoretically improve the performance of the network. The first problem that arises is to know where to inject this additional information. Introducing new information to the network at the wrong place can be ignored by the network or can even lead to a performance drop. This is due to the fact that the additional information has the greatest impact when introduced at the similar complexity of the features. The selected anchor points for feature injection in the ResNet encoder are after each of its building blocks. Which combination of blocks is the most suitable one needs to be determined during hyperparameter search. Another question that arises is how to add new inputs to a CNN, especially when the new information does not have a spatial dimension. This is solved by repeating the value up to the corresponding dimension of according feature maps. Afterward, the newly created feature map can replace one of the original ones or it can be concatenated at the end of the original feature maps. For the latter, the concatenation is followed by a 1×1 point-wise convolution in order to preserve the dimensions of the network.

2.4.5. Loss Selection

The selection of the loss function formulates the optimization objective, it directly influences the convergence and resulting performance. There are different variants of the loss functions that can either optimize for a specific metric of interest or for a latent function that is not directly measured. Examples for the first case are the Jaccard loss (see Equation 1) or Dice loss (see Equation 2). Note that in this work IoU is one of the main performance metrics that are being tracked and is directly stated by the Jaccard loss. The same applies to the Dice loss which is a direct restatement of the F1 Score.

An example for optimization of a latent variable is the crossentropy loss (see equation 3). Minimizing crossentropy corresponds to maximizing the probability of predicting a given class correctly while minimizing the probability of misclassification.

Since a loss function is a general numerical objective, a combination of losses is possible as well. For this scenario, the Dice crossentropy loss (see Equation 4) was tested.

$$\text{Jaccard loss} = 1 - \frac{x[c_{true}]}{1 + \sum_{i=0}^{N} x[i] - x[c_{true}]} \tag{1}$$

$$\text{Dice loss} = 1 - \frac{2 \cdot x[c_{true}]}{1 + \sum_{i=0}^{N} x[i]} \tag{2}$$

$$\text{Crossentropy loss} = -log(x[c_{true}])) \tag{3}$$

$$\text{Dice Crossentropy loss = Dice loss + Crossentropy loss} \tag{4}$$

where:

$$N = \text{number of classes}$$
$$c_{true} = \text{true class}$$
$$x[i] = \text{probability of class } i$$

Note that the equations above are stated for one individual sample.

2.4.6. Year Variability

Due to the small dataset size, a year-wise cross-validation experiment was conducted. This means that 1 year was kept for validation and testing while the remaining years were used for training. The allocation of images can be seen in **Figure 1**. Additionally, the validation and testing splits were rotated as well. The split of validation and testing data was conducted at random. The exact same model was then trained with the same parameters on different folds of the dataset.

This experiment should provide insights into the variance of the dataset with regard to its completeness and difficulty. In an ideal case with sufficient dataset size, the performance should however converge to the same value, as there would not be any unexpected cases that did not appear in the training data.

2.4.7. Data Augmentation

As the dataset consists of mere 190 images, data augmentation becomes an important part of artificially increasing the dataset size. Altering the images can produce new samples that can improve generalization as they leverage the prior knowledge about the task. This can be realized in form of classical operations such as random flipping, rotation, and cropping of the image. For humans it is clear that the augmented image is the same underlying data, but for the algorithm it is a brand new sample.

In addition, up- and down-scaling with bilinear interpolation were tested. The reasoning was to simulate the data at multiple scales, where down-scaling reduces the amount of data that needs to be processed and up-scaling provides pseudo data at higher resolution.

In order to address the changing lighting conditions, random jittering of contrast, saturation and brightness was implemented. Based on prior knowledge, small changes to these parameters should not have an effect on the segmentation. One might even argue that collecting more data will provide fluctuations to contrast, saturation, and brightness naturally.

To make up for camera dynamics, especially the amount of noise, Gaussian noise was applied to the input images at random. This step should resemble the noise that is contained naturally in the images and make the predictions more robust toward it.

All of the data augmentation methods are done randomly on the fly during training when the data is being loaded. Since training uses the data multiple times, it leads to different variations of the same image. This means that the training data is slightly altered every epoch. In the proposed setting, the network is trained for 150 epochs. This leads to 150 sets of augmented training images. As 154 images are used for training, this results in 28,500 different images.

2.4.8. Transfer Learning and Finetuning

Models that are trained on different datasets tasks can still deliver additional generalization even though the pretraining domain and the target domain are unrelated. Since the complexity of features increases with the network's depth, some of the earlier layers with low-level features such as gradients or edges do not need to change much when changing the domain. The idea of reusing the pretrained features while learning domain-specific complex features is called finetuning. During training, this can be enforced by freezing different layers while training parts of the network only. The frozen layers are still incorporated in the forward propagation of the input however their weights do not get updated. Which layers exactly should be preserved and which ones should be adapted, is a matter of finding the best performing combination. Typically, the layers of a network are iteratively being frozen by additionally freezing deeper layers and assessing

TABLE 2 | Benchmarks on the EWS dataset.

Benchmark	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
Presegmentation	0.836	0.568	0.657	0.782	0.873
Yu et al. (2017)	0.917	0.666	0.779	0.866	0.925
Sadeghi-Tehran et al. (2017)	0.903	0.638	0.760	0.845	0.912
Rico-Fernández et al. (2019)	0.909	0.691	0.805	0.839	0.908
DeepLab v3+ ResNet50	0.924	0.707	0.814	0.866	0.926
DeepLab v3+ Pretrained ResNet50	0.938	0.747	0.842	0.888	0.939
Proposed method	**0.945**	**0.775**	**0.863**	**0.899**	**0.951**

Trained on 2018–2020, reporting on the 2017 subset.

the overall performance. In this work, in order to decrease the number of needed experiments, whole blocks of layers (see building blocks in He et al., 2016) were iteratively frozen. In addition, combinations of deep and shallow blocks were trained and their performance was observed. This enables for the option where not only the highly specific features need to be updated but the low level features as well. The reasoning behind this is that color is a crucial characteristic of plants and the optimal color transformations which occur early in the network might require adjustments for better performance. The overall depth of the network and availability of training data also influences the learning dynamics in terms of transfer learning and finetuning. On the one hand, deeper networks are more prone to overfitting when retrained finetuned on limited data. On the other hand, deeper networks are able to transfer their larger generalization capabilities from the original domain compared to their shallow counterparts. Therefore, a trade-off in transferred generalization and efficient adaptation to the new domain based on the selection of the network depth and the finetuning mode is to be expected.

2.4.9. Input Data Transformation

Based on the methodology used in remote sensing and manual or automatic thresholding, a number of different hand engineered features and visual indices are used to enhance the contrast between the plants and soil. According to the contributions of Milioto et al. (2018), a selection of these hand engineered features can be used jointly with a deep convolutional network. Therefore, a test with additional inputs to the proposed method was conducted. In addition to the normal RGB inputs, different sets of additional inputs were tested. **Table S1** shows an overview of different transformation sets. Also, note that stacking different transformations of an image on top of each other greatly increases the necessary GPU memory and therefore has to be compensated with for example lower batch size. Additionally, the exact implementation of feature transformation is unknown therefore the results need to be taken with caution.

3. RESULTS

3.1. EWS Dataset Benchmark

The achieved benchmarks of the tested methods (see section 2.4.1) can be seen in **Table 2**. Additional numerical insights to the statistical significance of individual metrics are reported in **Appendix 2.7**.

The presegmentation method performs the worst on every tracked metric. We see a major improvement when moving toward (Sadeghi-Tehran et al., 2017; Yu et al., 2017). Both of these methods use machine learning approaches on individual pixels independently. Next, Rico-Fernández et al. (2019) present another advance in performance. This method explicitly incorporates pixel neighborhood and enables for neighboring regions interactions.

Moving on to deep learning based methods that implicitly use relations between neighboring pixels, another boost in performance can be seen when training a DeepLab v3+ ResNet50 purely on the EWS dataset from scratch. The performance was further improved by utilizing pretrained weights. Additionally, implementing a combination of supplementary techniques which represent the proposed method pushed the benchmark even further. With respect to the performance of this method, various sources of error can be linked to the quality of the labels and to the algorithm (see **Figure 5**). Prediction examples can be seen in **Appendix 1.1**. For the performance comparison of different methods refer to section 3.1.

3.2. Human Annotations in Perspective

Human annotators deliver a solid, consistent performance when dealing with diffuse light and high contrast images (shown in 1st column in **Figure 4**). The only inconsistencies arise on the boundaries of leaves or consider very thin parts of leaves. The performance evaluated on IoU and F1 score is well above 0.95 with little overall variance in the metrics. As soon as the complexity of the scene increases due to shadows, thinner leaves, or ambiguous classification of vegetative active or inactive material, the performance of annotations drops. Various degrees of shadows in the images (shown in 2nd and 4th column in **Figure 4**) lead to worse overall results but what is worth noting is that a performance gap between the annotators becomes visible. This occurs because the underexposed areas in the images are hard to classify as plant or soil due to the low signal-to-noise ratio. Another effect that can be observed in human annotations is the different interpretations of plant parts in the image (shown in 3rd column in **Figure 4**). The score of individual annotators indicates that they exhibit higher consistency within the same

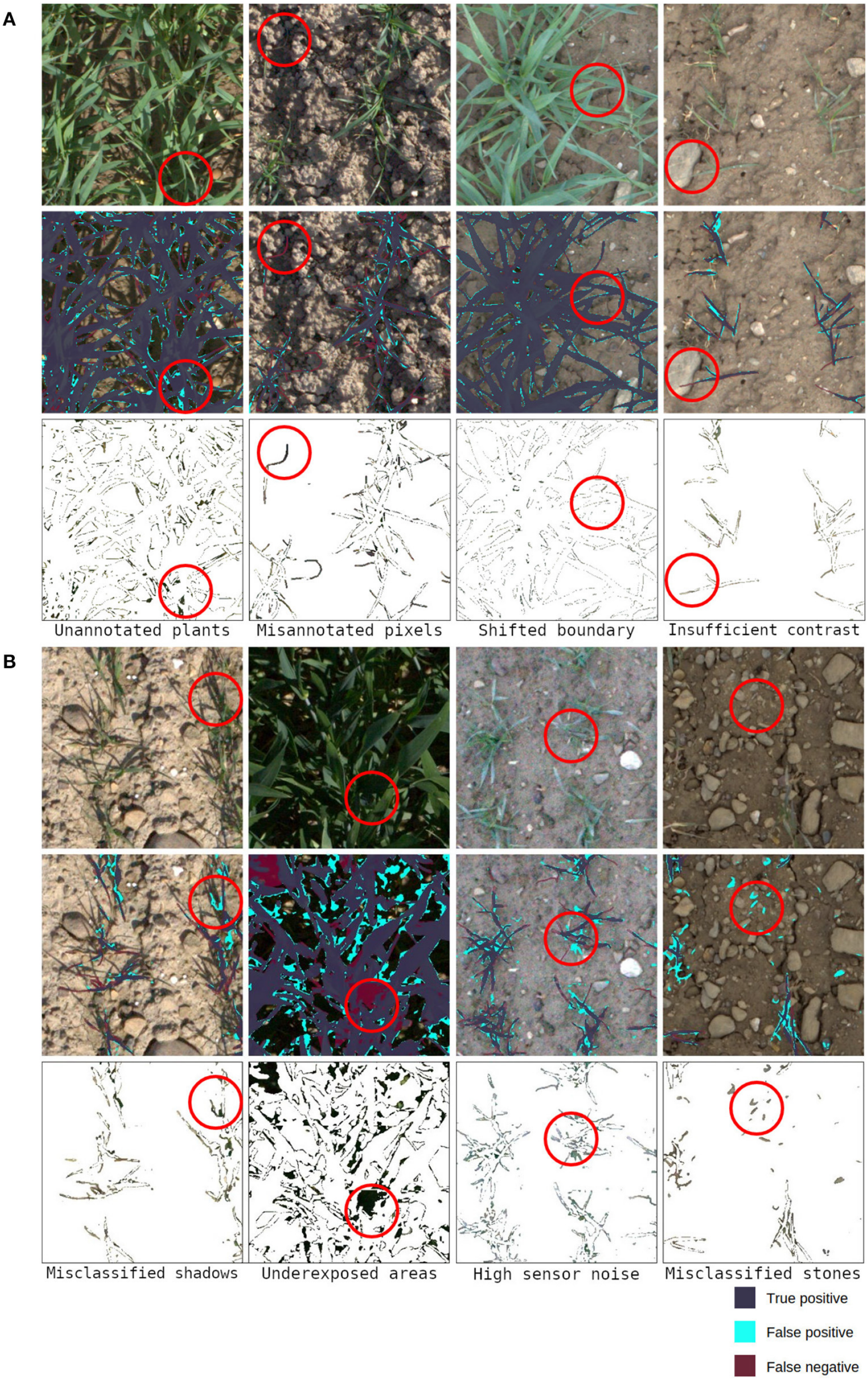


FIGURE 5 | Examples of sources of error: **(A)** depicts cases linked to inconsistencies in labels, **(B)** shows failure cases of the algorithm. 1st row-original image, 2nd row-evaluated predictions, 3rd row-underlying values of misclassified parts of the image. Please do zoom in for inspection on the pixel level. The highlighted areas will be referred to in sections 3.2 and 4.

TABLE 3 | Influence of different ResNet encoder depths and dataset sizes.

ResNet depth	Training images	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
18	142	**0.945**	0.760	0.849	0.904	0.948
18	76	0.937	0.740	0.836	0.893	0.942
34	142	**0.945**	**0.763**	**0.852**	**0.906**	0.942
34	76	0.940	0.756	0.849	0.896	0.943
50	142	**0.945**	0.757	0.843	0.905	**0.949**
50	76	0.943	0.761	0.851	0.900	0.946

TABLE 4 | Influence of finetuning different ResNet encoder depths.

ResNet depth	Frozen layers	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
18	No	0.945	0.760	0.849	0.904	0.948
18	Yes	0.944	0.754	0.842	**0.908**	**0.950**
34	No	0.945	0.763	0.852	0.906	0.942
34	Yes	0.945	0.762	0.850	0.905	0.949
50	No	0.945	0.757	0.843	0.905	0.949
50	Yes	**0.947**	**0.767**	**0.853**	**0.908**	**0.950**

annotator for both annotators. A plausible explanation for this is that different annotators hold different but consistent opinions on what should be considered a part of a plant. This can be seen in the data because both annotators are in the upper percentile of performance while the cross-annotator performance is notably worse. However, these interpretations need to be taken with caution due to the limited sample size of observations. The success cases depicted in **Figure 5A** represent some of the annotator uncertainty. The highlighted area of the first image shows an area that was predicted as part of a plant and was skipped by the annotator. The second and the last images show a part of a plant that should not be annotated as vegetative active material. The third image shows that in good lighting conditions only minor disagreements along with the leaves' boundaries are present. The failure cases of the proposed method are presented in **Figure 5B** which shows problematic scenarios. The first image corresponds to a bright scenario where the plants' shadows are misclassified. The second image points out problematic underexposed areas due to the high dynamic range of the image. The third image shows the difficulties of the network when dealing with high sensor noise due to low light. Finally, the last image demonstrates the scenario with limited contrast, where stones get misclassified as parts of the plants.

3.3. Architecture and Finetuning

The commonly used DeepLab v3+ was selected as an architecture of choice. The underlying backbone is the well-proven ResNet. In order to mitigate the size of the dataset, imagenet-pretrained ResNet weights were applied. The influence of different ResNet architectures as backbones was analyzed and is reported in **Table 3**. This has shown that more complex ResNet50 outperforms thinner ResNet18 on limited data and that ResNet34 performs the best when trained on the whole dataset.

Table 4 reports the performance with and without freezing layers from middle blocks 2 and 3 of a ResNet. The results show that ResNet50 benefits the most from freezing layers while ResNet18 experiences even a performance drop. Meanwhile, ResNet34 achieves comparable performance regardless of freezing layers. These results can be interpreted as an improved way to preserve the generalization capabilities from the pretraining domain and reduce potential overfitting. Note that results with freezing layers introduce a better score with ResNet50 than training the whole ResNet34 network from the previous experiment (shown in **Table 3**).

3.4. Feature Injection

In the following experiments, the potential of injecting various metadata into the image segmentation network is shown. Data from 3 different categories were included. It consists of sensor data (ISO, F-Number, exposure time) and knowledge about the scene (date, time, and GDD). Note that all these extra inputs are available during the inference. The performance with injected features according to the strategy depicted in **Figure 3** is reported in **Table 5**. The most beneficial features to inject was the combination of ISO, F-Number, and exposure, however, the introduced benefits are limited. The inclusion of date and time also led to a subordinate improvement. The benefits of using GDD or ISO alone are limited.

3.5. Loss Selection

The selection of potential losses was based on the common losses that are used by the scientific community. This covers

TABLE 5 | Feature injection influence when using different metadata.

Additional inputs	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
None	0.945	0.757	0.843	0.905	**0.949**
GDD	0.947	0.762	0.852	**0.905**	**0.949**
Date, Time	**0.949**	0.767	0.857	0.902	0.946
ISO	0.945	0.761	0.853	0.902	0.945
ISO, F-Number, Exposure	0.946	**0.770**	**0.861**	0.903	0.948

TABLE 6 | Testing of various losses.

Loss	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
Dice loss	0.936	0.726	0.823	0.890	0.940
IoU loss	0.936	0.721	0.818	0.891	0.940
Dice crossentropy loss	**0.946**	0.766	0.852	**0.906**	0.949
Crossentropy loss	**0.946**	**0.772**	**0.859**	**0.906**	**0.950**

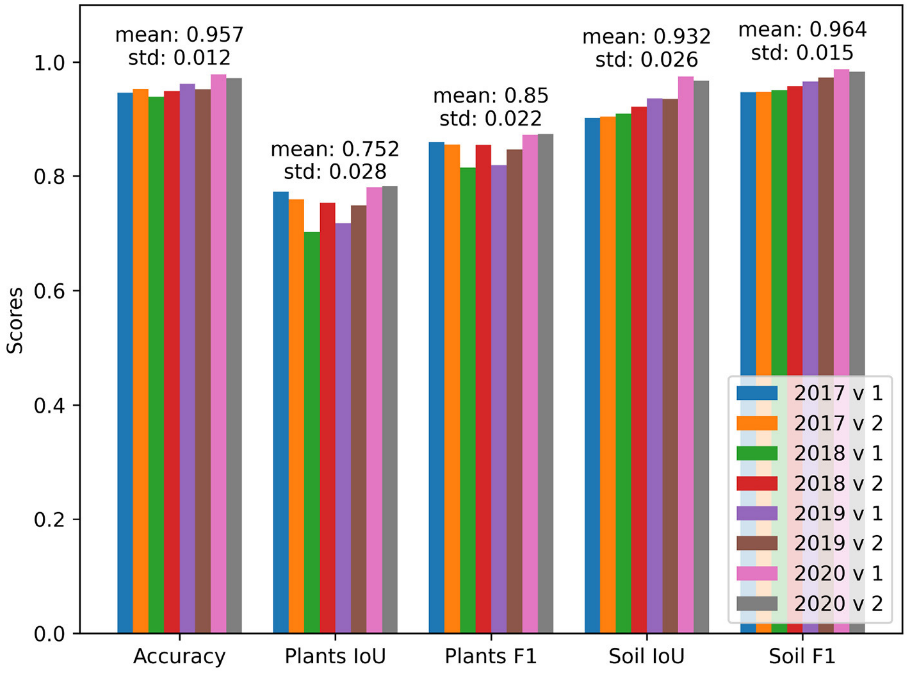


FIGURE 6 | Performance on different folds of the EWS dataset. The years in the Figure's legend refer to the year used for validation and testing. The "v1" and "v2" refer to the permutations of testing and validation subset within the selected year, respectively.

the crossentropy loss, dice loss, IoU loss, and dice crossentropy loss. Their relative performance can be compared in **Table 6**. The crossentropy loss achieved the best overall performance. Note that it overperformed the IoU loss on the IoU metrics even though IoU loss directly optimizes for those.

3.6. Year Variability

The results from training on both allocations and different folds are reported in **Table 6**. The reported performance indicates that the choice of a subset for validation and testing introduces fluctuations to the model performance. The differences in performance come especially from the distribution of the challenging samples. The different folds of the dataset within the same year were selected at random. The fact that a random split into folds has an effect on the performance (shown for example 2018 v1 in **Figure 6**) can be interpreted as insufficient dataset size and/or insufficient representation of different lighting conditions. In this case, the worst performing fold was negatively influenced during testing by more difficult images with a slight snow cover and high sensor noise due to low light.

3.7. Data Augmentation

In the following experiment, the influence of data augmentation was analyzed. The benchmark consists of random horizontal and

TABLE 7 | Data augmentation ablation study.

Augmentation method	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
w/o upscaling	0.938	0.743	0.833	0.890	0.939
w/o rotation	0.945	0.757	0.845	0.903	0.947
w/o color jitter	**0.947**	0.767	0.859	**0.908**	**0.951**
w/o noise	**0.947**	0.772	0.860	0.907	0.949
Proposed method	0.945	**0.775**	**0.863**	0.899	**0.951**

TABLE 8 | Influence of pretrained weights from the Imagenet and 2016 Sugar Beets dataset for transfer learning.

Pretraining method	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
No pretraining	0.939	0.750	0.8451	0.892	0.941
Imagenet	**0.946**	**0.772**	**0.859**	**0.906**	**0.950**
Imagenet + Sugar Beets 2016	0.943	0.767	0.857	0.895	0.943

vertical flipping, cropping to 224 × 224 px, and rotation up to 20 degrees. Consequently, multiple different modules were tested. The first one consisted of adding Gaussian noise to the data. Afterward, different jittering of brightness, saturation, and contrast was applied. Finally, upscaling the image to a higher resolution was tested. The results of the module ablation study can be seen in **Table 7**.

The greatest impact comes from upscaling to 200% size. This positively impacts the performance of all the different encoder depths. Since upscaling is a basic bilinear interpolation, the most probable hypothesis is that the size of visual cues in the images is at its limit. This can be interpreted as visual cues possibly being too small. This is analog to the testimonies of annotators who state that the resolution is too low to accurately label thin parts of leaves.

The positive impact of randomly rotating the image can be interpreted as an extension of the dataset size. However, one has to note that due to the resolution limits, the rotation can worsen the image quality on the critical parts of the image that are already being on the edge of being correctly classified. So, the benefits of extending the dataset can be at the cost of vague plant boundaries during training.

Color jittering in the images has a positive influence on the performance. The rationale is that this contributes to artificially increasing the size of the dataset. However, during the selection of individual jittering parameters, the performance of the network started to suffer as the jittering became more aggressive. With selected parameters of random jittering of brightness up to 1%, saturation up to 25%, contrast up to 10%, and no jittering in hue, the network became more robust to changes in lighting; however, as the mild performance difference suggests this cannot substitute natural light changes, a larger dataset with more samples for diverse lighting conditions.

Introducing a zero mean Gaussian noise with a SD of 0.001 should help to mitigate the noise coming directly from the camera sensor. Since most of the images are taken with ISO 100 which means a relatively high signal-to-noise ratio (shown in ISO histogram in **Figure 2**) introducing small noise fluctuations to the images means should make the network more robust toward sensor noise. However, it is decreasing the overall image quality for the benefit of few images burdened with sensor noise captured with high ISO. This yields a plausible explanation why introducing noise is not particularly effective why further increasing SD leads to a performance drop.

3.8. Transfer Learning

Since the number of images for training is limited, transfer learning becomes an important part of the training pipeline as it can yield additional generalization. The performance of training from scratch and using a pretrained network was compared. More accurately, two different datasets were used for pretraining, namely, the ImageNet and the Sugar Beets 2016 dataset (Chebrolu et al., 2017). This has led to three different pretraining methods which are compared in **Table 8**. The data shows that there is an obvious benefit of using pretrained weights compared to training from scratch. The difference between using only ImageNet weights or ImageNet weights trained on Sugar Beets 2016 dataset as a starting point results in slightly better performance for using only the ImageNet. A possible explanation for the cause of this behavior is differences between the datasets. In contrast to Sugar Beets 2016 EWS dataset operates on denser, uncovered plants where the plant and soil appearance changes based on the weather, lighting conditions, and the date. This also involves having different crops as the main objective, namely, sugar beet and winter wheat. While Imagenet consists of a very larger set of highly diverse classes, the Sugar Beets 2016 is highly specialized in terms of the scene composition and objective. Therefore, it is possible that pretraining on Sugar Beets 2016 starting with Imagenet weights might offer limited additional knowledge about our task.

3.9. Input Data Transformation

The results from feeding stacked color transformations to the network during training can be seen in **Table 9**. None of the introduced transformations improved the performance. This may be linked to the learning capability of the network, which can

TABLE 9 | Input transformations, for overview of transformations refer to **Table S1**.

Input set	Pixel accuracy	Plants IoU	Plants F1	Soil IoU	Soil F1
RGB	**0.947**	**0.777**	**0.863**	**0.905**	**0.948**
Set 1	0.941	0.759	0.852	0.893	0.941
Set 2	0.944	0.759	0.851	0.900	0.945
Set 3	0.946	0.762	0.853	0.904	0.948

extract such transformations directly from the data when trained end-to-end. Additionally, it may be linked to the fact that the network uses weights that are pretrained on pure RGB dataset and can therefore be re-learning features from scratch, overfitting to the new inputs.

4. DISCUSSION

This section provides an interpretation of high-level concepts resulting from the learnings gained during this work. The first part of the discussion is dedicated to the dataset. It consists of its potential, shortcomings, and proposed improvements for future work. The second part is assessing the use of deep learning for the segmentation of field-grown plants with a focus on the methodology developed in this work.

4.1. Eschikon Wheat Dataset

The EWS dataset is a new field segmentation dataset that offers various metadata in addition to the images. While some of them were leveraged in this work (see section 3.4), others (e.g., temperature, location in the field) remain unused. An important asset of the EWS dataset is the uncontrolled lighting conditions that the photographed canopies were exposed to over multiple years in high temporal resolution and large number of phenotypes. However, the vast majority of the acquired data still remains unlabeled. The amount of annotated data is the biggest shortcoming of the EWS dataset in its current form. As shown in **Table 3** increasing the dataset size resulted in a performance boost of the image processing pipeline. This behavior is expected to continue with further increases in dataset size. But increasing the dataset size is not the full challenge. Human annotations can become tricky as soon as the image quality decreases. Therefore, the goal for a future expansion of EWS is gathering new data with high quality human annotations.

4.1.1. Temporal Variance

Due to the long runtime of the field experiments, the introduced dataset contains a high amount of different lighting conditions settings. One of the repeating scenarios for failures in predictions is different lighting. In general, this results in low contrast of plants with respect to the soil, a large portion of underexposed shadows, and a high amount of noise (see **Figure 5B**). This behavior comes from the lighting distribution which is linked to the different weather patterns occurring each year. Ideally, the dataset would contain a sufficient amount of data so that the performance is constant between the years. But as seen in **Table 6** the performance of the different years is varying. With more data, the metrics should ideally converge to similar results. When the performance of the algorithm would converge to the same value, it would indicate that the network is able to generalize well over all relevant weather and plant patterns that are contained in the data and that the dataset contains an adequate representation of the data for every year.

4.1.2. Image Quality

Improving the exposure with techniques, such as HDR, would also increase the quality and consistency of the data while decreasing the semantic ambiguity of parts of the images due to the high dynamic range of outdoor plants and soil. Addition, in the current setting, when the leaves are not perpendicular to the camera view but rather rotated in some direction, they are very thin on the imaging plane leading to mixed pixels of plants and soil in the extreme. The amount of mixed pixels can be decreased by using a higher physical resolution. Alternatively, multiple viewpoints could be used to prevent the very thin leaves projections. However, multiple images would have to be taken simultaneously due to the possible movement of the plants as a result of external influences, such as wind. Also, the introduction of a multi camera approach would allow for the extraction of depth which would add another information layer to the acquired data.

4.1.3. Dataset Expansion

Since it is expected to get better performance with larger dataset size, annotation of new images is going to be a part of future developments. As the dataset size increases, the optimization of annotation workflow becomes a crucial element that can potentially save a great part of the expensive annotation efforts. The EWS dataset was created using approximately 80 human annotation hours for 190 images. This amount of required annotation time per image can be optimized in the future through specialized annotation frameworks that offer fast workflows and support pre-segmentation active learning with already trained methods (for examples such as CVAT[9], Lightly [10], Labelbox[11], Supervisely[12]).

However, the provided labels and the corresponding labeling strategy can be improved based on the annotation artifacts (shown in **Figure 5A**). The current labels contain small amounts of high-frequency noise in form of holes or left out plant parts that exhibit low contrast, sharpness, or are underexposed in general. Besides that, the distinction between vegetative active and inactive material is not always easily visible, the introduction

[9] https://cvat.org/

[10] https://www.lightly.ai/

[11] https://labelbox.com/

[12] https://supervise.ly/

of more classes might show beneficial in the future. The distinction between soil and plant material which is then further divided into active and inactive material should decrease the room for personal interpretation from the annotators. This two-step classification should yield more consistent data as it would not miss out on any plant pixels due to different annotator's interpretations and the second step of annotations can be easily tuned during dataset revision. Ultimately, the misclassifications of plants in favor of soil could be penalized differently when inactive plant material is misclassified.

Fortunately, there is a vast number of images to choose for future annotations. The most logical next step would be to keep adding more different dates to the dataset in order to improve the coverage of the varying outdoor conditions.

Another approach would be to keep adding images where the prediction confidence (the difference between class probabilities) is the lowest. These samples should be theoretically the most beneficial ones as they provide information for the edge cases, where the network is unsure about its predictions.

Furthermore, the fact that the images represent a growth cycle can be leveraged for performance quantification. Since the plant growth dynamics can be approximated to a canopy cover measure which monotonically increases as the plants mature, potential outliers can be identified by inspecting the canopy cover development over time. These outliers can then be labeled and used for training to improve the overall performance.

With the increasing size of training data, the dynamics of the presented approaches will change. This effect can be seen in section 3.3. First, more complex networks yield higher generalization capabilities when trained on limited data (see **Table 3**) due to their larger amount of already trained features. However, when trained on all available data, this relationship changed in favor of less complex networks and especially ResNet34 because simpler networks are able to adapt faster and with less potential overfitting to the new domain.

Nonetheless, the benefit of finetuning pretrained deeper networks is expected to eventually decrease when the amount of training data is increased (Soekhoe et al., 2016). When a pretrained network is trained on a large dataset the importance of preserving pretrained features will diminish as more relevant and specialized features for the task can be extracted directly from the data.

4.2. Deep Learning for Outdoor Agriculture

The proposed deep learning algorithm achieves a solid performance on the EWS dataset even with its challenging dataset size. It is hard to estimate how accurate is human performance without labeling a major part of the dataset multiple times. Looking at the performance of the proposed segmentation algorithm (see green dots in **Figure 4**), multiple performance patterns can be identified. The first image with good contrast and diffuse light shows the consistently worse performance of the algorithm compared to the human annotators, while still achieving solid performance (around 0.95 on all tracked metrics). During the direct light and good contrast scenario in the second image, the different annotators and the algorithm show performance with the high variance between the annotators and between the algorithm based on which annotation attempt is considered ground truth. For the remaining two samples, the performance of the algorithm is well within the variance of the human annotators. This means that the worse performing samples show a similar agreement between the annotators and the algorithm. Increasing the agreement of human annotations would be beneficial to the method and would deliver more consistent benchmarking as well.

Using deep learning based methods yields important additional benefits besides the superior performance as described in **Table 2**. First, the abundant expressivity of deep neural networks leads to a buffer in their pattern learning capabilities. Thus, their performance scales with the data as more complex patterns (for example with regard to the growth stage or weather) can be learned from the additional information. Furthermore, neural networks are capable of dealing with large datasets by design. This is further utilized by the contemporary deep learning frameworks (such as Pytorch or Tensorflow) that are heavily runtime optimized and yield scalable approaches. This can be especially seen in the form of utilizing GPUs and distributed learning and/or inference which scale with the available hardware. This makes for a clear differentiation in comparison to approaches like SVC as proposed in Rico-Fernández et al. (2019) that struggle with larger data volumes due to their current single threaded CPU implementation. In addition, the proposed contextual information can be learned implicitly by using convolutions in the neural net architecture that are capable of extracting patterns not only in the color space input but in the feature space as well. Ensemble methods such as Random Forrest as proposed by Sadeghi-Tehran et al. (2017) offer better paralellization capabilities as the individual predictors can be trained simultaneously. However, their vanilla implementation does not account for any spatial patterns. Thus, each individual pixel is handled independently which misses out on any spatial information and most probably contributes to the performance gap. Spatial information in general is an additional layer of data for prediction making. The benefits of incorporating spatial information into the method would be even more important for other tasks such as semantic segmentation of multiple plant species as for example the shape of the leaves, plant center or the amount of dead plant tissue are crucial species features.

4.2.1. Training on Limited Data

Deep neural networks are capable of extracting and learning useful information from large datasets. When training on limited data, they are prone to overfitting and thus can deliver poor results. This issue can be mitigated by employing different approaches such as fine tuning and data augmentation.

The use of fine tuning technique, where multiple layers were frozen, was beneficial especially for ResNet50 (see **Table 4**) as it limited the amount of parameters that were being optimized and thus reduced the overfitting potential. A possible explanation for the best performance with freezing the middle layers might appear due to the strong visual color cues that plants exert. Color cues should appear relatively early in the network, and it can therefore be beneficial to retrain the early layers as well. In this way, network can learn new low-level features, such as color transformations, from the target domain and combine them with highly specialized features at the later stages of the network.

The data augmentation did indeed improve the performance of the network (see **Table 7**) as it artificially alters the images and thus increases the dataset size. The upscaling of the image showed the greatest improvements, whereas the remaining modules show only minor changes in performance. An interesting phenomenon is the brightness, color, and contrast jittering as the data augmentation method. From the problem description, the lighting seems to be one of the key bottlenecks of performance. However, its impact on the overall performance did not fulfill the expectations of being the key element of the data augmentation pipeline. This might be due to the number of lighting conditions already contained in the dataset and the resulting generalization of the network with respect to lighting. Another possible explanation is that the color jittering does not greatly represent the real changes in color and therefore might not be generating accurate variations to lighting conditions.

4.2.2. Leveraging Metadata

The introduced dataset provides a lot of metadata in addition to the images. The collected metadata is common in agricultural applications, as camera parameters are stored as Exchangeable Image File Format (EXIF) and weather station is a frequently used equipment. The network benefits from using different metadata as they can reveal high-level information about the scene (see section 3.4). Using camera parameters as additional inputs led to minor improvements. The camera parameters correlate with the luminance of the imaged area and affect the quality of the image along with the noise dynamics. The impact of this approach with respect to dataset size is up to a discussion as the network can either learn the information from the pure image data or the benefits of injecting metadata can become more relevant as more data is provided for training.

Note that feeding additional inputs is not the only possibility for leveraging the metadata. Alternative approaches, such as sample weighting based on metadata, multitask learning for additional generalization and/or pretraining for metadata classification, regression, are good candidates for future work.

4.2.3. Future Opportunities and Remaining Challenges

While Deep Learning methods can be applied on datasets with limited data, a possibility of standardized benchmarking on a large dataset is missing. This in fact makes the search for the current state of the art in agricultural applications extremely time intensive and replication difficult.

We see a great opportunity in broad collaboration of different phenotyping research stations as it is a key for moving toward a universal dataset. Since the imaging method of RGB imagery from a nadir view is common in the phenotyping community, it should be possible to combine partial datasets into a central one. In addition, individual research groups usually operate in a fixed locations. When multiple research groups would contribute to a public dataset, the regional variance between the location would be contained in the data. Afterward, researchers could optimize their focus to keep improving the best performing methods.

Another opportunity is that in the discipline of high-throughput field phenotyping, research stations typically produce large amounts of images. The relevant analysis pipelines are developed only using a small annotated subset of the available data, with the rest of the data remaining unused in the process. Therefore, exploring different modes of learning such as semi-supervised learning, weakly supervised learning, and/or sophisticated data curation might offer additional benefits as a significantly larger amount of data could be used in the development process.

One of the major challenges in this application is that when the imaging method is updated and new data is being collected. Multiple years are required in order to get at least a small sample of the possible variances in the lighting conditions, weather patterns. Therefore, the iteration cycle for the method development is very long unless the old data can be reused in spite of a different imaging method.

4.3. Conclusion

Semantic segmentation for phenotyping is yet another discipline for contemporary deep learning research. This work provides insights into the challenges of outdoor computer vision applications in agriculture, a metadata-rich segmentation dataset, and methods for an additional performance boost of typical segmentation architecture. Due to the limited availability of large scale datasets, training on a challenging amount of data needs to be addressed.

An approach in form of the established DeepLab V3+ architecture with custom adjustments to the training pipeline and mild changes to the architecture delivers a solid performance close to human annotator variance, which was calculated on an inspection dataset subset (shown in **Figure 4**). Failures occur when the physical resolution of the camera is too low and/or in extreme lighting conditions. The shortcomings due to the limited dataset size can be mitigated with techniques that utilize transfer learning (see section 3.3), augmenting the training data (see section 3.7), or injecting additional information as additional inputs (section 3.4). Even on a small dataset, the deep learning based proposed method outperformed the benchmarked machine learning based methods (see section 3.1). The benchmarked machine learning based methods showed a better performance with an increasing number of input transformations and by considering neighboring pixels. The superior performance of deep learning methods results from learning the so far hand-selected relations implicitly and directly from the data. The superior performance of deep learning is expected to further scale with additional data and expand the performance gap.

The presented dataset is the first dataset to cover the same field over multiple years with a number of different lighting conditions scenarios (shown in **Table 1**). The proposed method achieved the best performance compared to the selected methods used in the scope of phenotyping (shown in **Table 2**). Even at this limited dataset size, the deep learning based approach is able to outperform its machine learning counterparts and therefore the dataset size threshold for feasible deep learning is lower than one might think. Furthermore, the performance of the proposed method is expected to further increase when more data is labeled and/or the shortcomings of the dataset are addressed. In this context, high resolution images with a sufficient dynamic range are the key for further development as human annotators reach their limits due to ambiguous cases where the labels vary throughout multiple attempts and lead to inconsistencies even when labeled by the same person (see **Figure 4**).

A high quality, large-scale dataset would benefit the scientific community as the high soil and lighting conditions variance is the hardest problem that is yet to be solved (see **Figure 5**). In addition, a standardized benchmark is currently missing in the research cycle as most methods are reported on their own data whilst code availability is a bottleneck for reproducibility and method comparison.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are publicly available. The dataset is stored at: https://www.research-collection.ethz.ch/handle/20.500.11850/512332 with the reserved doi: 10.3929/ethz-b-000512332. The code is available at: https://github.com/RadekZenkl/EWS.

AUTHOR CONTRIBUTIONS

RZ wrote the manuscript, initiated the project, designed, and implemented the methodology. The results were interpreted by RZ, RT, HA, and NK. AH, LR, AW, and LV contributed to the manuscript. RT contributed to the method and manuscript. HA contributed to the manuscript and coordinated the project. NK contributed to the manuscript and was responsible for data acquisition and pre-processing. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Swiss National Science Foundation in the framework of the project PhenoCOOL (project no. 169542).

ACKNOWLEDGMENTS

We thank DSP for providing seeds for the experiments. We specially thank Jonas Anderegg and Lukas Kronberger who helped to acquire the data. Last but not least thanks to Hansueli Zellweger and Brigitta Hergoz who took care of the field management.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.774068/full#supplementary-material

REFERENCES

Aasen, H., Honkavaara, E., Lucieer, A., and Zarco-Tejada, P. J. (2018). Quantitative remote sensing at ultra-high resolution with uav spectroscopy: a review of sensor technology, measurement procedures, and data correction workflows. *Remote Sens*. 10:1091. doi: 10.3390/rs10071091

Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., et al. (2019). Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. *Comput. Electron. Agric*. 167:105091. doi: 10.1016/j.compag.2019.105091

Aich, S., and Stavness, I. (2017). "Leaf counting with deep convolutional and deconvolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

Andrade-Sanchez, P., Gore, M. A., Heun, J. T., Thorp, K. R., Carmo-Silva, A. E., French, A. N., et al. (2013). Development and evaluation of a field-based high-throughput phenotyping platform. *Funct. Plant Biol*. 41, 68–79. doi: 10.1071/FP13126

Araus, J. L., and Cairns, J. E. (2014). Field high-throughput phenotyping: the new crop breeding frontier. *Trends Plant Sci*. 19, 52–61. doi: 10.1016/j.tplants.2013.09.008

Bai, G., Ge, Y., Hussain, W., Baenziger, P. S., and Graef, G. (2016). A multi-sensor system for high throughput field phenotyping in soybean and wheat breeding. *Comput. Electron. Agric*. 128, 181–192. doi: 10.1016/j.compag.2016.08.021

Bai, X., Cao, Z., Wang, Y., Yu, Z., Hu, Z., Zhang, X., et al. (2014). Vegetation segmentation robust to illumination variations based on clustering and morphology modelling. *Biosyst. Eng*. 125, 80–97. doi: 10.1016/j.biosystemseng.2014.06.015

Breiman, L. (2001). Random forests. *Mach. Learn*. 45, 5–32. doi: 10.1023/A:1010933404324

Burkart, A., Hecht, V., Kraska, T., and Rascher, U. (2018). Phenological analysis of unmanned aerial vehicle based time series of barley imagery with high temporal resolution. *Precision Agric*. 19, 134–146. doi: 10.1007/s11119-017-9504-y

Candiago, S., Remondino, F., De Giglio, M., Dubbini, M., and Gattelli, M. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from uav images. *Remote Sens*. 7, 4026–4047. doi: 10.3390/rs70404026

Carlson, T. N., and Ripley, D. A. (1997). On the relation between ndvi, fractional vegetation cover, and leaf area index. *Remote Sens. Environ*. 62, 241–252. doi: 10.1016/S0034-4257(97)00104-1

Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., and Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *Int. J. Rob. Res*. 36, 1045–1052.doi: 10.1177/0278364917720510

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 801–818.

Chiu, M. T., Xu, X., Wang, K., Hobbs, J., Hovakimyan, N., Huang, T. S., et al. (2020). "The 1st agriculture-vision challenge: methods and results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (Seattle, WA: IEEE).

Chollet, F. (2017). "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 1251–1258.

David, E., Madec, S., Sadeghi-Tehran, P., Aasen, H., Zheng, B., Liu, S., et al. (2020). Global wheat head detection (gwhd) dataset: a large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics* 2020:3521852. doi: 10.34133/2020/3521852

Fiorani, F., and Schurr, U. (2013). Future scenarios for plant phenotyping. *Annu. Rev. Plant Biol*. 64, 267–291. doi: 10.1146/annurev-arplant-050312-1 20137

Fuentes-Pacheco, J., Torres-Olivares, J., Roman-Rangel, E., Cervantes, S., Juarez-Lopez, P., Hermosillo-Valadez, J., et al. (2019). Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sens*. 11:1157. doi: 10.3390/rs11101157

Guo, W., Rage, U. K., and Ninomiya, S. (2013). Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model. *Comput. Electron. Agric.* 96, 58–66. doi: 10.1016/j.compag.2013.04.010

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.

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), 4700–4708.

Hund, A., Kronenberg, L., Anderegg, J., Yu, K., and Walter, A. (2019). Non-invasive field phenotyping of cereal development. *Adv. Breed. Techniq. Cereal Crops* 249–292. doi: 10.19103/AS.2019.0051.13

Jung, H., Choi, M.-K., Jung, J., Lee, J.-H., Kwon, S., and Young Jung, W. (2017). "Resnet-based vehicle classification and localization in traffic surveillance systems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (Honolulu, HI: IEEE), 61–67.

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

Kirchgessner, N., Liebisch, F., Yu, K., Pfeifer, J., Friedli, M., Hund, A., et al. (2017). The eth field phenotyping platform fip: a cable-suspended multi-sensor system. *Funct. Plant Biol.* 44, 154–168. doi: 10.1071/FP16165

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105. doi: 10.1145/3065386

Lin, B., Xie, J., Li, C., and Qu, Y. (2018). "Deeptongue: tongue segmentation via resnet," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Calgary, AB: IEEE), 1035–1039.

Liu, S., Baret, F., Andrieu, B., Burger, P., and Hemmerle, M. (2017). Estimation of wheat plant density at early stages using high resolution imagery. *Front. Plant Sci.* 8:739. doi: 10.3389/fpls.2017.00739

Milioto, A., Lottes, P., and Stachniss, C. (2018). "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 2229–2235.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classifiers* 10, 61–74.

Pretty, J., Sutherland, W. J., Ashby, J., Auburn, J., Baulcombe, D., Bell, M., et al. (2010). The top 100 questions of importance to the future of global agriculture. *Int. J. Agric. Sustainab.* 8, 219–236. doi: 10.3763/ijas.2010.0534

Reddy, A. S. B., and Juliet, D. S. (2019). "Transfer learning with resnet-50 for malaria cell-image classification," in *2019 International Conference on Communication and Signal Processing (ICCSP)* (Chennai: IEEE), 0945–0949.

Reynolds, M., and Langridge, P. (2016). Physiological breeding. *Curr. Opin. Plant Biol.* 31, 162–171. doi: 10.1016/j.pbi.2016.04.005

Rico-Fernández, M., Rios-Cabrera, R., Castelan, M., Guerrero-Reyes, H.-I., and Juarez-Maldonado, A. (2019). A contextualized approach for segmentation of foliage in different crop species. *Comput. Electron. Agric.* 156, 378–386. doi: 10.1016/j.compag.2018.11.033

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* (Springer), 234–241.

Roth, L., Aasen, H., Walter, A., and Liebisch, F. (2018). Extracting leaf area index using viewing geometry effects—a new perspective on high-resolution unmanned aerial system photography. *ISPRS J. Photogram. Remote Sens.* 141, 161–175. doi: 10.1016/j.isprsjprs.2018.04.012

Roth, L., Camenzind, M., Aasen, H., Kronenberg, L., Barendregt, C., Camp, K.-H., et al. (2020). Repeated multiview imaging for estimating seedling tiller counts of wheat genotypes using drones. *Plant Phenomics* 2020:3729715. doi: 10.34133/2020/3729715

Ruckelshausen, A., Biber, P., Dorna, M., Gremmes, H., Klose, R., Linz, A., et al. (2009). Bonirob-an autonomous field robot platform for individual plant phenotyping. *Precision Agric.* 9, 1.

Sadeghi-Tehran, P., Virlet, N., Sabermanesh, K., and Hawkesford, M. J. (2017). Multi-feature machine learning model for automatic segmentation of green fractional vegetation cover for high-throughput field phenotyping. *Plant Methods* 13, 103. doi: 10.1186/s13007-017-0253-8

Simmons, S. R. (1987). Growth, development, and physiology. *Wheat Wheat Improv*. 13, 77–113. doi: 10.2134/agronmonogr13.2ed.c3

Singh, V., and Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* 4, 41–49. doi: 10.1016/j.inpa.2016.10.005

Smith, A. G., Petersen, J., Selvan, R., and Rasmussen, C. R. (2020). Segmentation of roots in soil with u-net. *Plant Methods* 16, 1–15. doi: 10.1186/s13007-020-0563-0

Soekhoe, D., Van Der Putten, P., and Plaat, A. (2016). "On the impact of data set size in transfer learning using deep neural networks," in *International Symposium on Intelligent Data Analysis* (Springer).

Torres-Sánchez, J., López-Granados, F., and Pe na, J. M. (2015). An automatic object-based method for optimal thresholding in uav images: application for vegetation detection in herbaceous crops. *Comput. Electron. Agric.* 114, 43–52. doi: 10.1016/j.compag.2015.03.019

Ulmas, P., and Liiv, I. (2020). Segmentation of satellite imagery using u-net models for land cover classification. *arXiv preprint* arXiv:2003.02899.

Virlet, N., Sabermanesh, K., Sadeghi-Tehran, P., and Hawkesford, M. J. (2017). Field scanalyzer: an automated robotic field phenotyping platform for detailed crop monitoring. *Funct. Plant Biol.* 44, 143–153. doi: 10.1071/FP16163

Walter, A., Liebisch, F., and Hund, A. (2015). Plant phenotyping: from bean weighing to image analysis. *Plant Methods* 11, 1–11. doi: 10.1186/s13007-015-0056-8

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* doi: 10.1109/TPAMI.2020.2983686

Wang, M., Zhang, X., Niu, X., Wang, F., and Zhang, X. (2019). Scene classification of high-resolution remotely sensed image based on resnet. *J. Geovisualizat. Spatial Anal.* 3, 1–9. doi: 10.1007/s41651-019-0039-9

Wu, A., Zhu, J., and Ren, T. (2020). Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Comput. Electr. Eng.* 81:106454. doi: 10.1016/j.compeleceng.2019.106454

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 1492–1500.

Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv [Preprint]*. arXiv:1511.07 122v3.

Yu, K., Kirchgessner, N., Grieder, C., Walter, A., and Hund, A. (2017). An image analysis pipeline for automated classification of imaging light conditions and for quantification of wheat canopy cover time series in field phenotyping. *Plant Methods* 13, 1–13. doi: 10.1186/s13007-017-01 68-4

Zheng, L., Zhang, J., and Wang, Q. (2009). Mean-shift-based color segmentation of images containing green vegetation. *Comput. Electron. Agric.* 65, 93–98. doi: 10.1016/j.compag.2008.08.002

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: 11 January 2022
doi: 10.3389/fpls.2021.763479

Convolutional Neural Network Models Help Effectively Estimate Legume Coverage in Grass-Legume Mixed Swards

*Ryo Fujiwara[1], Hiroyuki Nashida[2], Midori Fukushima[2], Naoya Suzuki[2], Hiroko Sato[1], Yasuharu Sanada[1] and Yukio Akiyama[1]**

[1] Hokkaido Agricultural Research Center, NARO, Sapporo, Japan, [2] BANDAI NAMCO Research Inc., Tokyo, Japan

Evaluation of the legume proportion in grass-legume mixed swards is necessary for breeding and for cultivation research of forage. For objective and time-efficient estimation of legume proportion, convolutional neural network (CNN) models were trained by fine-tuning the GoogLeNet to estimate the coverage of timothy (TY), white clover (WC), and background (Bg) on the unmanned aerial vehicle-based images. The accuracies of the CNN models trained on different datasets were compared using the mean bias error and the mean average error. The models predicted the coverage with small errors when the plots in the training datasets were similar to the target plots in terms of coverage rate. The models that are trained on datasets of multiple plots had smaller errors than those trained on datasets of a single plot. The CNN models estimated the WC coverage more precisely than they did to the TY and the Bg coverages. The correlation coefficients (*r*) of the measured coverage for aerial images vs. estimated coverage were 0.92–0.96, whereas those of the scored coverage by a breeder vs. estimated coverage were 0.76–0.93. These results indicate that CNN models are helpful in effectively estimating the legume coverage.

Keywords: convolutional neural network models, legumes, grass-legume mixed swards, image analysis, unmanned aerial vehicle

OPEN ACCESS

Edited by:
*Bangyou Zheng,
Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia*

Reviewed by:
*Azlan Zahid,
Texas A&M University, United States
Pilar Hernandez,
Institute for Sustainable Agriculture, Spanish National Research Council (CSIC), Spain*

****Correspondence:***
*Yukio Akiyama
akky@affrc.go.jp*

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *24 August 2021*
Accepted: *10 December 2021*
Published: *11 January 2022*

Citation:
Fujiwara R, Nashida H, Fukushima M, Suzuki N, Sato H, Sanada Y and Akiyama Y (2022) Convolutional Neural Network Models Help Effectively Estimate Legume Coverage in Grass-Legume Mixed Swards. Front. Plant Sci. 12:763479. doi: 10.3389/fpls.2021.763479

INTRODUCTION

Grass-legume mixtures are applied in a forage production to obtain a greater productivity and a higher nutritive value of forage. Compared with the grass monocultures, pasture yields improve in grass-legume mixed swards owing to nitrogen fixation by legumes (Lüscher et al., 2014; Suter et al., 2015). In mixed swards, nitrogen fixed by forage legumes from the atmosphere is transferred to non-legumes (Pirhofer-Walzl et al., 2012; Thilakarathna et al., 2016). Furthermore, nitrogen fixed by legumes in mixed swards is higher than that in the legume monocultures (Nyfeler et al., 2011). Consequently, grass-legume mixtures improve the productivity of swards. Feeding the forage

Abbreviations: Bg, background; CNN, convolutional neural network; ExG, excess green; ExR, excess red; FCN, fully convolutional network; HSL, hue, saturation, and lightness; MAE, mean absolute error; MBE, mean bias error; MMAE, mean of MAE; MMBE, mean of MBE; OG, orchard grass; RC, red clover; TY, timothy; UAV, unmanned aerial vehicle; WC, white clover.

legumes to livestock can enhance the milk yields and the nutritional quality (Dewhurst et al., 2009; Peyraud et al., 2009). Therefore, the forage obtained from the grass-legume mixed swards can also be beneficial in terms of feed quality. In Japan, timothy (*Phleum pratense* L., TY) and white clover (*Trifolium repens* L., WC) are widely utilized for grass-legume mixed swards.

Legume proportion in mixed swards fluctuates dynamically over time, and patterns of the fluctuation vary depending on the proportion of seeds in the mixture, soil fertility, and climate conditions (Rasmussen et al., 2012; Suter et al., 2015; Bork et al., 2017). To maintain an appropriate legume proportion, it is crucial to obtain suitable forage varieties and to ensure proper management of grass-legume mixtures. Therefore, in breeding and in cultivation research, the evaluation of legume proportions is necessary. In Japan, for several times a year, the forage breeders score the coverage of grass and legume as an indicator of legume proportion. However, estimating the legume proportion in swards through observations of researchers may be subjective, and separating the legumes from the non-legumes by harvest measurements is time-consuming.

Unmanned aerial vehicles (UAVs) make it possible to obtain big data from images in a short time and conduct precise image analysis. The use of UAVs is becoming widespread in various fields, including agricultural analysis (Colomina and Molina, 2014). Analysis of UAV-based aerial images is also applied to remote sensing of sward height and of biomass in grasslands (Michez et al., 2019).

The image analysis method for objective and time-efficient estimation of legume proportions has been examined. Himstedt et al. (2012) applied color segmentation with legume-specific thresholds in hue saturation and light (HSL) color space to images of swards and predicted legume coverage and dry matter contribution. McRoberts et al. (2016) extracted local binary patterns (LBP), one of the texture descriptors in image classification, and developed regression models to estimate grass composition in alfalfa-grass fields. Mortensen et al. (2017) distinguished plant material from soil with excess green (ExG) and excess red (ExR) vegetation indices calculated from the RGB images, and detected the legume leaves with an edge detection and a reconstruction using flood filling.

In addition to image analysis methods using local color indices or feature extractors, convolutional neural networks (CNNs) are utilized in image classification or object detection. Convolutional neural network (CNNs) are a multi-layer neural networks equipped with convolutional and pooling layers, and they have a strong ability of complicated feature recognition (LeCun et al., 2015). There have been many studies on the application of CNNs in various aspects of agriculture (Kamilaris and Prenafeta-Boldú, 2018), including crop grain yield estimation (Yang et al., 2019), weed detection in grasslands (Yu et al., 2019a,b), and crop pest recognition (Thenmozhi and Srinivasulu Reddy, 2019; Li et al., 2020).

Some studies have applied CNNs to the estimation of legume proportion, especially in methods involving semantic segmentation. Semantic segmentation is a pixel-to-pixel classification task. The fully convolutional network (FCN) has been developed for solving the problem of segmentation (Shelhamer et al., 2017). Skovsen et al. (2017) trained an FCN architecture to distinguish clover, grass, and weed pixels. Larsen et al. (2018) examined the data collection workflow with UAVs and demonstrated the network (Larsen et al., 2018). Bateman et al. (2020) developed a new network for semantic segmentation, called the local context network, which distinguished clover, ryegrass, and the background more accurately than the FCN. Despite these studies, few examples of CNN application in the estimation of a legume proportion are available, and the knowledge required to develop the CNN models has not been fully accumulated. Besides, understanding how to develop models suitable to various fields and comparison between the models using different datasets may be useful.

GoogLeNet is a CNN model equipped with Inception modules and is the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 competition (Szegedy et al., 2015). Mehdipour Ghazi et al. (2017) demonstrated a plant identification with three CNN architectures, GoogLeNet, AlexNet, and VGGNet, using the dataset of LifeCLEF 2015. In the study, VGGNet was the most accurate, AlexNet was the fastest in terms of training, but GoogLeNet achieved competitive results both in terms of accuracy and of training speed. Because GoogLeNet has a well-balanced architecture, we considered it desirable to develop and compare multiple models.

In the current study, the CNN model estimating the coverage area of timothy, white clover, and the background (Bg) from UAV-based aerial images was trained by fine-tuning GoogLeNet. Multiple CNN models were trained on different datasets under the same conditions, and their accuracies were compared. To evaluate the usability of the CNN models, the correlations between the scored coverage by a breeder, measured coverage using aerial images, and estimated coverage by the CNN models were analyzed.

MATERIALS AND METHODS

Field Experiment and Data Collection

The field experiment and data collection were conducted at Hokkaido Agricultural Research Center (Hokkaido, Japan). Each of three white clover cultivars under a variety test ("cultivar A," "cultivar B," and "cultivar C") was mix-sowed with timothy on May 31, 2016. The plot size was 2 m × 3 m for each replicate (four replicates with three cultivars), and the amount of seeds sown was TY: 150 g/a and WC: 30 g/a in each plot. The plot design was determined using a randomized block design.

Coverage estimation, through scoring by a breeder and image acquisition with a UAV, was conducted 2 years after the seeding. A coverage score (%) for the three categories (TY, WC, and Bg) was assigned by a breeder on October 9, 2018 (scored coverage). The UAV-based aerial image of each plot was taken using DJI Phantom 4 Pro (SZ DJI Technology Co., Ltd., Shenzhen, China) on October 10, 2018, 14 days after the 3rd cutting of that year. The camera of Phantom 4 Pro had lens with an 8.8 mm focal length and a 1″ CMOS 20 M sensor. The UAV hovered above each plot at an altitude of 4 m and took one image. The image was stored as a Digital Negative (DNG), a format of RAW images. The ground

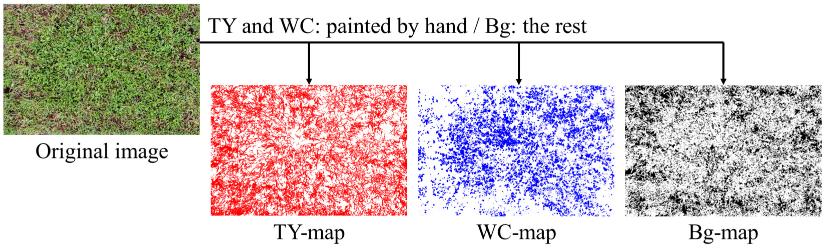


FIGURE 1 | Example of timothy (TY), white clover (WC), and background (Bg) maps generated from a UAV-based aerial images.

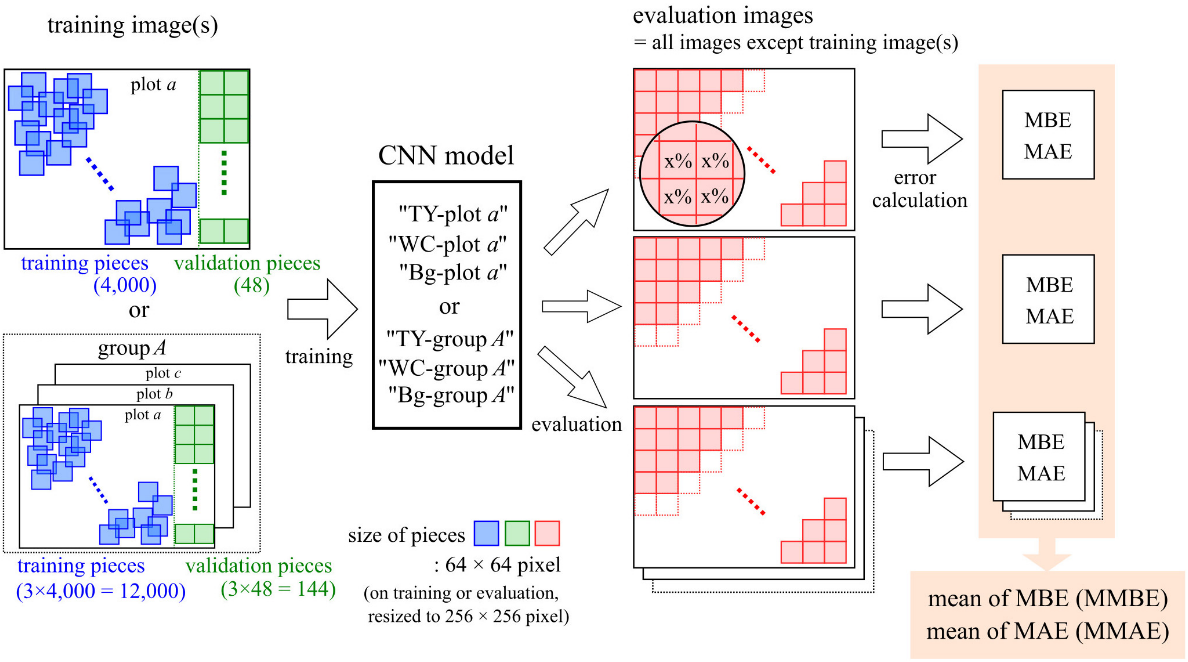


FIGURE 2 | The process of training and evaluation of the convolutional neural network (CNN) models.

sample distance was ∼1 mm/pixel. The images were imported to a personal computer and were adjusted with Photoshop CC (Adobe, San Jose, CA, United States). After auto-correction, the images were converted to PNG format. The images were cropped to the region of the plots and were keystone-corrected with the perspective crop tool. The size of the cropped images was approximately 2,000 × 3,000 pixel.

On each image of plots, blank layers for three categories (TY, WC, and Bg) were generated. Pixels belonging to TY and WC were painted on its respective layer with Photoshop CC using a pen display (Wacom Cintiq 16, Saitama, Japan) by hand. Pixels not belonging to TY or WC were painted as Bg category. Therefore, each layer acted as a map for that category (**Figure 1**). The layers were output as PNG files. The rates (%) of painted pixels on the maps were calculated with Python 3.6.8 (Python Software Foundation, 2018), Numpy 1.19.4 (Harris et al., 2020), and Pillow 8.0.1 (Clark, 2021). Thus, the percentage of the painted pixels represents the coverage rate of each category measured on the aerial image (measured coverage).

Training and Evaluation of the Convolutional Neural Network Models

The process of training and evaluation of the CNN models is shown in **Figure 2**. This process was conducted on a Windows 10 PC using a Core i9 7900X CPU, an RTX 2080 Ti GPU, and 64 GB RAM. The environment for CNN was constructed with Anaconda (Anaconda Software Distribution, 2021) using Python 3.6.2 (Python Software Foundation, 2017), CUDA 10.1 (NVIDIA Corporation, Santa Clara, CA, United States), cuDNN 7.5 (NVIDIA Corporation), Chainer 6.5.0 (Tokui et al., 2019), and cupy 6.5.0 (Okuta et al., 2017). Our previous research (Akiyama et al., 2020) was referenced in training the CNN models.

Formation of Training Datasets and Training of the Convolutional Neural Network Models

As the training dataset for a model, image pieces were cut from an aerial image of one plot or from aerial images of three plots

TABLE 1 | Classification of the rate of positive pixel in each region.

Class	Class value (%)	Class	Class value (%)	Class	Class value (%)
$0 \leq r_p \leq 0.025$	0	$0.325 < r_p < 0.375$	35	$0.675 \leq r_p \leq 0.725$	70
$0.025 < r_p < 0.075$	5	$0.375 \leq r_p \leq 0.425$	40	$0.725 < r_p < 0.775$	75
$0.075 \leq r_p \leq 0.125$	10	$0.425 < r_p < 0.475$	45	$0.775 \leq r_p \leq 0.825$	80
$0.125 < r_p < 0.175$	15	$0.475 \leq r_p \leq 0.525$	50	$0.825 < r_p < 0.875$	85
$0.175 \leq r_p \leq 0.225$	20	$0.525 < r_p < 0.575$	55	$0.875 \leq r_p \leq 0.925$	90
$0.225 < r_p < 0.275$	25	$0.575 \leq r_p \leq 0.625$	60	$0.925 < r_p < 0.975$	95
$0.275 \leq r_p \leq 0.325$	30	$0.625 < r_p < 0.675$	65	$0.975 \leq r_p \leq 1$	100

in a group. For training, 4,000 pieces of 64 × 64-pixel images were randomly cut from the region, excluding 128 pixels on the right side of the plot image. For validation, 48 pieces of 64 × 64-pixel size images were cut from 128 pixels on the right side, in order, from the upper left without overlaps. On the maps of TY, WC, and Bg, the rate of painted pixels (r_p) was calculated at the location of each piece. The r_p of each category was divided into 21 classes set every 5%, as shown in **Table 1** (the handling of values on the boundary is due to the behavior of the round function of Python). Sixty-four by sixty-four pixel-sized pieces were resized to 256 × 256 pixels by the nearest neighbor interpolation. These pieces and classes of TY, WC, and Bg coverage were used as the training dataset for a CNN model.

GoogLeNet, with the weights pre-trained on ImageNet, was trained on these datasets. The hyper-parameters were learning rate: 0.01, batch size: 32, optimizer: momentum Stochastic Gradient Descent (SGD; momentum = 0.9), and training epochs: 500. The accuracy (the rate of correct prediction on 21-classes classification) of each training model was checked with validation datasets every 1,000 iterations. The weight was saved during the validation. After training was completed, the weight with the highest accuracy upon validation was selected as the model for that dataset. The training mentioned above was conducted using datasets of 16 image sets (= 12 plots + 4 groups) across three categories (TY, WC, and Bg). A model trained on a dataset of a plot was named "(TY, WC, or Bg)-plot *a*" (*a* = plot code), and a model of a group was named "(TY, WC, or Bg)-group *A*" (*A* = group number). The properties of the models are shown in **Table 2**.

Evaluation of the Convolutional Neural Network Models

The trained model was evaluated using the evaluation images, which were the images not used in the training of each model. Images 64 × 64 pixels in size were cut from the evaluation images without overlaps (the remainder at the end of the image was not used) and resized to 256 × 256 pixels. One thousand two hundred to one thousand five hundred pieces of image were cut from each image. These pieces were applied to the CNN model to obtain the predicted class value of each piece. On the maps of TY, WC, and Bg, the class value on the location of each piece was measured in the same way as on the training datasets. Using the predicted class value and the measured class value, the mean bias error (MBE) and the mean absolute error (MAE) were calculated (Willmott, 1982; Willmott and Matsuura, 2005) as follows:

$$MBE = \frac{1}{n}\sum_{j=1}^{n}\left(P_j - O_j\right) \tag{1}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|P_j - O_j\right| \tag{2}$$

where n is the number of cases in the evaluation (pieces cut from an image), P_j is the predicted class value, and O_j is the observed class value.

The MBE indicates the bias of the model. Particularly, when the MBE is positive, the model tends to over-estimate; and when it is negative, the model tends to under-estimate. The MAE indicates the magnitude of the prediction error of the model.

One set of MBE and MAE values was obtained when one model was employed to predict pieces that were cut from an image of one plot (one-model-to-one-plot prediction). For the evaluation of the models, the means of MBEs (MMBE) and MAEs (MMAE) were calculated for each model using the following formulae:

$$MMBE = \frac{1}{N}\sum_{i=1}^{N} MBE_i \tag{3}$$

TABLE 2 | The properties of the models trained in this study.

Model's name (xx = TY, WC, or Bg)	Plot(s) for training	Number of pieces	
		Training	Validation
xx-plot 1-1	plot 1-1	4,000	48
xx-plot 1-2	plot 1-2	4,000	48
xx-plot 1-3	plot 1-3	4,000	48
xx-plot 1-4	plot 1-4	4,000	48
xx-plot 2-1	plot 2-1	4,000	48
xx-plot 2-2	plot 2-2	4,000	48
xx-plot 2-3	plot 2-3	4,000	48
xx-plot 2-4	plot 2-4	4,000	48
xx-plot 3-1	plot 3-1	4,000	48
xx-plot 3-2	plot 3-2	4,000	48
xx-plot 3-3	plot 3-3	4,000	48
xx-plot 3-4	plot 3-4	4,000	48
xx-group 1	plot 1-1, 2-1, 3-1	12,000	144
xx-group 2	plot 1-2, 2-2, 3-2	12,000	144
xx-group 3	plot 1-3, 2-3, 3-3	12,000	144
xx-group 4	plot 1-4, 2-4, 3-4	12,000	144

$$MMAE = \frac{1}{N}\sum_{i=1}^{N} MAE_i \quad (4)$$

where N is the number of images used for evaluating the model (all images except the ones used in the training), and MBE_i and MAE_i are the MBE and MAE of each one-model-to-one-plot prediction, respectively.

Estimation of the Measured and Scored Coverage

The estimation process is shown in **Figure 3**. In each one-model-to-one-plot prediction, the predicted values of the pieces were averaged. The average was regarded as the estimated coverage of the plot by the model. The model estimated the coverage of the plots from the dataset of a group, except of the ones used for the training. For the verification of the CNN models, the correlations between scored coverage, measured coverage, and estimated coverage by the model were analyzed.

In previous studies, the background has been distinguished from plant bodies using the excess green (ExG) and excess red (ExR) vegetation indices (Meyer and Neto, 2008; Mortensen et al., 2017). In our datasets of the 12 plots, the rate of pixels with zero or negative excess green minus, that of pixels with zero or negative excess red indices (ExG – ExR), was calculated as the estimated coverage of the background, as per the method of Meyer and Neto (2008). For comparison with the CNN method, the correlation of the measured coverage on aerial images vs. the estimated coverage with ExG – ExR was analyzed.

Evaluation of the Convolutional Neural Network Models for Predicting Legume Coverage Using Different Datasets by Grass or Legume Species

Datasets that are different to those used in training by grass or by legume species mix-sowed in the field were used to evaluate the accuracy of legume coverage prediction by the trained CNN models. On the date which the image was taken, the UAV used in aerial photographing, and the pasture species of grass, orchard grass (OG), and legume, WC or red clover (RC), are shown in **Table 3**. These images were taken over the fields in Hokkaido Agricultural Research Center (mentioned above). As shown in the table, DJI Phantom 4 RTK (SZ DJI Technology Co., Ltd., Shenzhen, China) was used for both OG-RC 3 and OG-RC 4, while Phantom 4 Pro was used for the others. The spec of the camera of Phantom 4 RTK is the same as that of the Phantom 4 Pro. The OG-RC 3 and OG-RC 4 were taken from the same plot on different dates, while the plots of other images were different to each other. The legume coverage maps of these images were generated (as shown in **Figure 1**). Images 64 × 64 pixels in size were cut from the generated images and predicted by the CNN model trained for each group. The coverage of RC was also predicted with the WC models. In the same way, MBEs and MAEs were calculated for the evaluation of the models.

RESULTS

Scored and Measured Coverage on Each Plot

The scored coverage by the breeder and the measured coverage on aerial images (measured using painted maps) are shown in **Table 4**. The sum of the measured coverage of the three categories (TY, WC, and Bg) on each plot was not precisely 100% because the maps of the categories were painted individually. The scored coverage tended to be higher in WC and lower in Bg, compared with the measured coverage. In every category, the range of the scored coverage was wider; that is, the breeder scored plots without much difference in the dynamically measured coverage. The correlation coefficient of the scored and measured coverages was high in WC but not in TY and Bg.

Evaluation and Comparison of the Convolutional Neural Network Models

The training time for the CNN models from one plot was approximately 4,000 s, and that from a group (three plots) was approximately 12,000 s. The MBEs for every one-model-to-one-plot prediction are shown in **Figure 4**, and the MAEs are shown in **Figure 5**. In these figures, the MBEs and the MAEs for predicting the images used in training each model are also shown in gray squares. The models trained on data from the plots, whose measured coverage rates were high (such as "TY-plot 1-4," "WC-plot 3-4," and "Bg-plot 2-1"; **Table 4**), tended to over-estimate; they had positive and high MBEs for predicting other plots. Contrary to this, the models trained on data from plots with low coverage rates ("TY-plot 2-1," "WC plot 2-1," and "Bg-plot 1-1") tended to under-estimate. The prediction errors (MAE) were high when these over or under-estimating models were used.

For prediction using the models trained on plots, whose measured coverage rates were close to the target (e.g., model: "TY-plot 2-1" and target: plot 2-4, and vice versa), the MBEs were close to zero, and the MAEs were low. The MAEs for predicting WC coverage of plot 3-4, the plot with high WC coverage, were high in many models but were lower in the model trained on data from another high-coverage plot (such as "WC-plot 1-1" and "WC-plot 2-3"). The Plot 2-2, which shared "cultivar B" but did not have high WC coverage, was predicted with high MAEs by "WC-plot 1-1," "WC-plot 2-3," and "WC-plot 3-4." Therefore, in this case, the main factor influencing the tendency of the model to predict with high MAEs was the WC coverage, not the cultivar.

The MMBE and the MMAE are shown in **Figure 6**. The calculation of MMBE and MMAE of each model did not include the MAEs and the MBEs for predicting the images used in the training. Therefore, the MMBE and the MMAE are the averages of each row without the gray squares in **Figures 4**, **5**. Overall, the MMAE was lower in WC than that in TY and Bg. Compared with the models trained on data from a plot, the MMAE of the models trained on data from a group was lower. Moreover, though the MMAEs of some models trained on data from a plot were extremely high, the MMAEs of the models trained on data from a group were relatively stable. This showed that the models trained on datasets representing multiple

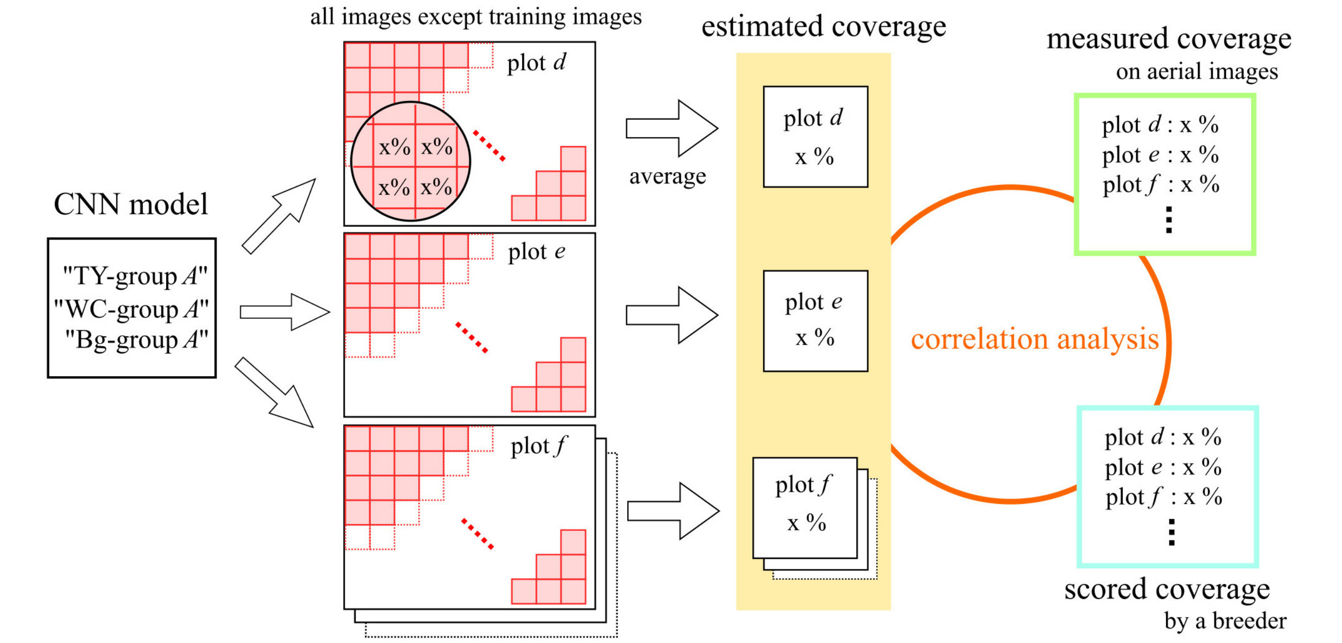


FIGURE 3 | The estimation process of measured and scored coverage by the CNN models.

conditions could predict wider target images accurately. When the MMAE of a model was high, such as in the case of "TY-plot 2-1," "WC-plot 3-4," and "Bg-plot 2-1," the absolute value of the MMBE was also high, that is, such a model tended to over or under-estimate.

Estimation of the Measured and Scored Coverage

For the models of TY, WC, and Bg trained on a dataset from each group, the scatter plots and the correlation coefficients (r) of scored coverage, measured coverage, and estimated coverage are shown in **Figure 7**. The results were different between models even in the scored vs. measured coverage pair because the plot data used to train the models were omitted in each pair. For WC, the correlation coefficients in every pair of scored, measured, and estimated coverage were high: r = 0.92–0.96 in measured vs. estimated coverage (the highest was "WC-group 2": r = 0.961), and r = 0.76–0.93 in scored vs. estimated coverage (the highest was "WC-group 1": r = 0.934). For TY and Bg, the correlation coefficients of measured vs. estimated coverage were lower, r = 0.24–0.75, in TY and r = 0.41–0.74 in Bg. In TY, the correlation coefficient of scored vs. estimated coverage exceeded that of measured vs. estimated coverage with every model.

The scatter plot of the estimated coverage of Bg with ExG – ExR and the measured coverage of Bg on aerial images of the 12 plots is shown in **Figure 8**. The correlation coefficient of the estimated coverage with ExG – ExR vs. measured coverage was 0.51, the same extent as with the CNN Bg- models (r = 0.41–0.74).

TABLE 3 | Status of the images used in evaluation of legume prediction on fields differing in grass or legume species.

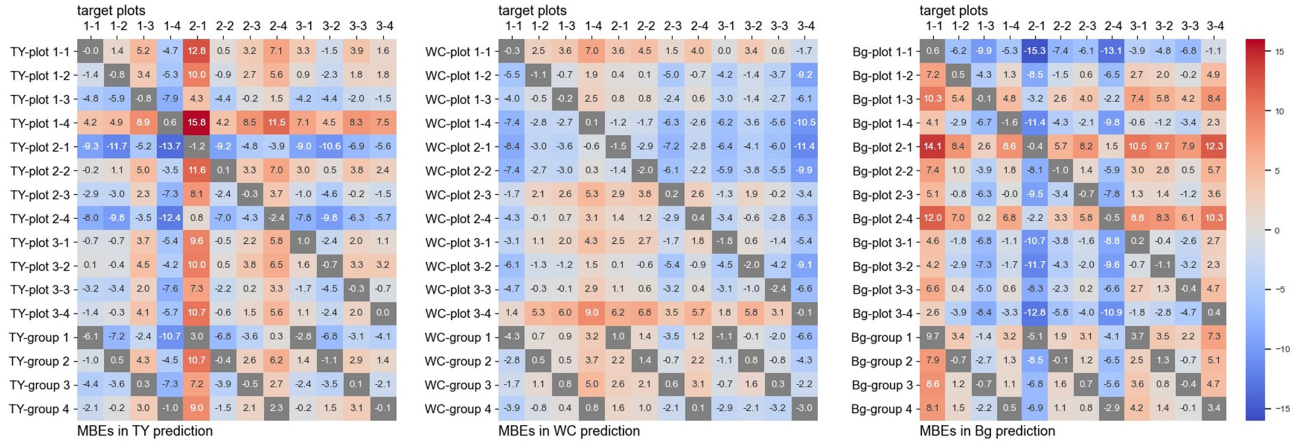
Image	Grass	Legume	Date taken	Lapsed days after cutting	UAV
OG-WC 1	OG	WC	2018/5/31	16 days	DJI Phantom 4 Pro
OG-WC 2	OG	WC	2018/5/31	16 days	DJI Phantom 4 Pro
OG-RC 1	OG	RC	2019/7/19	53 days	DJI Phantom 4 Pro
OG-RC 2	OG	RC	2019/7/19	53 days	DJI Phantom 4 Pro
OG-RC 3	OG	RC	2020/10/19	20 days	DJI Phantom 4 RTK
OG-RC 4	OG	RC	2020/10/26	27 days	DJI Phantom 4 RTK

Evaluation of Legume Coverage Prediction Using Different Datasets by Grass or Legume Species

Using the WC model trained on the dataset of each group, legume coverage on images of the OG-WC and OG-RC fields was predicted. The MBEs and MAEs for the prediction are shown in **Figure 9**. The coverage of the WC of OG-WC 1 and 2, taken on fields with a different grass species (OG), was predicted with MAEs lower than 10 by "WC-group 2," "WC-group 3," and "WC-group 4," though the MAEs increased by several points from those shown in **Figure 5**. When coverage of a different legume species (RC) was predicted by the WC-models, RC coverage of OG-RC 1 and 2 was predicted with relatively high MAEs and negative MBEs, that is, the models tended to under-estimate. In contrast, the RC coverage of OG-RC 3 and 4 was predicted with lower MAEs. Both the OG-RC 3 and OG-RC 4 differed from OG-RC 1 and 2 in season and year of the images being taken (**Table 3**). The difference between OG-RC 2 and OG-RC 4 in original images, prediction results by "WC-group 3," and details of the prediction are shown in **Figure 10**.

TABLE 4 | The scored coverage by a breeder and the measured coverage using aerial images.

			Scored coverage by a breeder (%)			Measured coverage using painted maps (%)		
Plot	**Group**	**Cultivar**	**TY**	**WC**	**Bg**	**TY**	**WC**	**Bg**
1-1	1	Cultivar B	50	50	0	41.4	27.8	31.5
1-2	2	Cultivar A	70	25	5	47.4	13.6	38.9
1-3	3	Cultivar C	45	35	20	38.3	16.8	44.8
1-4	4	Cultivar C	45	40	15	49.2	13.8	37.2
2-1	1	Cultivar C	70	25	5	37.7	13.2	46.5
2-2	2	Cultivar B	40	45	15	40.3	17.2	42.5
2-3	3	Cultivar B	45	50	5	37.6	26.8	35.6
2-4	4	Cultivar A	50	50	0	37.5	18.2	44.3
3-1	1	Cultivar A	70	30	0	42.2	22.0	35.8
3-2	2	Cultivar C	50	40	10	46.3	17.3	36.4
3-3	3	Cultivar A	55	45	0	41.2	20.9	37.8
3-4	4	Cultivar B	25	70	5	37.0	31.6	31.4
		Mean	51.3	42.1	6.7	41.3	19.9	38.6
		Range	45.0	45.0	20.0	12.2	18.3	15.1
Correlation coefficient (*r*) of scored vs. measured coverage						0.29	0.79	0.35



MBEs in TY prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
TY-plot 1-1	-0.0	1.4	5.2	-4.7	12.8	0.5	3.2	7.1	3.3	-1.5	3.9	1.6
TY-plot 1-2	-1.4	-0.8	3.4	-5.3	10.0	-0.9	2.7	5.6	0.9	-2.3	1.8	1.8
TY-plot 1-3	-4.8	-5.9	-0.8	-7.9	4.3	-4.4	-0.2	1.5	-4.2	-4.4	-2.0	-1.5
TY-plot 1-4	4.2	4.9	8.9	0.6	15.8	4.2	8.5	11.5	7.1	4.5	8.3	7.5
TY-plot 2-1	-9.3	-11.7	-5.2	-13.7	-1.2	-9.2	-4.8	-3.9	-9.0	-10.6	-6.9	-5.6
TY-plot 2-2	-0.2	1.1	5.0	-3.5	11.6	0.1	3.3	7.0	3.0	0.5	3.8	2.4
TY-plot 2-3	-2.9	-3.0	2.3	-7.3	8.1	-2.4	-0.3	3.7	-1.0	-4.6	-0.2	-1.5
TY-plot 2-4	-8.0	-9.8	-3.5	-12.4	0.8	-7.0	-4.3	-2.4	-7.8	-9.8	-6.3	-5.7
TY-plot 3-1	-0.7	-0.7	3.7	-5.4	9.6	-0.5	2.2	5.8	1.0	-2.4	2.0	1.1
TY-plot 3-2	0.1	-0.4	4.5	-4.2	10.0	0.5	3.8	6.5	1.6	-0.7	3.3	3.2
TY-plot 3-3	-3.2	-3.4	2.0	-7.6	7.3	-2.2	0.2	3.3	-1.7	-4.5	-0.3	-0.7
TY-plot 3-4	-1.4	-0.3	4.1	-5.7	10.7	-0.6	1.5	5.6	1.1	-2.4	2.0	0.0
TY-group 1	-6.1	-7.2	-2.4	-10.7	3.0	-6.8	-3.6	0.3	-2.8	-6.8	-3.1	-4.1
TY-group 2	-1.0	0.5	4.3	-4.5	10.7	-0.4	2.6	6.2	1.4	-1.1	2.9	1.4
TY-group 3	-4.4	-3.6	0.3	-7.3	7.2	-3.9	-0.5	2.7	-2.4	-3.5	0.1	-2.1
TY-group 4	-2.1	-0.2	3.0	-1.0	9.0	-1.5	2.1	2.3	-0.2	1.5	3.1	-0.1

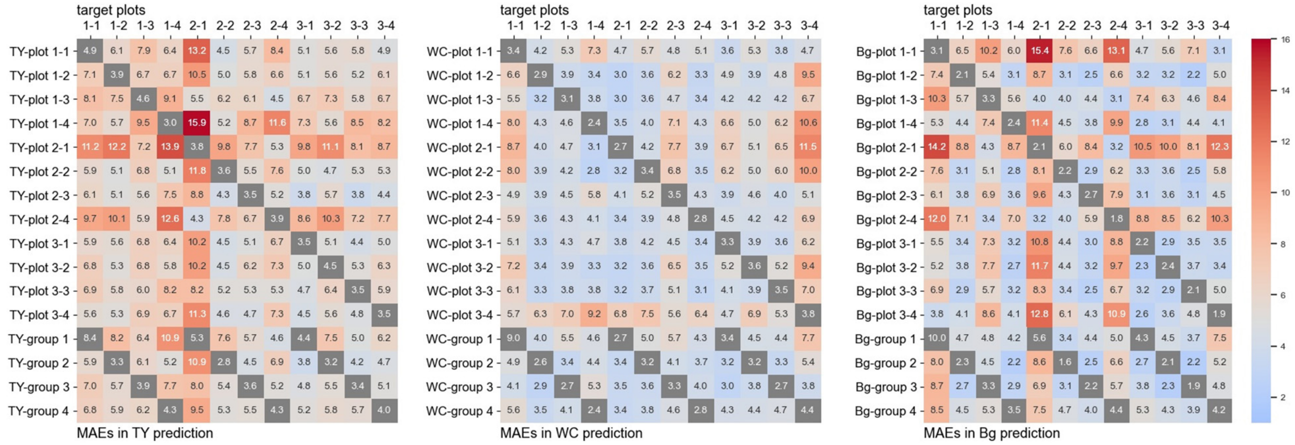
MBEs in WC prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
WC-plot 1-1	-0.3	2.5	3.6	7.0	3.6	4.5	1.5	4.0	0.0	3.4	0.6	-1.7
WC-plot 1-2	-5.5	-1.1	-0.7	1.9	0.4	0.1	-5.0	-0.7	-4.2	-1.4	-3.7	-9.2
WC-plot 1-3	-4.0	-0.5	-0.2	2.5	0.8	0.8	-2.4	0.6	-3.0	-1.4	-2.7	-6.1
WC-plot 1-4	-7.4	-2.8	-2.7	0.1	-1.2	-1.7	-6.3	-2.6	-6.2	-3.6	-5.6	-10.5
WC-plot 2-1	-8.4	-3.0	-3.6	-0.6	-1.5	-2.9	-7.2	-2.8	-6.4	-4.1	-6.0	-11.4
WC-plot 2-2	-7.4	-2.7	-3.0	0.3	-1.4	-2.0	-6.1	-2.2	-5.9	-3.8	-5.5	-9.9
WC-plot 2-3	-1.7	2.1	2.6	5.3	2.9	3.8	0.2	2.6	-1.3	1.9	-0.2	-3.4
WC-plot 2-4	-4.3	-0.1	0.7	3.1	1.4	1.2	-2.9	0.4	-3.4	-0.6	-2.8	-6.3
WC-plot 3-1	-3.1	1.1	2.0	4.3	2.5	2.7	-1.7	1.8	-1.8	0.6	-1.4	-5.4
WC-plot 3-2	-6.1	-1.3	-1.2	1.5	0.1	-0.6	-5.4	-0.9	-4.5	-2.0	-4.2	-9.1
WC-plot 3-3	-4.7	-0.3	-0.1	2.9	1.1	0.6	-3.2	0.4	-3.1	-1.0	-2.4	-6.6
WC-plot 3-4	1.4	5.3	6.0	9.0	6.2	6.8	3.5	5.7	1.8	5.8	3.1	-0.1
WC-group 1	-4.3	0.7	0.9	3.2	1.0	1.4	-3.5	1.2	-1.1	-0.1	-2.0	-6.6
WC-group 2	-2.8	0.5	0.5	3.7	2.2	1.4	-0.7	2.2	-1.1	0.8	-0.8	-4.3
WC-group 3	-1.7	1.1	0.8	5.0	2.6	2.1	0.6	3.1	-0.7	1.6	0.3	-2.2
WC-group 4	-3.9	-0.8	0.4	0.8	1.6	1.0	-2.1	0.1	-2.9	-2.1	-3.2	-3.0

MBEs in Bg prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
Bg-plot 1-1	0.6	-6.2	-9.9	-5.3	-15.3	-7.4	-6.1	-13.1	-3.9	-4.8	-6.8	-1.1
Bg-plot 1-2	7.2	0.5	-4.3	1.3	-8.5	-1.5	0.6	-6.5	2.7	2.0	-0.2	4.9
Bg-plot 1-3	10.3	5.4	-0.1	4.8	-3.2	2.6	4.0	-2.2	7.4	5.8	4.2	8.4
Bg-plot 1-4	4.1	-2.9	-6.7	-1.6	-11.4	-4.3	-2.1	-9.8	-0.6	-1.2	-3.4	2.3
Bg-plot 2-1	14.1	8.4	2.6	8.6	-0.4	5.7	8.2	1.5	10.5	9.7	7.9	12.3
Bg-plot 2-2	7.4	1.0	-3.9	1.8	-8.1	-1.0	1.4	-5.9	3.0	2.8	0.5	5.7
Bg-plot 2-3	5.1	-0.8	-6.3	-0.0	-9.5	-3.4	-0.7	-7.8	1.3	1.4	-1.2	3.6
Bg-plot 2-4	12.0	7.0	0.2	6.8	-2.2	3.3	5.8	-0.5	8.8	8.3	6.1	10.3
Bg-plot 3-1	4.6	-1.8	-6.8	-1.1	-10.7	-3.8	-1.6	-8.8	0.2	-0.4	-2.6	2.7
Bg-plot 3-2	4.2	-2.9	-7.3	-1.7	-11.7	-4.3	-2.0	-9.6	-0.7	-1.1	-3.2	2.3
Bg-plot 3-3	6.6	0.4	-5.0	0.6	-8.3	-2.3	0.2	-6.6	2.7	1.3	-0.4	4.7
Bg-plot 3-4	2.6	-3.9	-8.4	-3.3	-12.8	-5.8	-4.0	-10.9	-1.8	-2.8	-4.7	0.4
Bg-group 1	9.7	3.4	-1.4	3.2	-5.1	1.9	3.1	-4.1	3.7	3.5	2.2	7.3
Bg-group 2	7.9	-0.7	-2.7	1.3	-8.5	-0.1	1.2	-6.5	2.5	1.3	-0.7	5.1
Bg-group 3	8.6	1.2	-0.7	1.1	-6.8	1.6	0.7	-5.6	3.6	0.8	-0.4	4.7
Bg-group 4	8.1	1.5	-2.2	0.5	-6.9	1.1	0.8	-2.9	4.2	1.4	-0.1	3.4

FIGURE 4 | The mean bias errors (MBEs) for every one-model-to-one-plot prediction.



MAEs in TY prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
TY-plot 1-1	4.9	6.1	7.9	6.4	13.2	4.5	5.7	8.4	5.1	5.6	5.8	4.9
TY-plot 1-2	7.1	3.9	6.7	6.7	10.5	5.0	5.8	6.6	5.1	5.6	5.2	6.1
TY-plot 1-3	8.1	7.5	4.6	9.1	5.5	6.2	6.1	4.5	6.7	7.3	5.8	6.7
TY-plot 1-4	7.0	5.7	9.5	3.0	15.9	5.2	8.7	11.6	7.3	5.6	8.5	8.2
TY-plot 2-1	11.2	12.2	7.2	13.9	3.8	9.8	7.7	5.3	9.8	11.1	8.1	8.7
TY-plot 2-2	5.9	5.1	6.8	5.1	11.8	3.6	5.5	7.6	5.0	4.7	5.3	5.3
TY-plot 2-3	6.1	5.1	5.6	7.5	8.8	4.3	3.5	5.2	3.8	5.7	3.8	4.4
TY-plot 2-4	9.7	10.1	5.9	12.6	4.3	7.8	6.7	3.9	8.6	10.3	7.2	7.7
TY-plot 3-1	5.9	5.6	6.8	6.4	10.2	4.5	5.1	6.7	3.5	5.1	4.4	5.0
TY-plot 3-2	6.8	5.3	6.8	5.8	10.2	4.5	6.2	7.3	5.0	4.5	5.3	6.3
TY-plot 3-3	6.9	5.8	6.0	8.2	8.2	5.2	5.3	5.3	4.7	6.4	3.5	5.9
TY-plot 3-4	5.6	5.3	6.9	6.7	11.3	4.6	4.7	7.3	4.5	5.6	4.8	3.5
TY-group 1	8.4	8.2	6.4	10.9	5.3	7.6	5.7	4.6	4.4	7.5	5.0	6.2
TY-group 2	5.9	3.3	6.1	5.2	10.9	2.8	4.5	6.9	3.8	3.2	4.2	4.7
TY-group 3	7.0	5.7	3.9	7.7	8.0	5.4	3.6	5.2	4.8	5.5	3.4	5.1
TY-group 4	6.8	5.9	6.2	4.3	9.5	5.3	5.5	4.3	5.2	5.8	5.7	4.0

MAEs in WC prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
WC-plot 1-1	3.4	4.2	5.3	7.3	4.7	5.7	4.8	5.1	3.6	5.3	3.8	4.7
WC-plot 1-2	6.6	2.9	3.9	3.4	3.0	3.6	6.2	3.3	4.9	3.9	4.8	9.5
WC-plot 1-3	5.5	3.2	3.1	3.8	3.0	3.6	4.7	3.4	4.2	4.2	4.2	6.7
WC-plot 1-4	8.0	4.3	4.6	2.4	3.5	4.0	7.1	4.3	6.6	5.0	6.2	10.6
WC-plot 2-1	8.7	4.0	4.7	3.1	2.7	4.2	7.7	3.9	6.7	5.1	6.5	11.5
WC-plot 2-2	8.0	3.9	4.2	2.8	3.2	3.4	6.8	3.5	6.2	5.0	6.0	10.0
WC-plot 2-3	4.9	3.9	4.5	5.8	4.1	5.2	3.5	4.3	3.9	4.6	4.0	5.1
WC-plot 2-4	5.9	3.6	4.3	4.1	3.4	3.9	4.8	2.8	4.5	4.2	4.2	6.9
WC-plot 3-1	5.1	3.3	4.3	4.7	3.8	4.2	4.5	3.4	3.3	3.9	3.6	6.2
WC-plot 3-2	7.2	3.4	3.9	3.3	3.2	3.6	6.5	3.5	5.2	3.6	5.2	9.4
WC-plot 3-3	6.1	3.3	3.8	3.8	3.2	3.7	5.1	3.1	4.1	3.9	3.5	7.0
WC-plot 3-4	5.7	6.3	7.0	9.2	6.8	7.5	5.6	6.4	4.7	6.9	5.3	3.8
WC-group 1	9.0	4.0	5.5	4.6	2.7	5.0	5.7	4.3	3.4	4.5	4.4	7.7
WC-group 2	4.9	2.6	3.4	4.4	3.4	3.2	4.1	3.7	3.2	3.2	3.3	5.4
WC-group 3	4.1	2.9	2.7	5.3	3.5	3.6	3.3	4.0	3.0	3.8	2.7	3.8
WC-group 4	5.6	3.5	4.1	2.4	3.4	3.8	4.6	2.8	4.3	4.4	4.7	4.4

MAEs in Bg prediction

target plots	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3	3-4
Bg-plot 1-1	3.1	6.5	10.2	6.0	15.4	7.6	6.6	13.1	4.7	5.6	7.1	3.1
Bg-plot 1-2	7.4	2.1	5.4	3.1	8.7	3.1	2.5	6.6	3.2	3.2	2.2	5.0
Bg-plot 1-3	10.3	5.7	3.3	5.6	4.0	4.0	4.4	3.1	7.4	6.3	4.6	8.4
Bg-plot 1-4	5.3	4.4	7.4	2.4	11.4	4.5	3.8	9.9	2.8	3.1	4.4	4.1
Bg-plot 2-1	14.2	8.8	4.3	8.7	2.1	6.0	8.4	3.2	10.5	10.0	8.1	12.3
Bg-plot 2-2	7.6	3.1	5.1	2.8	8.1	2.2	2.9	6.2	3.3	3.6	2.5	5.8
Bg-plot 2-3	6.1	3.8	6.9	3.6	9.6	4.3	2.7	7.9	3.1	3.6	3.1	4.5
Bg-plot 2-4	12.0	7.1	3.4	7.0	3.2	4.0	5.9	1.8	8.8	8.5	6.2	10.3
Bg-plot 3-1	5.5	3.4	7.3	3.2	10.8	4.4	3.0	8.8	2.2	2.9	3.5	3.5
Bg-plot 3-2	5.2	3.8	7.7	2.7	11.7	4.4	3.2	9.7	2.3	2.4	3.7	3.4
Bg-plot 3-3	6.9	2.9	5.7	3.2	8.3	3.4	2.5	6.7	3.2	2.9	2.1	5.0
Bg-plot 3-4	3.8	4.1	8.6	4.1	12.8	6.1	4.3	10.9	2.6	3.6	4.8	1.9
Bg-group 1	10.0	4.7	4.8	4.2	5.6	3.4	4.4	5.0	4.3	4.5	3.7	7.5
Bg-group 2	8.0	2.3	4.5	2.2	8.6	1.6	2.5	6.6	2.7	2.1	2.2	5.2
Bg-group 3	8.7	2.7	3.3	2.9	6.9	3.1	2.2	5.7	3.8	2.3	1.9	4.8
Bg-group 4	8.5	4.5	5.3	3.5	7.5	4.7	4.0	4.4	5.3	4.3	3.9	4.2

FIGURE 5 | The mean average errors (MAEs) for every one-model-to-one-plot prediction.

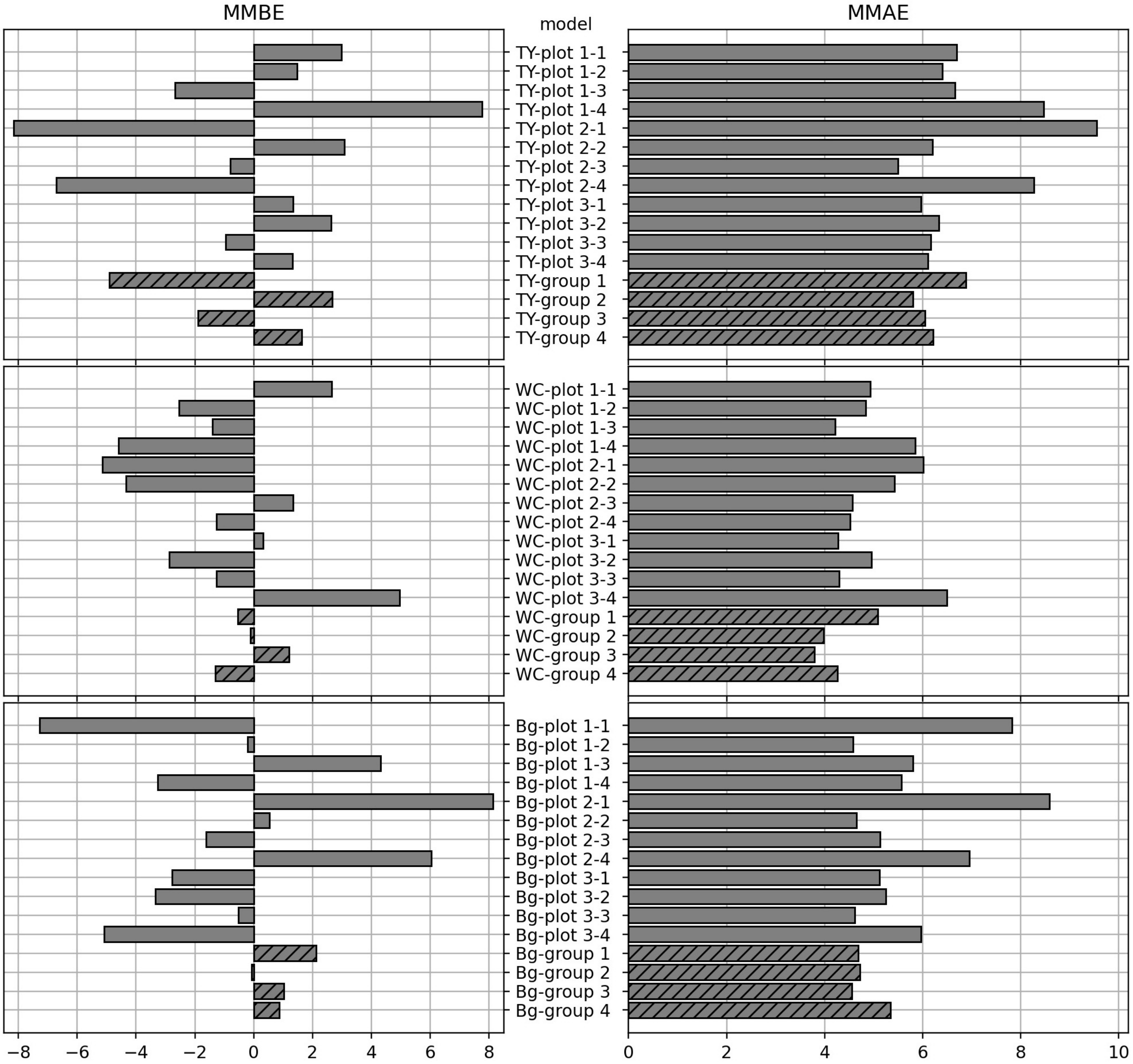


FIGURE 6 | The mean of mean bias errors (MMBE) and the mean of mean average errors (MMAE) for predicting the coverage of the plots by the CNN models. The calculation for each model doesn't include the MAEs and the MBEs on predicting the images used in the training.

DISCUSSION

The generalization of the CNN model is a major problem. The "WC-group 3" model used in this study was trained on the images of plot 2-3, a plot with high WC coverage, and other plots with a low coverage (**Table 4**). Consequently, the MAEs for predicting both high and low-coverage plots were suppressed (**Figure 5**), and the MMAE of the model was low (**Figure 6**). This model is likely to succeed in generalization. It is suggested that a wide distribution of coverage rate in training datasets leads to high accuracy of predicting different types of plots. However, "WC-group 1," trained on datasets including that from a high-coverage plot, plot 1-1, predicted other high-coverage plots with high MAEs. The reason of this may be the deficiency in fitting the model to the training datasets because this model also predicted plot 1-1, used in training the model, with a high MAE. A wide distribution of the coverage in training datasets, and a thorough training to fit the model to the datasets could be needed.

Judging by the correlation data shown in **Figure 7**, coverage estimation of legume by CNN models is likely to be easier than that of the grass or background. The reason for this may be the difference in the shape of leaves. Particularly, legume leaves are wider than those of grasses, thus, CNN can fully extract the features of legume leaves from the aerial images. Moreover, in

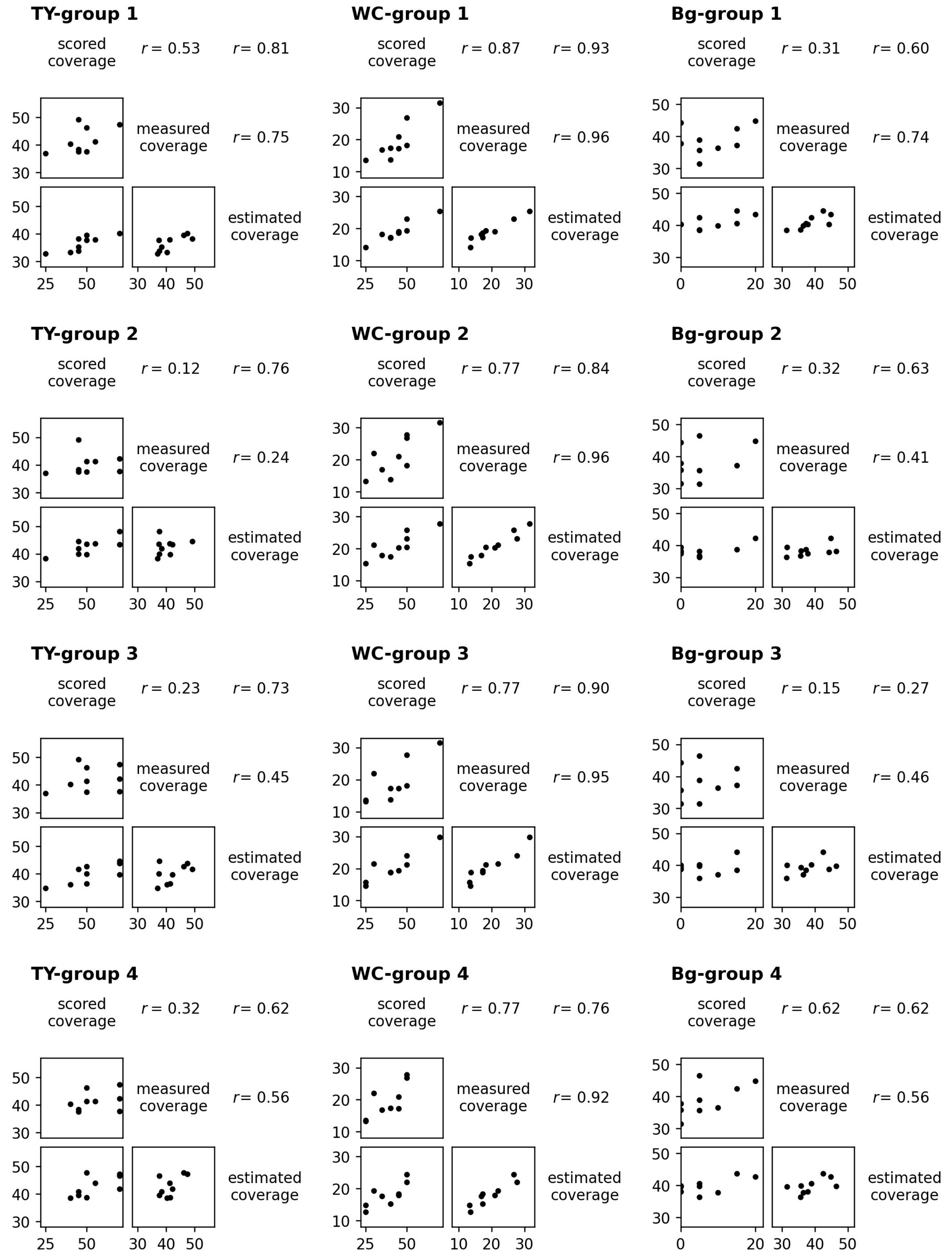


FIGURE 7 | The scatter plots and the correlation coefficients (r) of scored coverage by a breeder, measured coverage on aerial images, and estimated coverage by CNN models trained from groups.

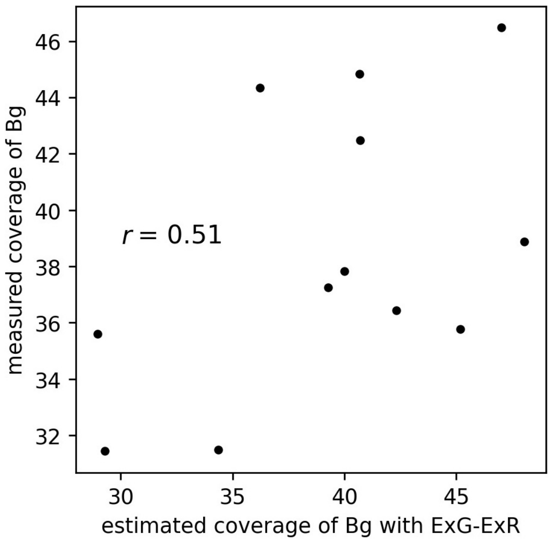


FIGURE 8 | The scatter plot of estimated coverage of Bg with ExG – ExR and measured coverage of Bg on aerial images of 12 plots.

this study, there were cases where distinguishing TY from the background was difficult on paintings of the location because there were withered TY leaves on the mixed swards in autumn. In such cases, the training datasets had some uncertainty. This may be one of the reasons why the coverage estimation of TY was inaccurate. The "TY-group 2" over-estimated the TY coverage of plot 2-1 (MBE: 10.7, MAE: 10.9), while the "Bg-group 2" under-estimated Bg coverage of plot 2-1 (MBE: −8.5, MAE: 8.6), as shown in **Figures 4**, **5**. The examples of the piece-level prediction are shown in **Figure 11**. In these examples, including withered TY leaves on sheets, TY class values were over-estimated and Bg class values were under-estimated. When maps of each category for training were painted on hand, the withered TY leaves were not painted as TY, and thus, painted as Bg. These withered TY (painted as Bg) areas are likely to be predicted as TY due to the shapes of the leaves. In this way, TY and Bg could be confused by the CNN models.

The background, the location with no plants present, lacks a characteristic shape. Feature extraction of the background by the CNN models may be difficult because the background does not have a unique shape. Using our datasets, the prediction of background coverage with ExG – ExR (Meyer and Neto, 2008) was not accurate (**Figure 8**). For the estimation of background coverage, other methods that involve vegetation indices or machine learning may be needed.

The comparison of the multiple models shown in **Figures 4**, **5** can be a variation of the cross validation with MBE and MAE, though the validation in our case was different to common cross validation in that the size of our validation datasets was larger than that of the training datasets. On the models generalized sufficiently, the prediction errors of validation datasets are near the prediction errors of training datasets in cross validation. From this point of view, the WC prediction models in our study were well-generalized, compared with those of TY and of Bg.

The scored coverage by a breeder reflects the 3D features that the aerial 2D images cannot grasp. Therefore, the scored coverage is not necessarily inferior to the measured coverage on images, though the scored coverage is subjective. It is likely that the CNN models can estimate both measured coverage and scored coverage for legumes based on the high correlations of predicting WC coverage observed (**Figure 7**). On the other hand, in TY and Bg, the correlations of scored vs. measured coverage were low. This may be due to the difference between the appearance of TY or Bg to a breeder and that from a UAV. It seems to be difficult to produce an estimation of a breeder by predicting TY or Bg coverage from images using CNN models.

However, in TY models, the correlations of scored vs. estimated coverage were higher than those of measured vs. estimated coverage (**Figure 7**). This means that the CNN models estimated the scored coverage more precisely even though the models were trained on measured coverage data. In general, the CNNs were likely to be trained on characteristic parts of images and made predictions using such parts, as demonstrated through visual explanation methods such as the Grad-CAM (Selvaraju et al., 2020). For the prediction of TY coverage, the CNN models may be trained mainly on data of the characteristic parts (= typical parts for TY) and may predict a high coverage using the plot images of such parts. Breeders also look at characteristic parts in plots and score the coverage. This may be the reason why the correlations of the scored coverage by a breeder vs. the estimated coverage by the CNN models were higher. These results suggest that the CNN models make predictions using the data generated through human decision-making more precisely than using data measured mechanically. Additional research is needed to confirm this.

When the WC coverage from the OG-WC images was predicted by the CNN models trained with TY-WC images, an

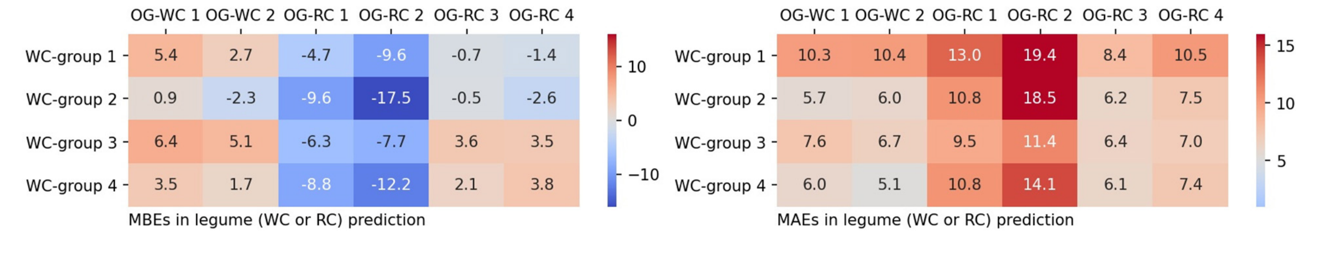


FIGURE 9 | The MBEs and the MAEs for predicting legume coverage of different images by grass or legume species to those used in training.

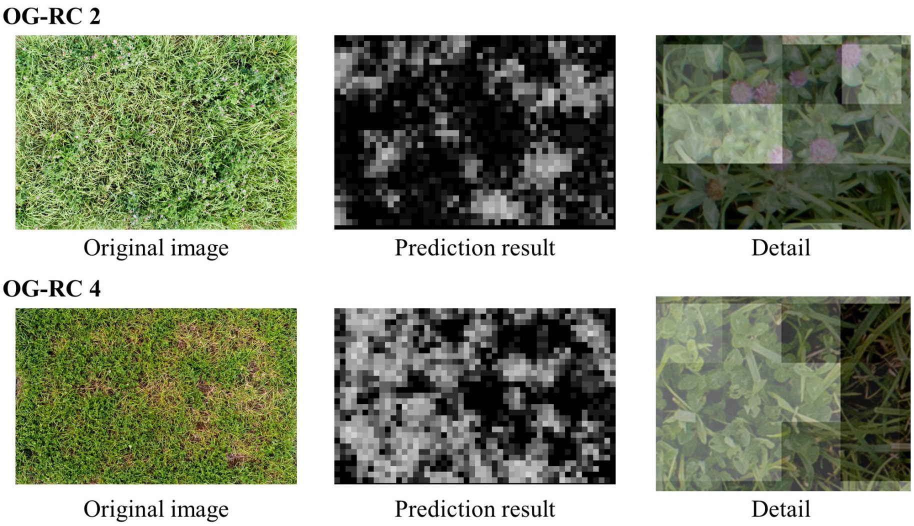


FIGURE 10 | The original plot images and the prediction results of OG-RC 2 and OG-RC 4. "Prediction result" is the result map of the legume (RC) coverage prediction by "WC-group 3" illustrated in grayscale (when a sheet is close to white, the predicted class value is high). "Detail" is an enlarged view of the original image on which the prediction result map overlapped (the opacity of the result map is adjusted in overlapping).

increase in MAEs was limited for "WC-group 2," "WC-group 3," and "WC-group 4" (**Figure 9**). It appears that the WC-models trained with TY-WC images are applicable to WC coverage prediction of mixed swards with a different grass species. On the other hand, when the RC coverage from OG-RC images was predicted by the WC-models, the MAEs increased on OG-RC 1 and 2 (**Figure 9**). In these images, there were pieces with RC presence that were predicted to have low legume coverage, as shown in "Detail" of OG-RC 2 in **Figure 10**. In OG-RC 1 and 2, the RC leaves stood upwardly, and thus, looked sharper. Such RC leaves had different shapes on imaging to WC leaves. In contrast, in OG-RC 3 and 4, RC leaves looked similar to WC leaves. This may be the reason why the WC-models predicted the RC coverage of OG-RC 1 and 2 with higher MAEs, and that of OG-RC 3 and 4 with lower MAEs. For training the model to predict RC coverage accurately, training datasets, which cover leaf shapes of various RC conditions, should be needed.

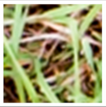
Piece	Observed class value	Predicted class value
	TY: 30	TY: 60
	Bg: 65	Bg: 40
	TY: 40	TY: 65
	Bg: 55	Bg: 35
	TY: 25	TY: 50
	Bg: 55	Bg: 35

FIGURE 11 | The examples of the piece-level prediction in which TY and Bg were confused. The pieces were cut from plot 2-1. The coverage of TY was predicted by "TY-group 2" and that of Bg was predicted by "Bg-group 2."

In this study, for comparing multiple models using different training datasets in the same conditions, adjustment of the architectures and hyperparameters of the CNN was not conducted. Adequate accuracy for coverage estimation of WC was achieved in this condition. The following points can be considered for further improvement of the models: (1) The architecture of the CNN: Yu et al. (2019a) reported that AlexNet and VGGNet achieved higher precision values for weed detection in perennial ryegrass than GoogLeNet. The CNN models for the coverage estimation of mixed swards can be improved with architectures other than GoogLeNet. (2) The optimizer used for training the CNN model: Momentum SGD was used as the optimizer in our study, but other optimizers, such as AdaGrad (Duchi et al., 2011) and Adam (Kingma and Ba, 2014), can be used. Adjustment of hyperparameters, including optimizers, may improve the coverage estimation models of grass-legume mixed swards. (3) The problem setting: In the predictions in this study, a 21-class classification was applied to the CNN models because GoogLeNet has been developed to address the issue of classification. The CNN models for regression problems, however, are also buildable. There are precedents for this in crop yield prediction (Nevavuori et al., 2019) in and maize tassels counting (Lu et al., 2017). The development of CNN regression models that predict coverage as a continuous value may be promising.

In previous studies, methods involving semantic segmentation have mainly been applied to the prediction of a legume proportion using CNNs (Skovsen et al., 2017; Larsen et al., 2018; Bateman et al., 2020). On the other hand, in this study, class values of coverage in separate regions were predicted. Using this method, many pieces of images for training can be obtained from a fixed number of aerial images. Moreover, prediction errors may be suppressed because values of coverage are predicted directly, and not by interposing the classification on each pixel. So far, the superiorities of these methods are not clear. Additionally, although the measured coverage on aerial images and the scored coverage by a breeder were used as indicators of legume proportion in this study, yield-based indicators such as dry matter yield are also likely to be useful. Comparative studies between the prediction methods of legume proportion are required.

The CNN system to investigate a small experimental field was developed in this study because of the difficulty to take high resolution images for a large field. However, the investigation system for the large production field is important. The capability to capture a large field mainly depends on the performance of UAVs; examples are flight time, the camera sensor size, and the camera lens. As the technology of UAVs becomes more advanced, this CNN system may be useful for the large production field in the future.

Multiple CNN models estimating the coverage of timothy (TY), white clover (WC), and the background (Bg) from UAV-based aerial images were trained and were compared. The accuracy of the CNN models used in our study was affected by the coverage on the plots in the training datasets, and thus, it was suggested that a wide distribution of the coverage rate in the training datasets was important for the generalization of the model. The WC coverage, both the measured coverage on aerial images and the scored coverage by a breeder, was precisely estimated by the CNN models.

The CNN model trained on data from a group of the three plots was shown to be useful for the estimation of the WC coverage. It is expected that further works based on the methods in this study will generate a practical system to estimate the coverage in grass-legume mixed swards.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

RF analyzed the results and wrote the manuscript. HN, MF, and NS conceived the idea and proposed the method. HS and YS performed the experiments. YA designed the experiments and analyzed the results. All authors contributed to the article and approved the submitted version.

FUNDING

This work was partly supported by the JSPS KAKENHI Grant Number JP 21K05551.

ACKNOWLEDGMENTS

We are grateful for the technical assistance of Satomi Shimada, Kana Takeda, and Yusuke Ozaki.

REFERENCES

Akiyama, Y., Nashida, H., Suzuki, N., and Sanada, Y. (2020). Development of a new evaluation method for individual selection in breeding of *Dactylis glomerata* L. with unmanned aerial vehicle (UAV) and deep learning. *Breed. Res.* 22, 21–27. doi: 10.1270/jsbbr.19J07

Anaconda Software Distribution (2021). *Anaconda Documentation*. Available online at: https://docs.anaconda.com/ (accessed November 18, 2021).

Bateman, C. J., Fourie, J., Hsiao, J., Irie, K., Heslop, A., Hilditch, A., et al. (2020). Assessment of mixed sward using context sensitive convolutional neural networks. *Front. Plant Sci.* 11:159. doi: 10.3389/fpls.2020.00159

Bork, E. W., Gabruck, D. T., McLeod, E. M., and Hall, L. M. (2017). Five-year forage dynamics arising from four legume-grass seed mixes. *Agron. J.* 109, 2789–2799. doi: 10.2134/agronj2017.02.0069

Clark, A. (2021). *Pillow (PIL Fork) Documentation*. Available online at: https://pillow.readthedocs.io/_/downloads/en/stable/pdf/ (accessed November 18, 2021).

Colomina, I., and Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: a review. *ISPRS J. Photogramm.* 92, 79–97. doi: 10.1016/j.isprsjprs.2014.02.013

Dewhurst, R. J., Delaby, L., Moloney, A., Boland, T., and Lewis, E. (2009). Nutritive value of forage legumes used for grazing and silage. *Ir. J. Agric. Food Res.* 48, 167–187.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12, 2121–2159.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. *Nature* 585, 357–362.

Himstedt, M., Fricke, T., and Wachendorf, M. (2012). The benefit of color information in digital image analysis for the estimation of legume contribution in legume-grass mixtures. *Crop Sci.* 52, 943–950. doi: 10.2135/cropsci2011.04.0189

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: a survey. *Comput. Electron. Agric.* 147, 70–90. doi: 10.1016/j.compag.2018.02.016

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv* [Preprint]. arXiv:1412.6980,

Larsen, D., Skovsen, S., Steen, K. A., Grooters, K., Eriksen, J., Green, O., et al. (2018). "Autonomous mapping of grass–clover ratio based on unmanned aerial vehicles and convolutional neural networks," in *Proceedings of the 14th International Conference on Precision Agriculture* Montréal, QC.

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

Li, Y. F., Wang, H. X., Dang, L. M., Sadeghi-Niaraki, A., and Moon, H. (2020). Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* 169:10. doi: 10.1016/j.compag.2019.105174

Lu, H., Cao, Z. G., Xiao, Y., Zhuang, B. H., and Shen, C. H. (2017). TasselNet: counting maize tassels in the wild via local counts regression network. *Plant Methods* 13:79. doi: 10.1186/s13007-017-0224-0

Lüscher, A., Mueller-Harvey, I., Soussana, J. F., Rees, R. M., and Peyraud, J. L. (2014). Potential of legume-based grassland-livestock systems in Europe: a review. *Grass Forage Sci.* 69, 206–228. doi: 10.1111/gfs.12124

McRoberts, K. C., Benson, B. M., Mudrak, E. L., Parsons, D., and Cherney, D. J. R. (2016). Application of local binary patterns in digital images to estimate botanical composition in mixed alfalfa-grass fields. *Comput. Electron. Agric.* 123, 95–103. doi: 10.1016/j.compag.2016.02.015

Mehdipour Ghazi, M. M., Yanikoglu, B., and Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing* 235, 228–235. doi: 10.1016/j.neucom.2017.01.018

Meyer, G. E., and Neto, J. C. (2008). Verification of color vegetation indices for automated crop imaging applications. *Comput. Electron. Agric.* 63, 282–293. doi: 10.1016/j.compag.2008.03.009

Michez, A., Lejeune, P., Bauwens, S., Herinaina, A. A. L., Blaise, Y., Castro Muñoz, E. C., et al. (2019). Mapping and monitoring of biomass and grazing in pasture with an unmanned aerial system. *Remote Sens.* 11:473. doi: 10.3390/rs11050473

Mortensen, A. K., Karstoft, H., Søegaard, K., Gislum, R., and Jørgensen, R. N. (2017). Preliminary results of clover and grass coverage and total dry matter estimation in clover-grass crops using image analysis. *J. Imaging* 3:59. doi: 10.3390/jimaging3040059

Nevavuori, P., Narra, N., and Lipping, T. (2019). Crop yield prediction with deep convolutional neural networks. *Comput. Electron. Agric.* 163:9. doi: 10.1016/j.compag.2019.104859

Nyfeler, D., Huguenin-Elie, O., Suter, M., Frossard, E., and Lüscher, A. (2011). Grass-legume mixtures can yield more nitrogen than legume pure stands due to mutual stimulation of nitrogen uptake from symbiotic and non-symbiotic sources. *Agric. Ecosyst. Environ.* 140, 155–163. doi: 10.1016/j.agee.2010.11.022

Okuta, R., Unno, Y., Nishino, D., Hido, S., and Loomis, C. (2017). "CuPy: a NumPy-compatible library for NVIDIA GPU calculations," in *Proceedings of the 31st Confernce on Neural Information Processing Systems* Tokyo.

Peyraud, J. L., Le Gall, A., and Luscher, A. (2009). Potential food production from forage legume-based-systems in Europe: an overview. *Ir. J. Agric. Food Res.* 48, 115–135.

Pirhofer-Walzl, K., Rasmussen, J., Høgh-Jensen, H., Eriksen, J., Søegaard, K., and Rasmussen, J. (2012). Nitrogen transfer from forage legumes to nine neighbouring plants in a multi-species grassland. *Plant Soil* 350, 71–84. doi: 10.1007/s11104-011-0882-z

Python Software Foundation (2017). *Python Release Python 3.6.2 | Python.org*. Available online at: https://www.python.org/downloads/release/python-362/ (Accessed November 18, 2021).

Python Software Foundation (2018). *Python Release Python 3.6.8 | Python.org*. Available online at: https://www.python.org/downloads/release/python-368/ (Accessed November 18, 2021).

Rasmussen, J., Søegaard, K., Pirhofer-Walzl, K., and Eriksen, J. (2012). N2-fixation and residual N effect of four legume species and four companion grass species. *Eur. J. Agron.* 36, 66–74. doi: 10.1016/j.eja.2011.09.003

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* 128, 336–359. doi: 10.1007/s11263-019-01228-7

Shelhamer, E., Long, J., and Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 640–651. doi: 10.1109/TPAMI.2016.2572683

Skovsen, S., Dyrmann, M., Mortensen, A. K., Steen, K. A., Green, O., Eriksen, J., et al. (2017). Estimation of the botanical composition of clover-grass leys from RGB images using data simulation and fully convolutional neural networks. *Sensors (Basel)* 17:18. doi: 10.3390/s17122930

Suter, M., Connolly, J., Finn, J. A., Loges, R., Kirwan, L., Sebastià, M. T., et al. (2015). Nitrogen yield advantage from grass-legume mixtures is robust over a wide range of legume proportions and environmental conditions. *Glob. Chang. Biol.* 21, 2424–2438. doi: 10.1111/gcb.12880

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* Boston, MA. doi: 10.1109/CVPR.2015.7298594

Thenmozhi, K., and Srinivasulu Reddy, U. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* 164:11. doi: 10.1016/j.compag.2019.104906

Thilakarathna, M. S., McElroy, M. S., Chapagain, T., Papadopoulos, Y. A., and Raizada, M. N. (2016). Belowground nitrogen transfer from legumes to non-legumes under managed herbaceous cropping systems. A review. *Agron. Sustain. Dev* 36:58. doi: 10.1007/s13593-016-0396-4

Tokui, S., Okuta, R., Akiba, T., Niitani, Y., Ogawa, T., Saito, S., et al. (2019). "Chainer: a deep learning framework for accelerating the research cycle," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* Tokyo, 2002–2011. doi: 10.1145/3292500.3330756

Willmott, C. J. (1982). Some comments on the evaluation of model performance. *Bull. Amer. Meteor. Soc.* 63, 1309–1313. doi: 10.1175/1520-04771982063<1309:SCOTEO<2.0.CO;2

Willmott, C. J., and Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* 30, 79–82. doi: 10.3354/cr030079

Yang, Q., Shi, L. S., Han, J. Y., Zha, Y. Y., and Zhu, P. H. (2019). Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images. *Field Crops Res.* 235, 142–153. doi: 10.1016/j.fcr.2019.02.022

Yu, J. L., Schumann, A. W., Cao, Z., Sharpe, S. M., and Boyd, N. S. (2019a). Weed detection in perennial ryegrass with deep learning convolutional neural network. *Front. Plant Sci.* 10:1422. doi: 10.3389/fpls.2019.01422

Yu, J. L., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019b). Deep learning for image-based weed detection in turfgrass. *Eur. J. Agron.* 104, 78–84. doi: 10.1016/j.eja.2019.01.004

Conflict of Interest: HN, MF, and NS were employed by company BANDAI NAMCO Research Inc.

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.

ORIGINAL RESEARCH
published: 23 February 2022
doi: 10.3389/fpls.2022.730190

Corn Seed Defect Detection Based on Watershed Algorithm and Two-Pathway Convolutional Neural Networks

Linbai Wang[1,2†], Jingyan Liu[1,2†], Jun Zhang[1,2], Jing Wang[1,2] and Xiaofei Fan[1,2*]

[1] State Key Laboratory of North China Crop Improvement and Regulation, Hebei Agricultural University, Baoding, China,
[2] College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding, China

Corn seed materials of different quality were imaged, and a method for defect detection was developed based on a watershed algorithm combined with a two-pathway convolutional neural network (CNN) model. In this study, RGB and near-infrared (NIR) images were acquired with a multispectral camera to train the model, which was proved to be effective in identifying defective seeds and defect-free seeds, with an averaged accuracy of 95.63%, an averaged recall rate of 95.29%, and an F1 (harmonic average evaluation) of 95.46%. Our proposed method was superior to the traditional method that employs a one-pathway CNN with 3-channel RGB images. At the same time, the influence of different parameter settings on the model training was studied. Finally, the application of the object detection method in corn seed defect detection, which may provide an effective tool for high-throughput quality control of corn seeds, was discussed.

Keywords: corn seed defect, multispectral image, object detection, watershed segmentation algorithm, convolutional neural network

OPEN ACCESS

Edited by:
Lisbeth Garbrecht Thygesen,
University of Copenhagen, Denmark

Reviewed by:
Abhimanyu Singh Garhwal,
AgResearch Ltd., New Zealand
Lin Qi,
Ocean University of China, China

***Correspondence:**
Xiaofei Fan
leopardfxf@163.com

†These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *24 June 2021*
Accepted: *25 January 2022*
Published: *23 February 2022*

Citation:
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. 13:730190. doi: 10.3389/fpls.2022.730190

INTRODUCTION

Corn is one of the most important crops in the world (Afzal et al., 2017), which is widely planted around the Earth. Its output and trade volume have kept increasing in recent years. In the process of circulation, appearance quality is a critical factor that influences corn seed price. Corn seeds are vulnerable to damage and mildew during storage and transportation, and phenotypic defect is an important index of seed quality evaluation. At present, seed quality detection still relies on the method of traditional manual identification, which employs low efficiency and strong subjectivity. With the development of computer vision technology (Rehman et al., 2018; Gutiérrez et al., 2019; Keiichi et al., 2019; Azimi et al., 2020; Arunachalam and Andreasson, 2021), image processing methods based on machine learning are applied to seed quality classification and have achieved good results. Kiratiratanapruk and Sinthupinyo (2012) proposed a method to classify more than 10 levels of seed quality by using color and texture features with a support vector machine (SVM) classifier. Ke-Ling et al. (2018) proposed a method of high-quality pepper seed screening based on machine vision, which could be used to predict the germination rate of seeds effectively, and therefore provided a guide for seed quality selection. Ali et al. (2020) discussed the feasibility of the machine learning method in corn seed classification. While the traditional machine learning methods normally require extracting the features manually, which are usually not comprehensive enough, the recognition accuracy, therefore, is limited.

In recent years, as a representative of deep learning technology, convolutional neural networks (CNNs) develop rapidly and are widely used for image recognition (Afonso et al., 2019; Altuntaş et al., 2019; Gao et al., 2020; Zhang C. et al., 2020). Compared with traditional machine learning technology, CNNs are naturally embedded with a feature learning part through the combination of low-level features to form more abstract high-level features. Many researchers have applied CNNs to the field of agriculture. Laabassi et al. (2021) proposed a CNN model to classify wheat varieties, and the accuracy of classification was between 85.00 and 95.68%. Pang et al. (2020) developed a method for rapid estimation and prediction of corn seed vigor using a hyperspectral imaging system with deep learning. The recognition accuracy of the 1D-CNN model reached 90.11%, and the recognition accuracy of the 2D-CNN model reached 99.96%. Sj et al. (2021) proposed a method to extract the characteristics of corn seeds by using a deep CNN and then classifying the varieties. The results showed that CNNs were effective in corn seed classification.

In this article, RGB and NIR images (Kusumaningrum et al., 2018) collected by a multispectral camera were used to train a CNN model. To solve the problem of corn seed adhesion and seed location during the recognition process, a watershed algorithm (Lei et al., 2019; Sta et al., 2019; Zhang et al., 2021) combined with a two-way CNN (Zhang J. J. et al., 2020) was proposed to detect corn seed defects. The results revealed that this method is with high accuracy, and the targets can be accurately located and classified. This method may provide a theoretical basis for the subsequent development of a seed quality control device.

MATERIALS AND METHODS

Experimental Material and Instruments

In this experiment, 2,365 corn seeds from three different varieties (Zhengdan 985, Keshi 982, Jiyu 517) were adopted as experimental materials. Some seeds were defect-free in appearance, and the other seeds were with defects, including mold, insect or mechanical damages, and discoloration. A 4-channel (RGB + NIR) multispectral camera (LQ-200CL, JAI, Denmark) was used for image acquisition, with 8 bits for each channel and a resolution of 1,296 * 964. A white LED ring light, coupled with a near-infrared ring light, and a white backlight panel were used to enhance the image contrast. The image acquisition platform is shown in **Figure 1**. At the same time, to prevent the seeds from overlapping, the vibration module was placed under the backlight panel (shown in **Figure 2**). The motor voltage is 12 V. The rotational speed of the motor is 8,000 rpm. The size of the vibrating head is 3.5 cm. It is found that the vibration module shows a very good effect in restraining seed overlap and shielding.

The experiment was based on Windows 10, a 64-bit operating system with CUDA 10.0, and python programming language, along with TensorFlow and Keras deep learning framework. The computer used for the experiment employed a GeForce GTX 1660 graphics card, with 6G memory, and an Intel (R) Core (TM) i5-9400f processor with the main frequency of 2.90 GHz.

Data Acquisition

A total of 50 samples of corn seed with no defects (1,066 single seeds overall) and fifty seed samples with different appearance defects (1,042 single seeds overall) were imaged. The images of another 10 samples with both defective and defect-free seeds were also acquired for the verification of the final model, with an overall 257 single seeds. Each sample was captured in one image deck, which contained RGB and NIR images, with a size of 1,296 × 964. The images acquired are shown in **Figure 3**. To solve the issue of adhesion among seeds in the images, a watershed algorithm was applied to each image, and all individual seeds were segmented. Eventually, each seed was extracted from the original image to form a new image, which was resized to 224 × 224 with bilinear interpolation.

To improve the performance of the model, data augmentation was implemented for image decks of individual seeds. The enhancement methods (Huang et al., 2019; Tiwari et al., 2021) included brightness adjustment, rotation, applying Gaussian noise, etc. The images of defect-free seeds were labeled as "good," and the images of defective seeds were labeled as "bad." Eventually, there were 3,913 images (RGB + NIR) of defect-free seeds and 3,913 images of defective seeds, respectively. The training set and the testing set were divided by 4:1, and therefore, 5,869 images with single seed were used for training, and 1,957 images were used for testing.

Watershed Algorithm

Every single seed in the image deck was segmented using the watershed algorithm. First, the original 4-channel image was converted to a grayscale image. By comparing the four layers (R, G, B, and NIR), the results showed that the B-channel image was the best to use for binarization. Binarization was then performed, and any noise in the binary image was removed by a morphological open operation. An expansion operation was then applied to the binary image, and a distance transforming algorithm was used to obtain the central region of each seed. The edge of the seed was the dilational image subtracted from the central regions. The central region of each seed was then naturally separated from each other. Finally, the watershed algorithm was used to extract the edge of the seeds, and each seed was segmented in the image by position coordinates. The segmentation processes are shown in **Figure 4**. The NIR images were then segmented using the position coordinates from the segmented RGB images. The combination of the RGB image and NIR image of each seed was used for training or detection processes.

Corn-Seed-Net Model Structure

Every single seed was separated by the watershed algorithm, and the position coordinates were obtained. The CNN model was then used to detect the quality of the corn seeds. The detection results were marked in the image according to the position coordinates. In this article, a two-pathway CNN, Corn-seed-Net (shown as **Figure 5**), was designed combining VGG16 (Simonyan and Zisserman, 2015) and ResNet50 (He et al., 2016). The model was used to extract deep features of 4-channel corn seed images and then classify them.

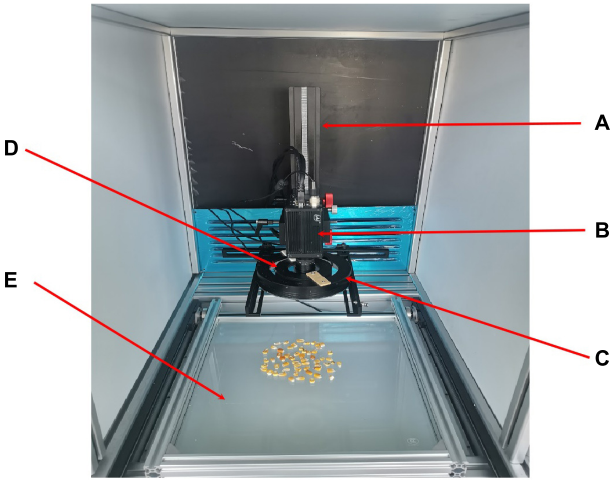


FIGURE 1 | Image acquisition platform. **(A)** Support, **(B)** camera, **(C)** circular white light source, **(D)** circular near-infrared light source, and **(E)** white backlight.

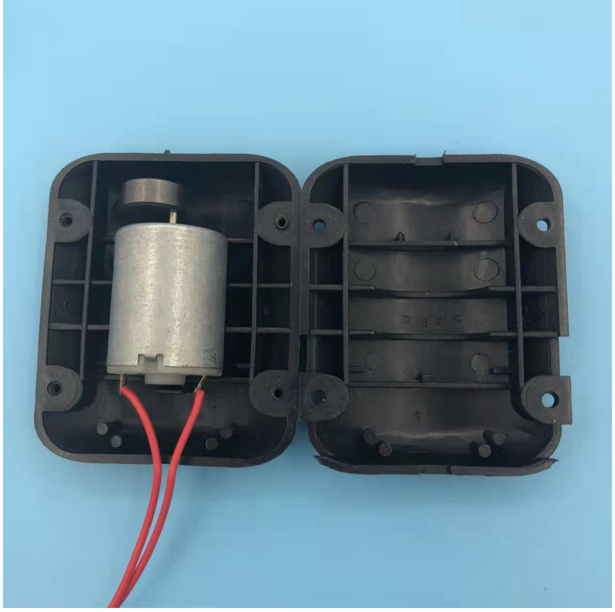


FIGURE 2 | The vibration module.

To reduce the number of parameters, continuous convolution kernels of 3 × 3 were used in the VGG16. Thirteen convolution layers were used to extract deeper image features and increase the fitting capacity and the expressive capacity of the model. However, as the number of network layers increases, the gradient of the model disappears or explodes, which makes the performance of the model plummet. However, the residual structure was added in the ResNet50, the input of the convolution layer was directly added to the output of the convolution layer, and it solves the degradation problem of deep CNN. Therefore, the advantages of both VGG16 and ResNet50 were combined in the Corn-seed-Net.

In this article, the VGG16 branch was optimized. The number of parameters of the last two fully connected layers of the original models was tremendous. To avoid feature information redundancy, a convolution layer of 7 × 7 was applied to the final max-pooling layer, with 512 channels, and two fully connected layers composed of 512 feature vectors were added. In this way, the number of parameters was reduced. For the ResNet50 branch, after the global average pooling layer, a fully connected layer composed of 512 feature vectors was added. The two branches were then fused with the final fully connected layer, and the vectors of the generated features were 1,024. Finally, the classification was completed through the Softmax layer, with the category number set to 2.

The Softmax function was used to calculate the probability of classification, and the calculation formula is as follows:

$$y_{im} = \frac{e^{z_{im}}}{\sum_{k=1}^{k} e^{z_{ik}}} \tag{1}$$

In the formula, y_{im} is the prediction probability that the ith sample belongs to class m, k is the number of categories, z_{im} is the product of the output vector of the ith sample and the parameter vector of class m, and z_{ik} is the product of the output vector of the ith sample and the parameter vector of class k.

Categorical cross-entropy was used to calculate the loss function of the model, and the formula is given as follows:

$$L = -\sum_{i=1}^{n} \hat{y}_{im} \lg y_{im} \tag{2}$$

In the formula, L is the loss function, n is the number of images in each batch, and y_{im} is the expected probability that the ith sample belongs to class m.

Parameter Set

To achieve the best result, the model was trained by setting different parameters. The momentum used in the final model was 0.9, and the initial value of the learning rate was set as 0.001. Stochastic gradient descent (SGD) (Samik and Sukhendu, 2018) algorithm was used with 100 epochs for the training. In the process of training, when the loss of the test set no longer decreased, the learning rate was reduced by half. Other parameters were set to default. The accuracy of the final training set was 100.00%, and the accuracy of the test set was 96.90%.

Hand-Crafted Feature Extraction

In this article, five hand-crafted feature extraction methods were used to extract the features of a single seed segmented by watershed algorithm, and then an SVM classifier was used for seed classification. The feature extraction methods were as follows:

(1) Morphological characteristics (MC) (Zhang L. et al., 2020) were used to binarize each seed's image. The ratio of the perimeter of the seed area, the diameter of the circle with

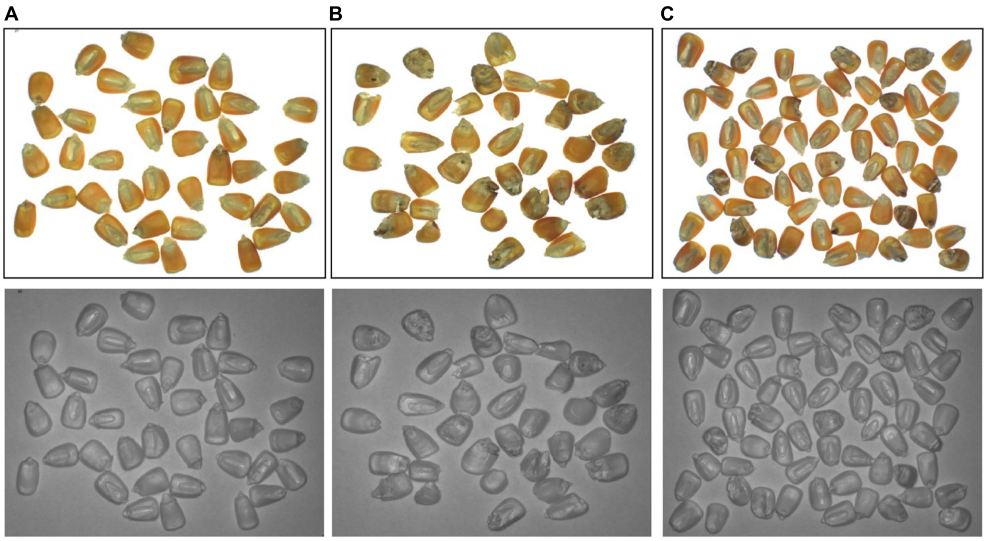


FIGURE 3 | The original image of **(A)** good quality corn seeds, **(B)** disfigured corn seeds, and **(C)** both situations.

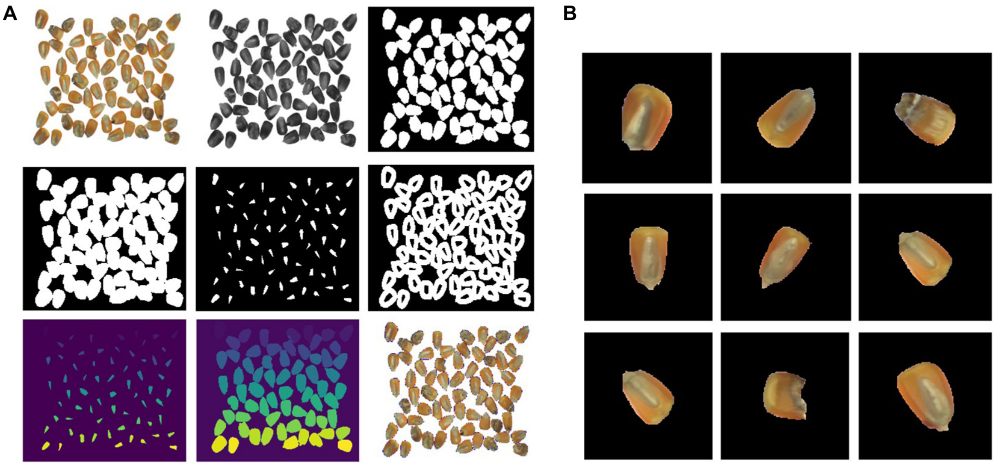


FIGURE 4 | Image processing procedures. **(A)** Segmentation processes and **(B)** segmentation results.

the same area, the eccentricity of the fitted ellipse, the ratio of the major axis to minor axis, and the ratio of area to bounding box area from a connected domain were then extracted. A total of five morphological features were used as feature vectors.

(2) Color features have little dependence on image size and position. In this article, the parameters related to color (RGB) histogram were extracted as feature vectors.

(3) Local shape information can be well captured by histogram of gradient (HOG) (Dalal, 2005), and it is relatively stable to the change of geometry and optics. In this article, the gradient information of the image was extracted as feature vectors.

(4) Gray-level co-occurrence matrix (GLCM) (Haralick et al., 1973) is a method of texture feature extraction based on statistics. The statistics constructed in this article include contrast, dissimilarity, homogeneity, energy, correlation, and angular second moment. These six characteristic parameters were used as feature vectors.

(5) Local binary pattern (LBP) (Ojala et al., 2002) features have the advantages of gray invariance and rotation invariant. In this article, the LBP value of the image was extracted and used to represent the texture information of the region. Finally, the statistical histogram of LBP features was used as the feature vectors.

Evaluation Index

In this article, to evaluate the accuracy and stability of the training model for seed quality identification, the precision and recall ratios were used to evaluate the model, and the F_1 value was used

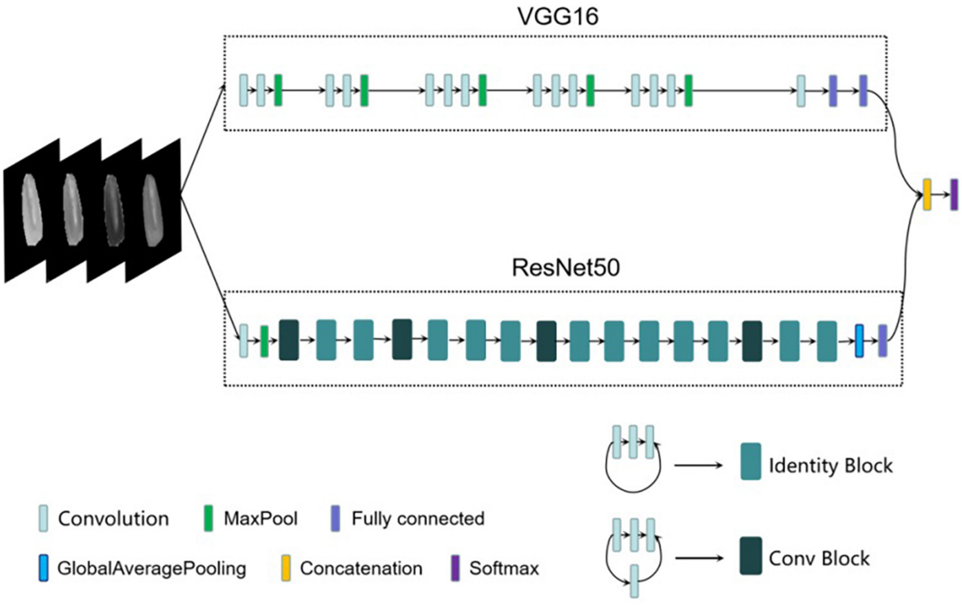


FIGURE 5 | Corn-seed-Net network architecture.

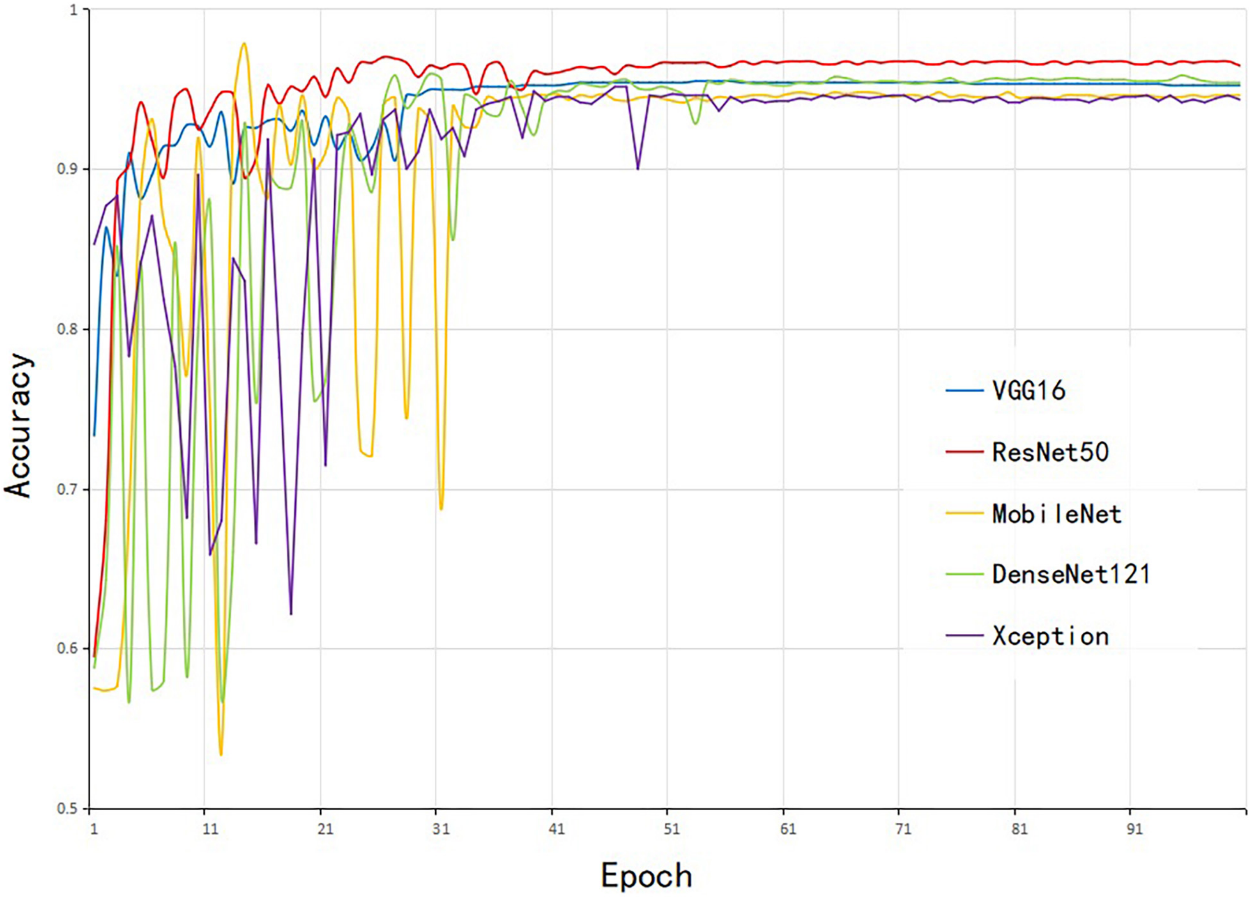


FIGURE 6 | The accuracy of the five models for the test set.

TABLE 1 | Training results of Corn-seed-Net with different model parameters.

Initial learning rate	Training algorithm	Epoch time/s	Training accuracy/%	Validation accuracy/%
0.001	Adam	180	100.00	94.23
0.0001	Adam	180	100.00	95.80
0.001	SGD	165	100.00	96.90
0.0001	SGD	165	99.98	94.59

as the average evaluation of them. The evaluation formulas are given as follows:

$$p = \frac{n\mathrm{TP}}{n\mathrm{TP} + n\mathrm{TF}} \times 100\% \tag{3}$$

$$R = \frac{n\mathrm{TP}}{n\mathrm{TP} + n\mathrm{FN}} \times 100\% \tag{4}$$

$$F1 = \frac{2PR}{P + R} \times 100\% \tag{5}$$

where n_{TP} is the number of corn seeds correctly identified, n_{FP} is the number of misidentified corn seeds, and n_{FN} is the number of unrecognized corn seeds.

RESULTS AND DISCUSSION

The Selection of Corn-Seed-Net Model Structure

To select an optimal model structure, the 4-channel images (RGB + NIR) were used for training five CNN models [e.g., VGG16, ResNet50, MobileNet, DenseNet121 (Huang et al., 2016), and Xception (Chollet, 2017)]. The weights trained on the ImageNet dataset were used for parameter initialization, and the same dataset was trained for the model; the accuracy of the test set is shown in **Figure 6**. The number of input channels for the model was set to 4, but the number of convolution layers was unchanged. The number of parameters for convolutional layers was the same as the original model, and therefore, the pre-trained weights from ImageNet were used in this study. The five models converged after 50 epochs, and the accuracy stabilized at a high value. For the five models, the ResNet50 model had the highest accuracy of 96.63%. The VGG16 model converged most rapidly. The DenseNet model achieved the characteristics of dense connection through repeated splicing, but the running memory consumption was large and the convergence time was long. The MobileNet model possessed a smaller amount of parameters but the accuracy was low. Deep separable convolution and residual connection were

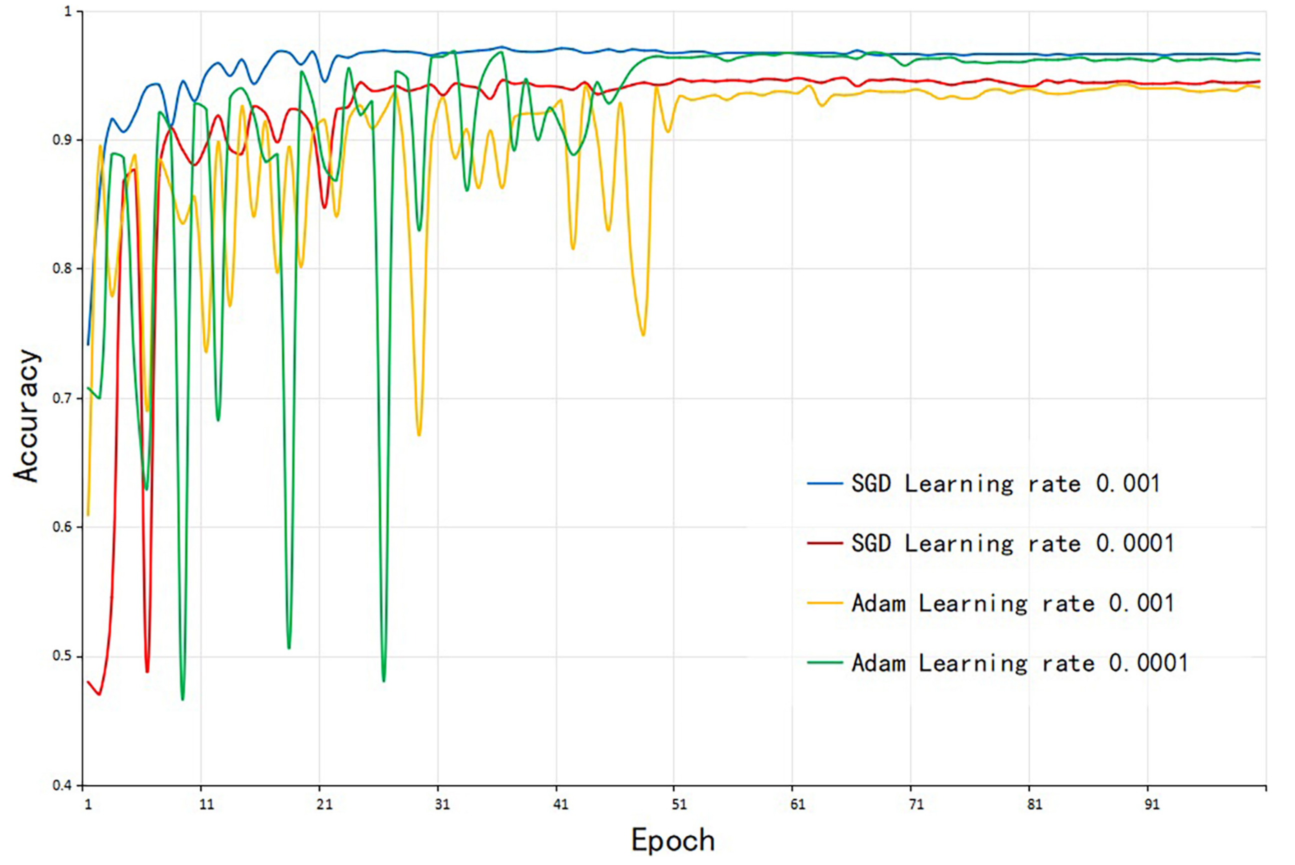


FIGURE 7 | The accuracy of Corn-seed-Net with different model parameters.

TABLE 2 | Test results of single seed with different models.

Model	Classes	Predict classes		Model performance		
		Good	Bad	Accuracy/%	Averaged accuracy/%	Detection time/ms
VGG16	Good	99	1	99	99.00	45.5
	Bad	1	99	99		
ResNet50	Good	100	0	100	99.00	41.7
	Bad	2	98	98		
MobeliNet	Good	97	3	97	98.00	22.9
	Bad	1	99	99		
DenseNet121	Good	97	3	97	97.50	58.0
	Bad	2	98	98		
Xception	Good	98	2	98	97.50	41.0
	Bad	3	97	97		
Corn-seed-Net	Good	100	0	100	100.00	68.0
	Bad	0	100	100		

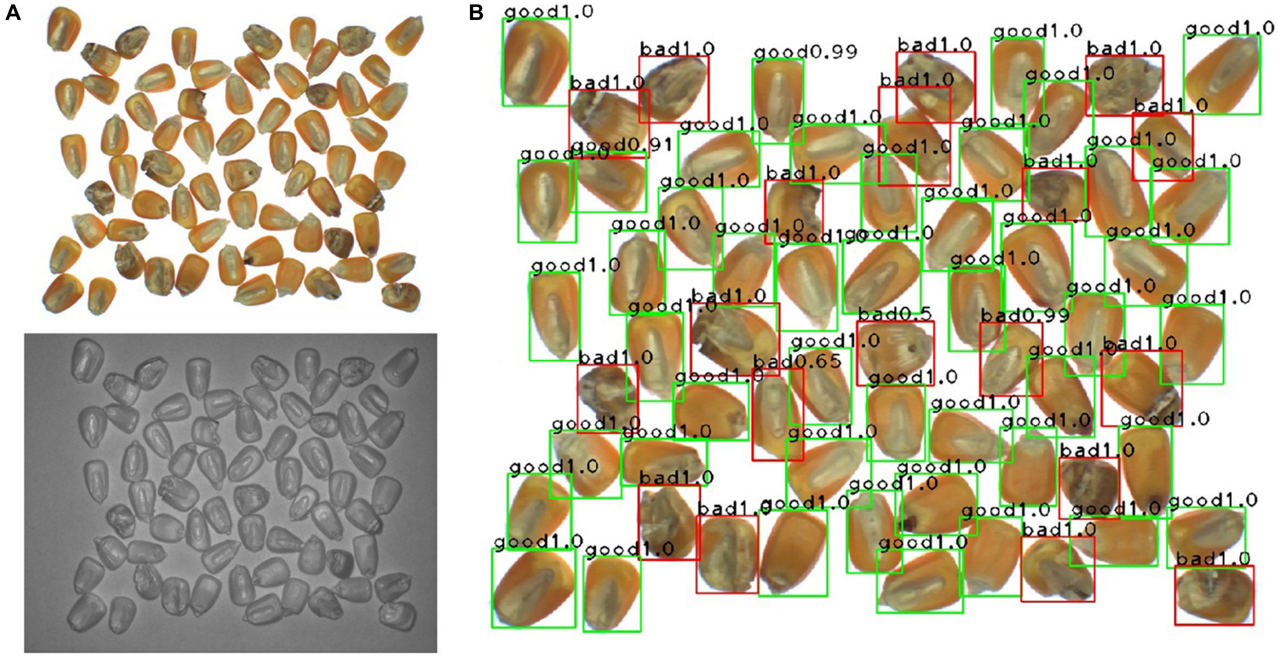


FIGURE 8 | (A) The original image and **(B)** object detection results.

used in the Xception model, and the accuracy was also relatively low. Considering the accuracy and convergence, VGG16 and ResNet50 were combined to construct the final model.

The Selection of Corn-Seed-Net Model Parameters

To obtain faster training speed and better convergence performance of the model, the same 4-channel images were used to train the model, and two branches of Corn-seed-Net were initialized with the weights trained using the ImageNet dataset. The influences of different initial learning rates and different optimization algorithms on the model were tested (as shown in **Table 1**). **Figure 7** shows that the SGD algorithm converged faster, and the Adam algorithm was unstable in the first half of the training process. Therefore, the SGD optimization algorithm was used in the experiment, and the initial learning rate was set to 0.001.

Test Results of the Corn-Seed-Net Model on a Single Seed

To verify the classification accuracy of the Corn-seed-Net model for 4-channel images of a single seed, 100 corn seeds without any defects in appearance and another 100 seeds with defected appearance were selected in this experiment. At the same time, other one-pathway CNN models were compared, and the results are shown in

TABLE 3 | Comparison of model performance combined with watershed algorithm.

Model	Classes	Precision/%	Recall/%	Averaged precision/%	Averaged recall/%	F1/%	Detection time/ms
VGG16	Good	90.91	96.55	93.05	94.71	93.87	139.5
	Bad	95.19	92.86				
ResNet50	Good	94.00	97.24	94.69	94.60	94.64	122.5
	Bad	95.37	91.96				
MobeliNet	Good	93.85	94.48	91.96	93.22	92.59	95.95
	Bad	91.96	91.96				
DenseNet121	Good	91.45	95.86	92.43	92.13	92.27	136.75
	Bad	93.40	88.39				
Xception	Good	94.48	94.48	93.26	93.67	93.46	133.85
	Bad	92.03	92.86				
Corn-seed-Net	good	94.08	98.62	95.63	95.29	95.46	149.55
	bad	97.17	91.96				

TABLE 4 | Comparison of model performance combined with watershed algorithm using RGB images.

Model	Classes	Precision/%	Recall/%	Averaged precision/%	Averaged recall/%	F1/%	Detection time/ms
RGB VGG16	Good	90.13	94.48	91.67	90.10	90.87	69.92
	Bad	93.20	85.71				
RGB ResNet50	Good	95.56	95.86	94.54	93.02	93.77	98.83
	Bad	93.52	90.18				
RGB Corn-seed-Net	Good	93.33	96.55	93.80	92.47	93.13	117.63
	Bad	94.28	88.39				
RGB + NIR Corn-seed-Net	Good	94.08	98.62	95.63	95.29	95.46	149.55
	Bad	97.17	91.96				

Table 2. The averaged accuracy of the Corn-seed-Net model for each single seed classification is up to 100%, which was better than other one-pathway models. The averaged detection time for a single seed was 68 ms, which indicated that the two-pathway model is suitable for seed classification.

Detection Results of the Corn-Seed-Net Model Combined With the Watershed Algorithm

To accurately locate each seed with the quality rating, the watershed algorithm was adopted and combined with the Corn-seed-Net model on 4-channel images of corn seed (**Figure 8A**). The conglutinated seeds were segmented using the watershed algorithm, and meanwhile, the position coordinates of each seed were also obtained. The detection results are shown in **Figure 8B**.

To evaluate the performance of this method, 10 groups of images were used for verification. At the same time, other one-pathway CNNs were compared, and the results are shown in **Table 3**. It showed that the watershed algorithm combined with the Corn-seed-Net model had the highest precision and recall rate on average, and the F1 value was 95.46%. Due to the addition of the operation of the watershed segmentation during image detection, there was an increase in the detection time, and the averaged detection time for a single seed was 149.55 ms. The results appeared that the model performance improves when the watershed algorithm was adopted and combined with two-pathway CNN.

RGB Images Detection Results

To investigate whether using 4-channel images (RGB + NIR) is superior to 3-channel images (RGB) in seed classification, RGB images of the same dataset were used in the experiment, with watershed algorithm combined with Corn-seed-Net model, and the results are shown in **Table 4**. It is shown that the extra information carried with the NIR band improved the model performance on both precision and recall rate, compared with the models obtained with RGB images only.

To fully evaluate the performance of the watershed algorithm combined with the Corn-seed-Net model, the watershed algorithm combined with the traditional feature extraction method was also studied, and SVM was used to classify

TABLE 5 | Comparison of object detection results.

Model	Average precision/%	Average recall/%	F1/%
GLCM + SVM	22.05	48.11	30.24
Color + SVM	60.97	58.74	59.83
HOG + SVM	64.30	64.07	64.18
MC + SVM	68.64	68.17	68.40
LBP + SVM	74.28	73.73	74.00
Corn-seed-Net	95.63	95.29	95.46

the corn seeds based on their quality. The results are shown in **Table 5**, and it indicates that the precision, recall, and F1 of the method we have proposed were all significantly higher than those of the traditional methods, as more deep image features were extracted in CNN.

DISCUSSION

At present, some studies have been devised in seed classification using imaging technology combined with machine learning and deep learning (Huang et al., 2019; Kozowski et al., 2019; Ansari et al., 2021). However, most of the studies were based on RGB imaging technology rather than using four-channel multispectral images. Moreover, there are few studies on seed quality detection using the current typical object detection algorithm. This article designed an end-to-end object detection model, and high accuracy was achieved in seed quality detection.

In this article, RGB and NIR images of corn seeds were obtained using a multispectral camera, and the watershed algorithm combined with the Corn-seed-Net model was used to predict the quality of corn seeds. The watershed algorithm is used to segment every single seed and obtain the precise location of the seed. At the same time, while the 4-channel image data with both RGB and NIR bands were used as the inputs of the Corn-seed-Net model, the accuracy of the model was better than that with RGB images only.

The Corn-seed-Net model combines the advantages of VGG16 and ResNet50, and deeper information could be extracted by deep networks. It employs a residual network structure, and the effect of degradation of the deep network is eliminated. With the optimized model, 200 single corn seeds were used for verification and compared with other single-pathway models, and the results revealed that the average classification accuracy of the Corn-seed-Net model reached 100.00%.

To evaluate the corn seed defect detection performance of the watershed algorithm combined with the Corn-seed-Net model, we compared the detection results with RGB images and traditional feature extraction methods. The experimental results showed that the proposed method in this article had the best performance, with an average precision of 95.63%, an average recall rate of 95.29%, and an F1 value of 95.46%.

CONCLUSION

In this study, an end-to-end corn seed object detection model was proposed, which combined watershed segmentation algorithm and CNNs. In comparison with mainstream object detection models (e.g., Faster-RCNN, SSD, and YOLO), our method uses a watershed segmentation algorithm to obtain more accurate target positions, which also reduces the complexity of the network at the same time. In addition, this method eliminates the manual annotation of the image and reduces the workload of dataset preparation. In the future, this method can be further optimized by simplifying the network structure, which may shorten the calculation time while ensuring the classification accuracy, to provide a basis for the subsequent development of a quality detection device.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

LW, JL, and XF conceived the idea, proposed the method, and revised the manuscript. LW, JW, and JZ contributed to the preparation of equipment and acquisition of data, wrote the code, and tested the method. LW, JZ, and JL contributed to the validation results. LW and XF wrote the manuscript. All authors read and approved the final manuscript.

FUNDING

This study was supported by the National Natural Science Foundation of China (32072572), the Hebei Talent Support Foundation (E2019100006), the Key Research and Development Program of Hebei Province (20327403D), the Talent Recruiting Program of Hebei Agricultural University (YJ201847), and the special project of talent research introduced by Hebei Agricultural University (YJ2020064).

REFERENCES

Afonso, M., Blok, P. M., Polder, G., Wolf, J., and Kamp, J. (2019). Blackleg detection in potato plants using convolutional neural networks. *IFAC PapersOnLine* 52, 6–11. doi: 10.1016/j.ifacol.2019.12.481

Afzal, I., Bakhtavar, M. A., Ishfaq, M., Sagheer, M., Baributsa, D., et al. (2017). Maintaining dryness during storage contributes to higher maize seed quality. *J. Stored Prod. Res.* 72, 49–53. doi: 10.1016/j.jspr.2017.04.001

Ali, A., Qadri, S., Mashwani, W. K., and Belhaouari, S. B. (2020). Machine learning approach for the classification of corn seed using hybrid features. *Int. J. Food Prop.* 23, 1097–1111. doi: 10.1080/10942912.2020.1778724

Altuntaş, Y., Cömert, Z., and Kocamaz, A. F. (2019). Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach. *Comput. Electron. Agric.* 163:104874. doi: 10.1016/j.compag.2019.104874

Ansari, N., Ratri, S. S., Jahan, A., Ashik-E-Rabbani, M., and Rahman, A. (2021). Inspection of paddy seed varietal purity using machine vision and multivariate analysis. *J. Agric. Food Res.* 3:100109. doi: 10.1016/j.jafr.2021.100109

Arunachalam, A., and Andreasson, H. (2021). Real-time plant phenomics under robotic farming setup: a vision-based platform for complex plant phenotyping tasks. *Comput. Electrical Eng.* 92:107098. doi: 10.1016/j.compeleceng.2021.107098

Azimi, S., Kaur, T., and Gandhi, T. K. (2020). A deep learning approach to measure stress level in plants due to nitrogen deficiency. *Measurement* 173:108650. doi: 10.1016/j.measurement.2020.108650

Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Honolulu: IEEE.

Dalal, N. (2005). "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, (San Diego, CA: IEEE), 886–893.

Gao, F., Fu, L., Zhang, X., Majeed, Y., and Zhang, Q. (2020). Multi-class fruit-on-plant detection for apple in snap system using faster r-cnn. *Comput. Electron. Agric.* 176:105634. doi: 10.1016/j.compag.2020.105634

Gutiérrez, S., Wendel, A., and Underwood, J. (2019). Spectral filter design based on in-field hyperspectral imaging and machine learning for mango ripeness estimation. *Comput. Electron. Agricu.* 164:104890. doi: 10.1016/j.compag.2019.104890

Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification, IEEE Trans. on Systems, Man and Cybernetics. *SMC* 3, 610–621. doi: 10.1109/tsmc.1973.4309314

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "*Deep Residual Learning for Image Recognition*, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV: IEEE).

Huang, G., Liu, Z., Laurens, V., and Weinberger, K. Q. (2016). *Densely Connected Convolutional Networks*. Honolulu, HI: IEEE Computer Society.

Huang, S., Fan, X., Sun, L., Shen, Y., and Suo, X. (2019). Research on classification method of maize seed defect based on machine vision. *J. Sens.* 2019, 11–25. doi: 10.1155/2019/2716975

Keiichi, M., Satoru, K., and Komaki, I. (2019). Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. *GigaScience* 8:giy153. doi: 10.1093/gigascience/giy153

Ke-Ling, T. U., Lin-Juan, L. I., Yang, L. M., Wang, J. H., and Sun, Q. (2018). Selection for high quality pepper seeds by machine vision and classifiers. *J. Integr. Agric.* 17, 1999–2006. doi: 10.1016/s2095-3119(18)62031-3

Kiratiratanapruk, K., and Sinthupinyo, W. (2012). "Color and Texture for Corn Seed Classification by Machine Vision," in *International Symposium on Intelligent Signal Processing and Communications Systems*, (Chiang Mai: IEEE), doi: 10.1109/ISPACS.2011.6146100

Kozowski, M., Górecki, P., and Szczypiński, P. M. (2019). Varietal classification of barley by convolutional neural networks. *Biosyst. Eng.* 184, 155–165.

Kusumaningrum, D., Lee, H., Lohumi, S., Mo, C., Kim, M. S., and Kwan Cho, B. (2018). Non-destructive technique for determining the viability of soybean (glycine max) seeds using ft-nir spectroscopy. *J. Sci. Food Agric.* 98, 1734–1742. doi: 10.1002/jsfa.8646

Laabassi, K., Belarbi, M. A., Mahmoudi, S., Mahmoudi, S. A., and Ferhat, K. (2021). Wheat varieties identification based on a deep learning approach. *J. Saudi. Soc. Agric. Sci.* 20, 281–289. doi: 10.1016/j.jssas.2021.02.008

Lei, T., Jia, X., Liu, T., Liu, S., Meng, H., and Nandi, A. K. (2019). Adaptive morphological reconstruction for seeded image segmentation. *IEEE Trans. Image Process.* 28, 5510–5523. doi: 10.1109/TIP.2019.2920514

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (Chiang Mai: IEEE), 971–987. doi: 10.1109/tpami.2002.1017623

Pang, L., Men, S., Yan, L., and Xiao, J. (2020). Rapid vitality estimation and prediction of corn seeds based on spectra and images using deep learning and hyperspectral imaging techniques. *IEEE Access.* 99, 1–1. doi: 10.1109/ACCESS.2020.3006495

Rehman, T. U., Mahmud, M. S., Chang, Y. K., Jin, J., and Shin, J. (2018). Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Comput. Electron. Agric.* 156, 585–605. doi: 10.1016/j.compag.2018.12.006

Samik, B., and Sukhendu, D. (2018). Mutual variation of information on transfer-cnn for face recognition with degraded probe samples. *Neurocomputing*.310, 299–315. doi: 10.1016/j.neucom.2018.05.038

Simonyan, K., and Zisserman, A. (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Published as a conference paper at ICLR 2015, (Chiang Mai: IEEE). doi: 10.3390/s21082852

Sj, A., Shma, B., Fjv, A., and Am, C. (2021). Computer-vision classification of corn seed varieties using deep convolutional neural network. *J. Stored Prod. Res.* 92:101800. doi: 10.1016/j.jspr.2021.101800

Sta, B., Xu, M. B., Zm, A., Long, Q. B., and Yw, B. (2019). Segmentation and counting algorithm for touching hybrid rice grains. *Comput. Electron. Agric.* 162, 493–504. doi: 10.1016/j.compag.2019.04.030

Tiwari, V., Joshi, R. C., and Dutta, M. K. (2021). Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecol. Inform.* 63:101289. doi: 10.1016/j.ecoinf.2021.101289

Zhang, C., Zhao, Y., Yan, T., Bai, X., and Liu, F. (2020). Application of near-infrared hyperspectral imaging for variety identification of coated maize kernels with deep learning. *Infrared Phys. Technol.* 111:103550. doi: 10.1016/j.infrared.2020.103550

Zhang, J. J., Ma, Q., Cui, X., Guo, H., and Zhu, D. H. (2020). High-throughput corn ear screening method based on two-pathway convolutional neural network. *Comput. Electron. Agric.* 175:105525. doi: 10.1016/j.compag.2020.105525

Zhang, L., Li, N., and Fang, H. (2020). Morphological Characteristics and Seed Physiochemical Properties of Two Giant Embryo Mutants in Rice. *Rice Sci.* 27, 81–85. doi: 10.1016/j.rsci.2019.04.006

Zhang, L., Zou, L., Wu, C., Jia, J., and Chen, J. (2021). Method of famous tea sprout identification and segmentation based on improved watershed algorithm. *Comput. Electron. Agric.* 184:106108. doi: 10.1016/j.compag.2021.106108

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: 09 June 2022
doi: 10.3389/fpls.2022.914771

Machine Learning Approaches for Rice Seedling Growth Stages Detection

*Suiyan Tan[1], Jingbin Liu[1], Henghui Lu[1], Maoyang Lan[1], Jie Yu[1], Guanzhong Liao[1], Yuwei Wang[2], Zehua Li[3], Long Qi[2] and Xu Ma[2]**

[1] College of Electronic Engineering, South China Agricultural University, Guangzhou, China, [2] College of Engineering, South China Agricultural University, Guangzhou, China, [3] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

Recognizing rice seedling growth stages to timely do field operations, such as temperature control, fertilizer, irrigation, cultivation, and disease control, is of great significance of crop management, provision of standard and well-nourished seedlings for mechanical transplanting, and increase of yield. Conventionally, rice seedling growth stage is performed manually by means of visual inspection, which is not only labor-intensive and time-consuming, but also subjective and inefficient on a large-scale field. The application of machine learning algorithms on UAV images offers a high-throughput and non-invasive alternative to manual observations and its applications in agriculture and high-throughput phenotyping are increasing. This paper presented automatic approaches to detect rice seedling of three critical stages, BBCH11, BBCH12, and BBCH13. Both traditional machine learning algorithms and deep learning algorithms were investigated the discriminative ability of the three growth stages. UAV images were captured vertically downward at 3-m height from the field. A dataset consisted of images of three growth stages of rice seedlings for three cultivars, five nursing seedling densities, and different sowing dates. In the traditional machine learning algorithm, histograms of oriented gradients (HOGs) were selected as texture features and combined with the support vector machine (SVM) classifier to recognize and classify three growth stages. The best HOG-SVM model obtained the performance with 84.9, 85.9, 84.9, and 85.4% in accuracy, average precision, average recall, and F1 score, respectively. In the deep learning algorithm, the Efficientnet family and other state-of-art CNN models (VGG16, Resnet50, and Densenet121) were adopted and investigated the performance of three growth stage classifications. EfficientnetB4 achieved the best performance among other CNN models, with 99.47, 99.53, 99.39, and 99.46% in accuracy, average precision, average recall, and F1 score, respectively. Thus, the proposed method could be effective and efficient tool to detect rice seedling growth stages.

Keywords: rice seedling, machine learning, deep learning, growth stage, histograms of oriented gradients, SVM

Edited by:
Wanneng Yang,
Huazhong Agricultural University,
China

Reviewed by:
Jian Zhang,
Huazhong Agricultural University,
China
Jingjing Zhai,
Northwest A&F University, China

****Correspondence:***
Xu Ma
maxu1959@scau.edu.cn

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *07 April 2022*
Accepted: *04 May 2022*
Published: *09 June 2022*

Citation:
Tan S, Liu J, Lu H, Lan M, Yu J, Liao G, Wang Y, Li Z, Qi L and Ma X (2022) Machine Learning Approaches for Rice Seedling Growth Stages Detection. Front. Plant Sci. 13:914771. doi: 10.3389/fpls.2022.914771

INTRODUCTION

Rice is the most important grain crop that feeds more than half of the world's population (Ruiz-Sánchez et al., 2010; Abid et al., 2015; Kargbo et al., 2016). It ranks first among the grain crops in China. Currently, commercial farming of rice mostly employs transplanting techniques, where seeds are sown and raised into seedlings in the nursery trays. Seedlings are later transplanted using compatible machinery (Tan et al., 2019). Healthy, disease-free, and well-nourished seedlings with uniform growth are the prerequisites for uniform field transplantation, and these seedlings must meet certain technical standards in the system of mechanical transplanting (Biswas et al., 2000; Cheng et al., 2018). Precise temperature control and proper timing of fertilizer, irrigation, cultivation, and disease control at different seedling growth stages must be considered to raise the standard seedlings. Thus, knowing the growth stages of seedling allows growers to properly time field operations to raise seedlings. Moreover, studies found out that the age of seedling at transplanting had a great impact on grain yield. Transplanting young seedling early and with high tiller production enhanced grain yield (Pasuquin et al., 2008; Ohsumi et al., 2012). However, the rice seedlings are often raised in paddy field. Environmental factors, such as changes in temperature, solar radiation, and rainfall, affect many traits that are responsible for the growth stages, including leaf photosynthesis (Makino et al., 1994; Maruyama and Nakamura, 1997), efficiency of nitrogen (N) used for leaf photosynthesis (Nagai and Makino, 2009), leaf emergence (Hiraoka et al., 1987), leaf elongation (Cutler et al., 1980), and the allocation of biomass and N to leaf (Kanno et al., 2009). Therefore, monitoring the seedling growth stage is crucial to ensure to have seedling transplant at the most suitable age. The phenology staging system of rice refers to BBCH scale which uses a decimal code to describe the growth of crops (Lancashire et al., 1991). For example, BBCH [10–19] represents seedling stages which is here the 0–9 leaves' development. The appropriate age of rice seedling at transplant is no later than BBCH13. At present, seedling growth stage detection mainly relies on manual field inspection, which is time-consuming, labor-intensive, and inaccurate. When large-scale field involved, manual inspections become inefficient. Therefore, there is a need for a low-cost, accurate, rapid, and objective approach for rice seedling growth stage detection.

During the entire growth cycle, crops change significantly in their external morphological structures and could be observed visually, which enables us to explore new technologies to automatically observe, detect, and distinguish different critical growth stages of crops. Computer vision technology has been reported in the application of seedling quality and growth stage detection. Tong et al. (2013) developed an improved watershed segmentation for overlapping leaf images and applied it to test the crop seedling quality. Yu et al. (2013) explored the application of computer vision to automatically detect two critical growth stages of maize, including the emergence and three-leaf stage. In this study, a crop segmentation method, namely, AP-HI, was put forward to extract the plants from images. Then, the spatial distribution feature was used to judge whether the field crop had reached the emergence stage or not. Skeleton endpoint detection was used to characterize the leaf of seedling and to judge whether the field crop had reached the three-leaf stage or not. Recently, Li et al. (2021) utilized computer vision to detect rice seedling hill in the paddy field. The preferred laboratory color model along with Otsu's method was used to extract rice seedling information, and the skeleton of the seedling hill was extracted using the thinning algorithm to effectively characterize the morphological structure of single seedling hill. Similar studies of seedling quality detection have been reported in wheat (Zhu et al., 2016), cotton (Chen et al., 2018), and rapeseed (Zhao et al., 2018).

With the rapid development of big data technology and high-performance computing, the machine learning technology has been widely used in the recent years to meet the growing demand for fast, accurate, and non-destructive applications in precision agriculture. Numerous applications of machine learning technology are reported in agricultural automation, such as yield estimation (Yang et al., 2019; Chu and Yu, 2020; Zhao et al., 2021), disease detection (Chowdhury et al., 2021; Zhang et al., 2021; Farman et al., 2022), weeds identification (Jin et al., 2021; Pandey et al., 2021), and continuous monitoring of crop status (Han et al., 2021; Taylor and Browning, 2022).

In the traditional machine learning algorithms, color, texture, and thermal features, which are extracted from RGB, multispectral and thermal images, are then fed into different machine learning algorithms, such as nearest neighbors, linear discriminant analysis, random forest, and support vector machine (SVM) to finish specific tasks. Histograms of oriented gradient (HOG) are a feature descriptor representing an image with a set of local histograms counting the occurrences of gradient orientations within a local image cell. It was successfully applied for pedestrian detection by Dalal and Triggs (2005), and the HOG descriptors significantly outperformed existing feature sets for human detection. HOG feature is widely reported in precision agriculture. Tan et al. (2018) calculated the HOG feature vectors from original color images of blueberry fruit, and then, a linear SVM classifier was trained to detect the fruit-like regions rapidly. Abouzahir et al. (2021) used the HOG to improve the performance for weed detection. In this study, HOG blocks were used as the key points to generate the visual words. A backpropagation neural network was adopted to detect weeds and classify plants for three different crop fields. This method classified plants with an accuracy of 90.4, 92.4, and 94.1% in sugar beet, carrot, and soybean fields, respectively.

The deep learning algorithms, a relatively new area of machine learning, allow computational models that are composed of multiple processing layers to learn complex data representations using artificial intelligence for image processing and data analysis (Liakos et al., 2018). One of the main advantages of deep learning algorithms is that the step of feature extraction is performed by the model itself. The performance of deep learning algorithms far exceeds that of the traditional machine learning in many applications. In fact, deep learning has been reported in the application of crop critical growth stage detection. Velumani et al. (2020) trained a convolution neural network (CNN) to identify the presence of wheat spikes in small patches acquired by a fixed RGB camera in the field. The heading date was then estimated from the dynamics of the spike presence in the patches over time. In a similar study, Bai et al. (2018) determined the arrival of the rice heading stage by the number of the spike patches detected by a CNN network. Rasti et al. (2021) investigated wheat and barley growth stage estimation by classification of proximal images using deep learning algorithm.

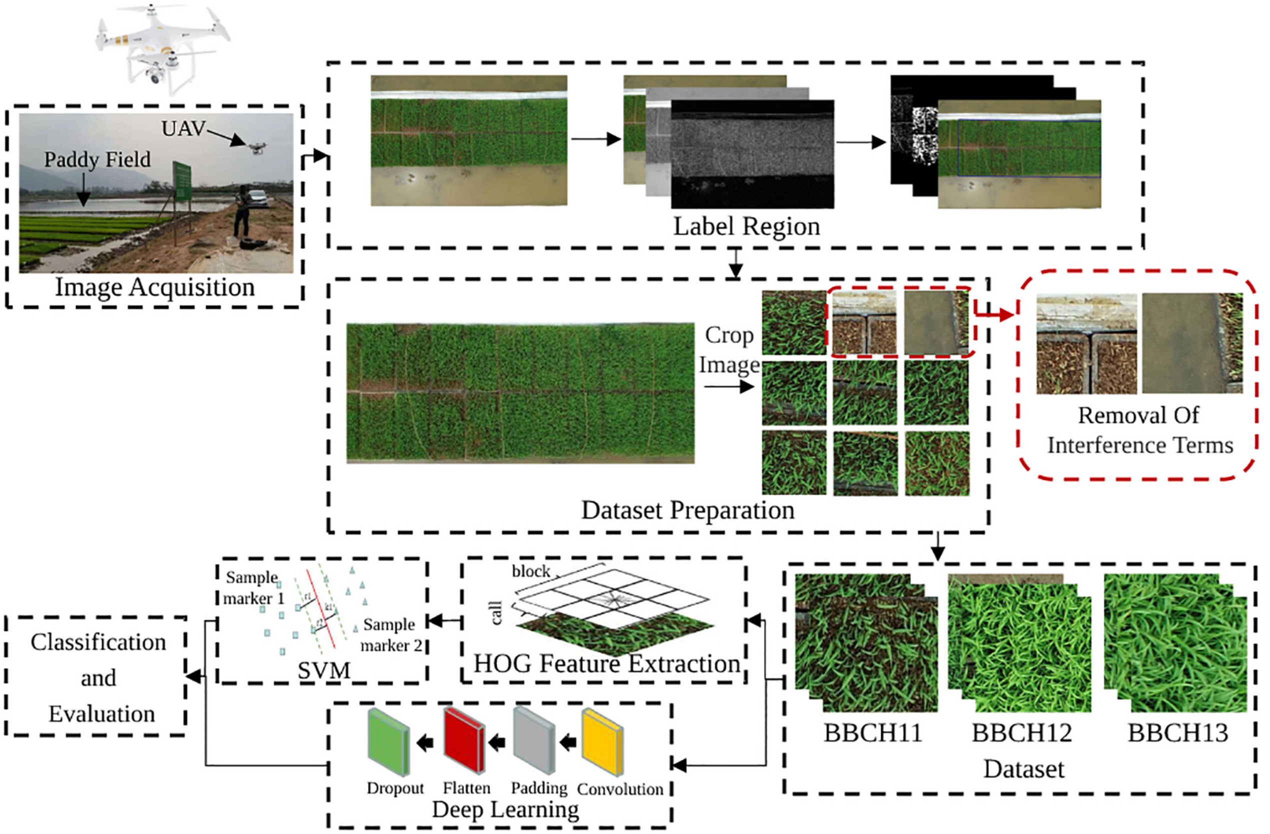


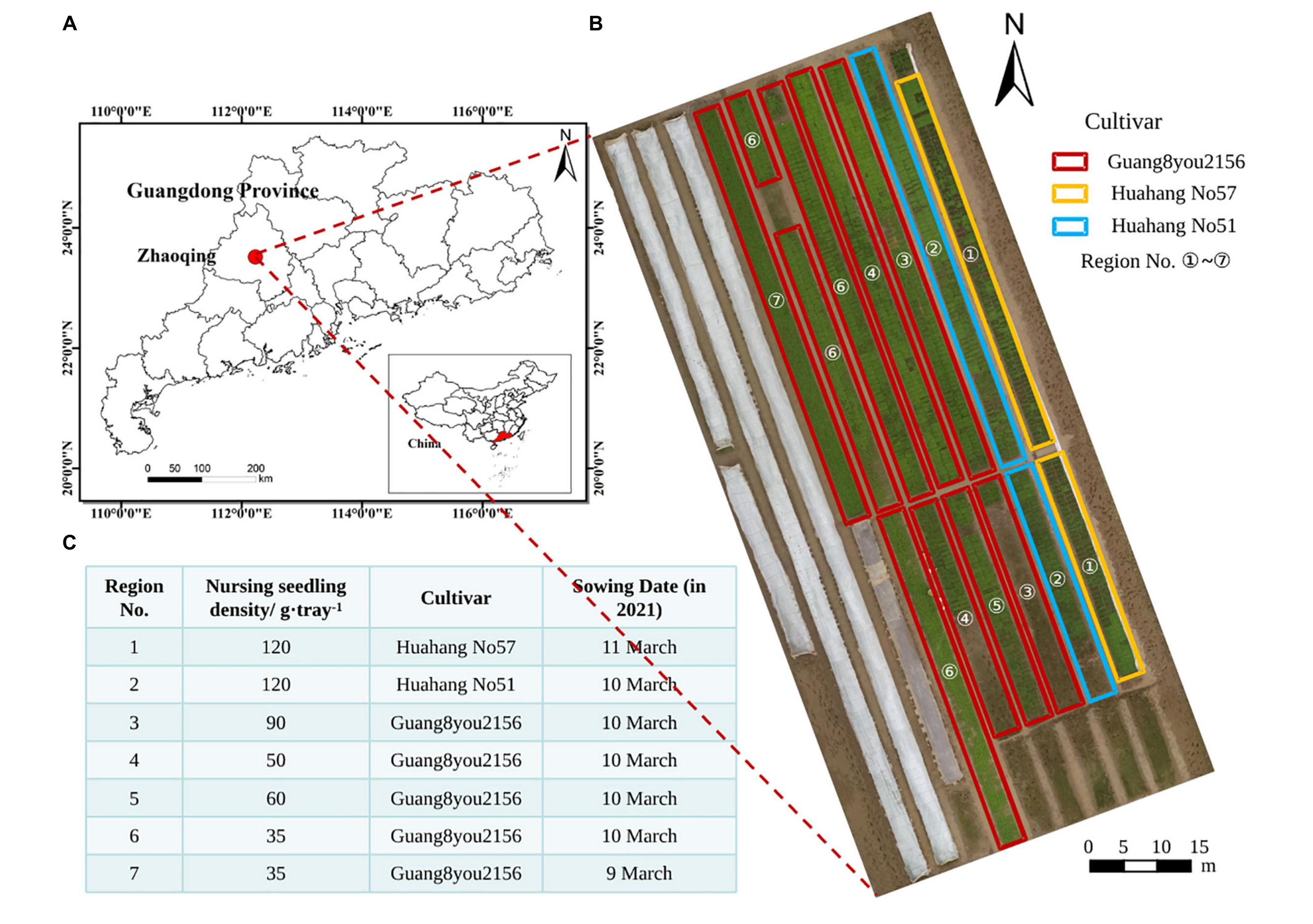
FIGURE 1 | The schematic diagram of the proposed method.

The classification was carried out using three different machine learning approaches on an image dataset of 12 growth stages of wheat and 11 growth stages of barley. The three machine learning approaches included a 5-layer CNN, a pretrained VGG19 network, and SVM. In the seedling growth stage detection, Samiei et al. (2020) developed four deep learning models, including the multiclass CNN, 2-class CNN, CNN-LSTM, and ConvLSTM, to classify three growth stages of two species of red clover and alfalfa. The three growth stages were emergence out of the soil, cotyledon opening, and appearance of the first leaf.

Other studies addressed critical crop growth stage detection through the analysis of a height-based continuous growth curve captured over the entire growth cycle (Zhao et al., 2021). To date, several studies of crop growth stage detection have been reported. However, to our best knowledge, few studies have been reported in rice seedling growth stages detection. On the other hand, with proper sensors well equipped on, unmanned aerial vehicle (UAV) is controllable and capable of performing multiple missions. The UAV platform exhibits many advantages, such as low cost, high spatial, and temporal resolution. Moreover, the application of UAV is highly flexible, and the use of process is relatively simple. Therefore, the combination of UAV technology and machine learning algorithms allows us to detect crop growth stage in a more precise and efficient way. The objective of this study was to explore efficient and robust ways to detect three main growth stages of rice seedlings, including BBCH11, BBCH12, and BBCH13. For this purpose, an RGB camera mounted on a UAV was used to capture the images of a rice paddy field. A total of two types of machine learning algorithms were investigated: (1) For the first time, HOG feature was extracted from the rice seedling canopy images, and then, SVM was adopted to classify the seedlings into three growing stages, BBCH11, BBCH12, and BBCH13. (2) Different deep learning models were adopted to classify the three seedling growth stages. (3) The performance of the machine learning algorithms was finally evaluated and compared by proper evaluation indexes, including accuracy, average precision, average recall, and F1 score.

MATERIALS AND METHODS

In this study, the main processes of rice seedling growth stages detection, such as field data acquisition, image preprocessing, machine learning model applications, and model performance evaluations, are summarized in **Figure 1**. First, RGB images of rice seedling were acquired using a DJI Phantom4 RTK UAV (DJI Innovations, Shenzhen, China), and a series of image preprocessing was performed to prepare the datasets.



Region No.	Nursing seedling density/ g·tray^{-1}	Cultivar	Sowing Date (in 2021)
1	120	Huahang No57	11 March
2	120	Huahang No51	10 March
3	90	Guang8you2156	10 March
4	50	Guang8you2156	10 March
5	60	Guang8you2156	10 March
6	35	Guang8you2156	10 March
7	35	Guang8you2156	9 March

FIGURE 2 | Study site and experimental design. **(A)** Location of the study site; **(B)** orthomosaic; **(C)** field tray nursing seedling experiment designs.

TABLE 1 | Details of RGB images acquisition and the corresponding phenological growth stages of the rice seedlings.

Inspection date (2021)	16 March		19 March		24 March	
Region No.	Growth stage	Images acquired	Growth stage	Images acquired	Growth stage	Images acquired
1.	BBCH11	56	BBCH12	83	BBCH13	108
2.	BBCH11	46	BBCH12	60	BBCH13	85
3.	BBCH11	45	BBCH12	92	BBCH13	108
4.	BBCH11	69	BBCH12	98	BBCH13	96
5.	BBCH11	20	BBCH12	32	BBCH13	28
6.	BBCH11	84	BBCH12	120	BBCH13	102
7.	×	×	BBCH12	55	BBCH13	40

Second, datasets created from the UAV images combined with field observations were processed through two groups of machine learning methods, namely, traditional machine learning and deep learning algorithms. Third, model performances were finally evaluated and compared, with the most desirable one(s) recommended.

Study Sites and Field Experiments

This study was a part of a comprehensive rice field experiments conducted at location of Research Centre Shapu, in Zhaoqing, Guangdong Province, China (23.16° N and 11.57° E). On the date 9–11 March 2021, rice seeds were sown onto the nursery trays by the 2ZSB-500 automatic precision rice seeding line. A total of three cultivars and five nursing seedling density were considered in the rice field experiment. A total of three cultivars included Huahang No. 51, Huahang No. 57, and Guang8you2156. A total of five nursing seedling densities, namely, 120 g/tray, 90 g/tray, 60 g/tray, 50 g/tray, and 35 g/tray, were adopted. After sowing seeds onto the trays, a total of 3,000 trays were classified and placed according to

TABLE 2 | Detailed information of the datasets.

The number of images	Growth stages of the seedlings		
	BBCH11	BBCH12	BBCH13
Original images	320	540	567
Image of 600 × 600 pixels	1,130	3,652	4,101
Image of 400 × 400 pixels	2,814	3,545	5,739
Image of 300 × 300 pixels	4,318	5,564	4,672
Image of 200 × 200 pixels	8,083	18,994	25,842
Image of 224 × 224 pixels	12,357	27,830	25,393
Image of 100 × 100 pixels	49,840	89,333	90,759

different sowing experiments and sowing dates. The trays were placed neatly in the paddy field. After the seedlings grew to an appropriate age, they were transplanted to a field of about 2.6 hectares for other comprehensive rice experiments. The study site and tray nursing seedling experiment design are shown in **Figure 2**.

Image Acquisition and Preprocessing

Image Acquisition

When rice seedlings raising in the field, manual inspections by technicians were collected on 16 March, 19 March, and 24 March. In each region, the technicians sampled 100 seedlings and observed the growth stage of the seedlings. If there were more than 80 seedlings exhibited the same growth stage, this stage was recorded as the rice seedling age of the region. At the same time, the field images of this region were acquired using the DJI Phantom 4 RTK UAV with a 1-inch 20-megapixel CMOS (RGB) sensor. The images were collected with the lens shooting vertically downward, and the flight height was set to 3 m with a ground sampling distance (GSD) of 0.08 cm/pixel. The adjacent images along the flight direction overlapped on an average of one-third. The original image sizes were 5,472 × 3,642 pixels and the images were separately saved as TIFF files. On 16 March, there were less than 80 seedlings that exhibited a same growth stage in Region no 0.7 by manual inspection. Therefore, the field images of this region were excluded from the dataset. Details of RGB image acquisition and the corresponding phenological growth stages of the rice seedlings are shown in **Table 1**.

Image Preprocessing and Dataset Preparation

There is redundant information in the original RGB images acquired by the UAV, such as the road and the field. In addition, the large image size is not suitable for machine learning application. Hence, image preprocessing is necessary. Image preprocessing algorithm is mainly divided into three main steps (**Figure 1**). First, Gaussian filtering was adopted to reduce the noise after the image gray scale. Then, image enhancement is done before edge gradient detection. After that, the seedling raising regions were coarsely extracted based on the edge gradient detection, that is, RGB images of rice seedling canopy are extracted. Next, the images were cropped into different sizes, including 100 × 100 pixels, 200 × 200 pixels, 224 × 224 pixels, 300 × 300 pixels, 400 × 400 pixels, and 600 × 600 pixels. Finally, images contained redundant information were removed. The rest of the images were prepared as the datasets. Detailed information of the datasets is shown in **Table 2**.

HOG-SVM-Based Rice Seedling Growth Stages Detection

Histograms of oriented gradient feature was used to capture and express texture features of the seedlings canopy caused by different seedling growth stages. To extract the HOG feature, the extracted images were divided into uniformly spaced non-overlapping cells of $c \times c$ pixels (**Figure 3**, top). The image gradient orientation of each cell was binned and aggregated into local histograms. Dalal and Triggs (2005) found that using an unsigned gradient orientation (0–180°) and 9 bins performed better than a lower number of bins and a signed gradient orientation (0–360°) with an increased number of bins (up to 18 bins). Therefore, the histogram binning was performed using the unsigned gradient orientation and 9 bins in this work. The cells were grouped into overlapping blocks of $b \times b$ cells. As such, a single cell could be included in multiple blocks. The cell histograms in each block were normalized with respect to the entire block. The HOG feature was thus comprised of all the normalized histograms of the gradient orientations (**Figure 3**, bottom). The cell size c and block size b were optimized through a grid search during training of the classifiers, and the block overlap was fixed to half the block size rounded up to reduce the search space.

Support vector machine has been proved to be a powerful tool for problems of classification and regression for many previous studies, and thus, it was adopted to classify the seedling images according to their growth stages based on the extracted HOG features. The images were classified into three growth stages, including the BBCH11, BBCH12, and BBCH13. Since SVMs are inherently two-class classifiers, a set of binary one-verse-one classifiers are built, which train one learning model for each pair of classes. Linear, quadratic, cubic, medium Gaussian, coarse Gaussian, and fine Gaussian functions were employed and evaluated based on their accuracy on the validation dataset.

Deep Learning-Based Rice Seedling Growth Stages Detection

Deep learning is an important and new branch of machine learning. It originates from the artificial neural network, which learns the representation of data by constructing artificial neural network. Currently, the most widely used deep learning networks are CNNs. Efficientnet is a family of CNNs of similar architecture, which achieves more efficient results by uniformly scaling depth, width, and resolution with a scale ratio between these sets of parameters (Tan and Le, 2019). In this study, Efficientnets were adopted to classify the seedling growth stages. The performance of the proposed model was compared with the state-of-art CNN models such as VGG16, ResNet50, and DenseNet121.

Efficientnet

Recently, Tan and Le (2019) developed Efficientnet architectures, which were based on CNN design, and systematic model scaling technique was developed by applying a simple but effective compounded coefficient to scale up all depth, width, and resolution dimensions evenly. Tan and Le (2019) showed that the Efficientnet leads to superior performance and higher efficiency than the existing CNN methods both in terms of the number of parameters and Top1 accuracy when applied to the ImageNet dataset. Efficientnet family consists of eight models, ranging from B0 to B7. With the increase of the version, the performance of the models improves gradually, but the corresponding model size and calculation resource will not increase considerably. The main building block in Efficientnet is the mobile inverted bottleneck convolution (MBConv), which is initially introduced with MobileNetV2. The MBConv block receives two inputs, the first one is data and the second is arguments of the block. In addition, blocks consist of a layer that first expands the channels and then compresses them, thereby reducing the number of channels for the subsequent layer. A set of attributes, such as input filters, output filters, expansion rate, and compression rate, are used in the MBConv. The network parameters of EfficientnetB0 are shown in **Table 3**.

In this study, Efficientnet architectures were utilized to detect seedling growth stages to determine the best model. A total of two fully connected layers were added, 1,792 nodes for the inner layer and 3 nodes for the output layer (according to the number of predicted growth stages types). **Figure 4** shows the diagram of the EfficientnetB4 used to detect rice seedling growth stages.

Other State-of-Art Convolution Neural Network Models

VGG16 presented by Simonyan and Zisserman (2015), which won the ILSVRC 2014, is a CNN architecture with approximately 138 million parameters. It consists of 5 maximum pooling layers, 13 convolution layers, 3 full connection layers, and a softmax classifier layer. Instead of having large number of hyperparameters, VGG16 always has the same convolution layers that use 3 × 3 filters with stride 1 and same padding and maximum pooling layers that use 2 × 2 filters with stride 2. All hidden layers are added with ReLU layers. After the first and second fully connected layers, the dropout technology is also used to prevent network overfitting. The input layer takes images of 224 × 224 pixels.

ResNet50 presented by He et al. (2016), which won the ILSVRC-2015 competition in 2015, is an architecture proposed to solve the problem of gradient disappearance and degradation problem. The architecture of ResNet50 is based on many stacked residual units. Residual units are used as the building blocks to build the network. These units consist of convolution and pooling layers. This architecture uses 3 × 3 filters as VGG16 and takes input images of 224 × 224 pixels.

DenseNet (Huang et al., 2017) was presented and won the best paper on CVPR2017. It encourages feature reuse and alleviates the problem of vanishing gradient. It is characterized in that DenseNet connects each layer with every other layer in a feed-forward manner, that is, the feature maps of all the previous layers are used as inputs for each layer, and their feature maps are used in all subsequent layers as inputs. This architecture has a dense connectivity pattern, therefore called a dense convolutional neural network.

Transfer Learning

Transfer learning recycles previously trained networks using the new data to update a small part of the original weights, which makes the learning process more efficient. Given that sufficient public dataset for rice seedlings does not exist, it is difficult to obtain a satisfactory result based on the training deep learning model from scratch. Therefore, transfer learning technology (Weiss et al., 2016) was adopted in our model training. First, to obtain the pretrained network, the Efficientnets are pretrained on ImageNet, which is currently the largest image recognition dataset in the world, with 1.2 million images of 1,000 categories. Then, the seedling images are loaded into the pretrained Efficientnets. Second, the last few layers of the trained network can be removed, and two new fully connected layers are built and retrained for the growth stage classification task (**Figure 4**). In the transfer learning approach, using the knowledge of the network previously trained with large amounts of visual data in a new task is very advantageous in terms of saving time and achieving high accuracy compared to training the model from scratch.

Training and Testing

In the HOG-SVM-based machine learning algorithm, 1,000 images in each growth stage, a total of 3,000 images, were randomly selected to form the basis dataset for SVM classifications. Then, the datasets in each growth stage were randomly shuffled and divided into training, validation, and test sets according to the ratio of 6:2:2. The HOG features have two hyperparameters, which were the cell size $c \in \{8, 16, 32\}$ and the block size $b \in \{2, 3, 4, 5, 6\}$. The two hyperparameters were optimized through a grid search on the training set by training a multiclass SVM for each combination and evaluating it on the validation set. Different SVM kernels and the sizes of input image were also evaluated for each of the SVMs trained in the grid searches. In the grid search of each HOG features, the combination of the kernel function and the input image size with the best performance was selected as the optimal kernel function and input image size. Afterward, the highest accuracy on the validation set was used to select the HOG hyperparameters.

In the deep learning classification, Efficientnets are adopted to perform the classification task of rice seedling growth stage, and then, the state-of-art CNN models are considered and compared with the best performance of the Efficientnets. **Table 2** formed the basis dataset for deep learning-based growth stage classification. The dataset in each growth stage was divided into training, validation, and test sets according to the ratio of 6:2:2. According to the different requirements

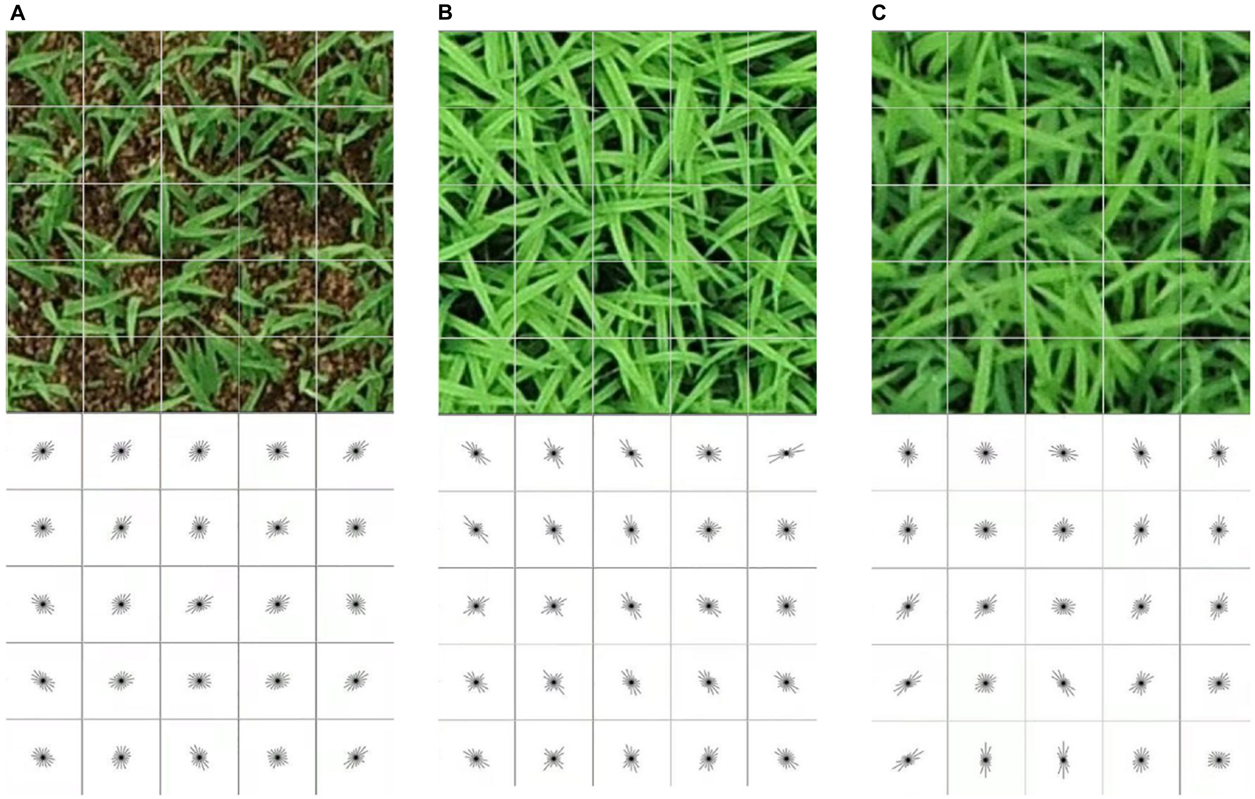


FIGURE 3 | Visualization of the extracted HOG features of different seedling growth stages using a cell size of 40 × 40 and block size of 5 × 5 on image of 200 × 200 pixels. Top row, RGB images; bottom row, HOG features of rice seedlings: **(A)** BBCH11, **(B)** BBCH12, and **(C)** BBCH13.

TABLE 3 | Parameters of the EfficientnetB0 network.

Stage (i)	Operator (F_i)	Resolution ($H_i \times W_i$)	Channels (C_i)	Layers (L_i)
1.	Conv3 × 3	224 × 224	32	1
2.	MBConv1, k3 × 3	112 × 112	16	1
3.	MBConv6, k3 × 3	112 × 112	24	2
4.	MBConv6, k5 × 5	56 × 56	40	2
5.	MBConv6, k3 × 3	28 × 28	80	3
6.	MBConv6, k5 × 5	14 × 14	112	3
7.	MBConv6, k5 × 5	14 × 14	192	4
8.	MBConv6, k3 × 3	7 × 7	320	1
9.	Conv1 × 1 & Pooling & FC	7 × 7	1,280	1

of input image sizes of deep learning models, images of 224 × 224 pixels were prepared for the EfficientnetB0. Then, they were resized to 240 × 240 pixels and 260 × 260 pixels, which were used for EfficientnetB1 and EfficientnetB2, respectively. Images of 300 × 300 pixels were prepared for EfficientnetB3. Similarly, images of 400 × 400 pixels were resized to 380 × 380 pixels and 456 × 456 pixels and then were fed into EfficientB4 and EfficientnetB5, respectively. Images of 600 × 600 pixels were used for EfficientnetB7 and then were resized to 528 × 528 pixels and used for EfficientnetB6. Images of 224 × 224 pixels were used for VGG16, ResNet50, and DenseNet121.

Our study was conducted in Windows 10 environment (processor: Intel core i9 10920X CPU; memory: 64G; graphics card: GeForce RTX 2080Ti 11G DDR6). Python3.8 was selected for image preprocessing, whereas the feature extraction and analysis were performed in MATLAB (version 2020b, the MathWorks, Inc., Natick, Massachusetts, United States) using the Computer Vision System Toolbox 9.3 and the Classification Learner App from the Statistical and Machine Learning Toolbox 12.0. The deep learning frameworks Pytorch1.8.1 and Python3.7, in combination with Cuda10.2, were used for deep learning model training. In the experiment design and training process of deep learning models, the initial learning

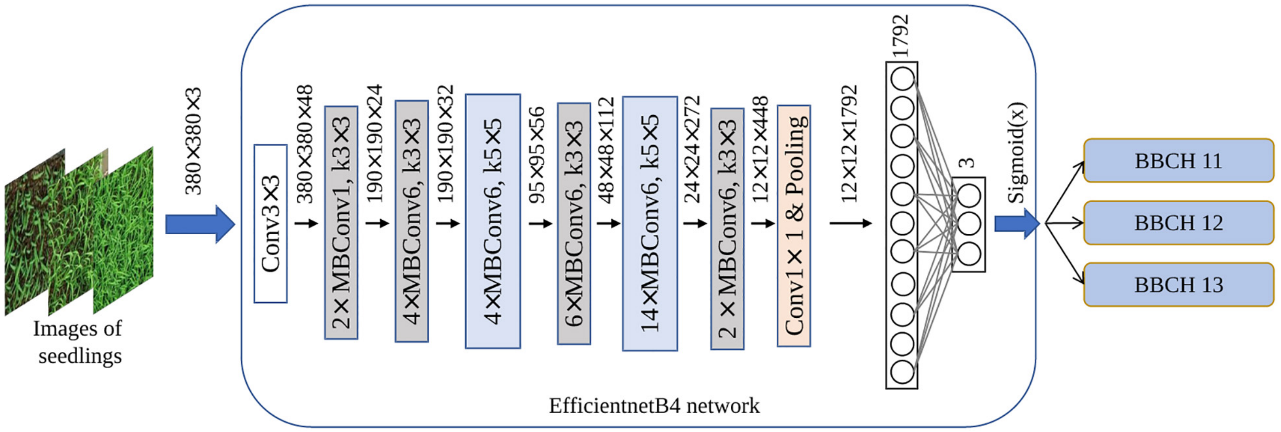


FIGURE 4 | Diagram of the EfficientnetB4 used to detect rice seedling growth stages.

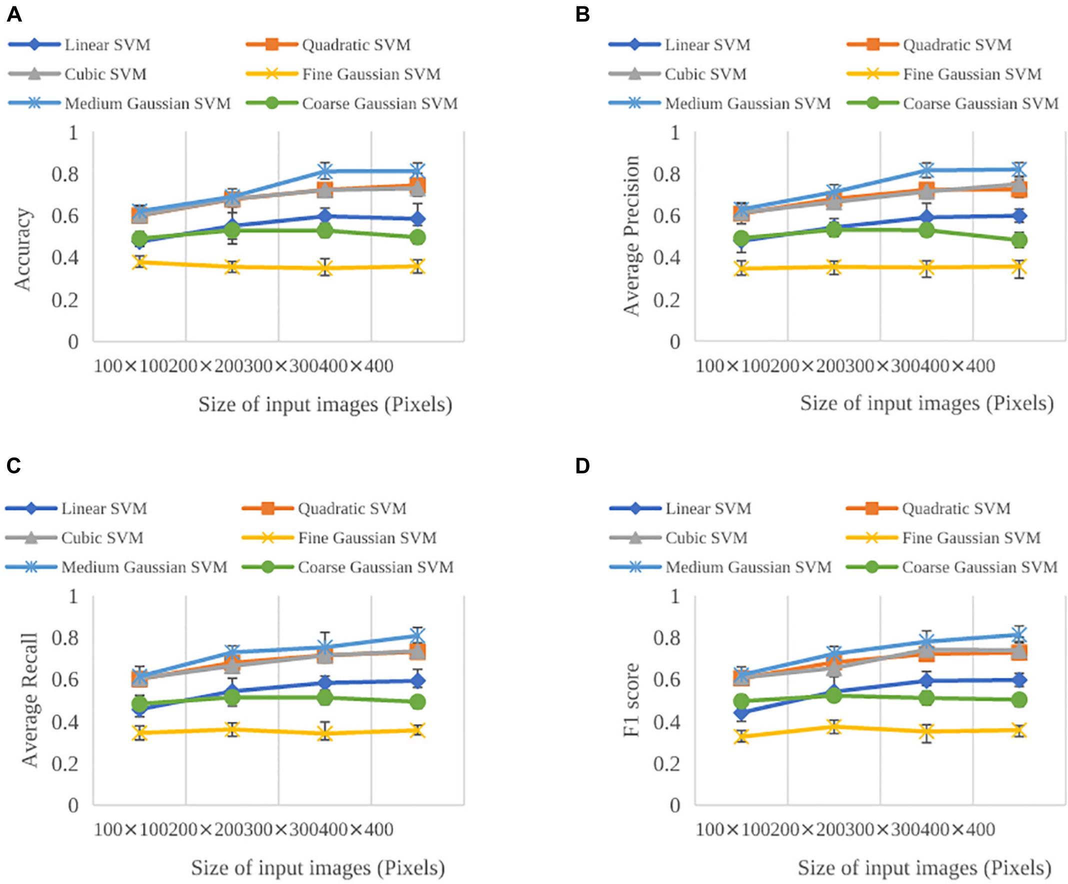


FIGURE 5 | Evaluation performance on the validation sets for SVMs trained on the HOG features as a function of SVM kernels and input image sizes: **(A)** Accuracy, **(B)** average precision, **(C)** average recall, and **(D)** F1 score. Error bars show the standard deviation across the SVMs.

rate was set to 0.001, and the network batch size of the training set and validation set was set to 32. Adam optimization algorithm was selected in this work. The epoch of network model was set to 50.

Performance Evaluation

In this study, the performance of machine learning algorithms was evaluated using four evaluation indexes of accuracy, precision, recall, and F1 score, which were given by the equations

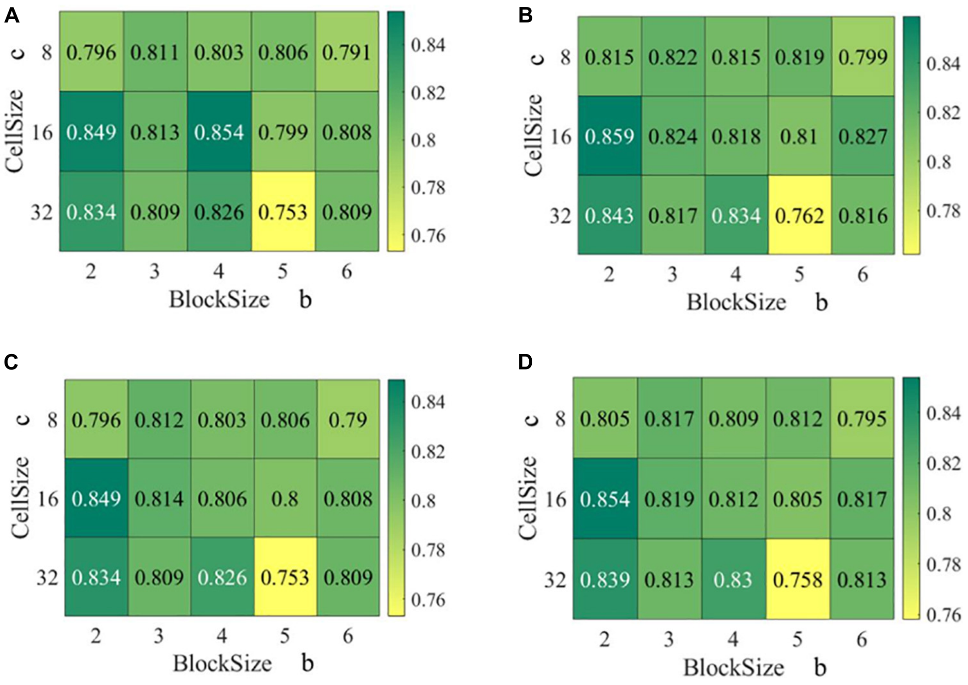


FIGURE 6 | Evaluation performance on the validation sets for SVMs trained on the HOG features as a function of cell size *c* and block size *b*: **(A)** Accuracy, **(B)** average precision, **(C)** average recall, and **(D)** F1 score.

(1)-(4). The accuracy indicates the rate of correctly classified images out of all the images in a test set for a particular growth stage class, which shows the overall effectiveness of the classifier. The precision represents the proportion of images that are true positive among all images predicted to be positive. The recall represents the proportion of images predicted to be positive among the images that are true positive. The values of four evaluation indexes range from 0 to1. The higher the value is, the better the efficiency of the algorithm is.

$$accuracy \quad \frac{\sum correctly\ classified\ images}{\sum images} \tag{1}$$

$$precision\,(GS) \quad \frac{\sum images\ with\ GS\ classified\ as\ GS}{\sum images\ classified\ as\ GS} \tag{2}$$

$$recall\,(GS) \quad \frac{\sum images\ with\ GS\ classified\ as\ GS}{\sum images\ with\ GS} \tag{3}$$

$$F_1 \quad \frac{2\ \ precision\ \ recall}{precision + recall} \tag{4}$$

where *GS* is the growth stage: "BBCH11," "BBCH12," or "BBCH13."

TABLE 4 | Confusion matrix for SVM using HOG feature with cell size of 16 and block size of 2 evaluated on the test set.

		Predicted			
		BBCH11	**BBCH12**	**BBCH13**	**Recall**
Observed	BBCH11	172	20	9	85.57%
	BBCH12	5	177	19	88.06%
	BBCH13	0	38	163	81.09%
Precision		97.18%	75.32%	85.34%	84.91%

The lower right cell shows the accuracy.

RESULTS AND ANALYSIS

Results of HOG-SVM-Based Rice Seedling Growth Stages Detection

A total of six SVM kernels and four input image sizes were first considered. Six kernels included linear, quadratic, cubic, medium Gaussian, coarse Gaussian, and fine Gaussian whereas four input image sizes included 100 × 100 pixels, 200 × 200 pixels, 300 × 300 pixels, and 400 × 400 pixels. **Figure 5** shows the performance evaluation of SVM classifiers with different kernels and input image sizes. For different kernels, medium Gaussian kernel resulted in the best in accuracy, average precision, average recall, and F1 score. The fine Gaussian kernel obtained the poorest results. Moreover, when input image size was selected as 400 × 400 pixels, the best performance was achieved, with accuracy, average precision, average recall, and F1 score of 84.91, 85.958, 84.90, and 85.43%, respectively. Therefore, the medium

TABLE 5 | Evaluation performance on test sets for HOG-SVM classifiers with different numbers of training images in each growth stage.

Input image size (pixels)	Number of training images in each growth stage	Accuracy	Average precision	Average recall	F1 score	Time of training (min)	Time of testing (sec)
400 × 400	1,000	84.9%	85.9%	84.9%	85.4%	3.03	12.374
400 × 400	1,500	81.0%	83.3%	81.1%	82.2%	4.16	35.741
400 × 400	2,000	81.4%	83.0%	81.4%	82.2%	7.24	52.320
400 × 400	2,500	79.8%	80.8%	79.8%	80.3%	11.6	76.156

TABLE 6 | Evaluation performance on the validation sets for EfficientnetB0-B7.

Models	Accuracy	Average precision	Average recall	F1 score	Time of training (min)	Time of test (sec)
EfficientnetB0	97.67%	97.54%	97.71%	97.62%	106.7	32
EfficientnetB1	97.52%	97.54%	97.75%	97.64%	40.9	12
EfficientnetB2	97.61%	97.61%	97.79%	97.70%	36.5	12
EfficientnetB3	98.43%	98.41%	98.58%	98.50%	50.0	15
EfficientnetB4	99.47%	99.53%	99.39%	99.46%	62.5	19
EfficientnetB5	98.78%	99.09%	99.01%	99.05%	83.2	24
EfficientnetB6	98.28%	98.60%	98.46%	98.53%	135.0	40
EfficientnetB7	98.72%	98.85%	98.74%	98.79%	220.9	66

Gaussian kernel and image sizes of 400 × 400 pixels were chosen in the further analysis of the HOG features.

After selecting the optimal SVM kernel and input image size, the HOG feature hyperparameter grid search was performed by training individual SVM classifiers on the training set and subsequently evaluating the classifiers on the validation set and test set. Cell sizes *c* of 8, 16, and 32 pixels as well as block sizes *b* of 2, 3, 4, 5, and 6 cells were evaluated. Using the medium Gaussian kernel and the input image size of 400 × 400 pixels, the accuracy, average precision, average recall, and F1 score across the different cell size *c* and block size *b* varied from 75.3 to 85.4, 76.2 to 85.9, 75.3 to 84.9, and 75.8 to 85.4%, respectively (**Figure 6**). Compared with the four evaluation indexes, they showed similar trends with respect to cell size *c* or block size *b*. However, when inspected the evaluation indexes separately, they showed no clear trends with respect to cell size *c* or block size *b*.

The HOG feature with a cell size of 16 and a block size of 4 resulted in the highest accuracy of 85.4%, and the second highest accuracy of 84.9% was found in HOG feature with a cell size of 16 and block size of 2. Moreover, HOG feature with cell size of 16 and block size of 2 resulted in the highest average precision, average recall, and F1 score. Therefore, the cell size of 16 and block size of 2 were chosen as the optimal parameters.

In addition, the SVM classifier trained with the HOG feature with a cell size of 16 and a block size of 2 was evaluated on the test set. **Table 4** shows the corresponding confusion matrix. A high precision (97.18%) and an adequate precision (85.34%) were found in BBCH11 and BBCH13, respectively, whereas the BBCH12 group showed an inadequate precision (75.32%). On the other hand, the recall rates in each growth stage group showed adequate performance, which achieved above 81%. Besides, BBCH12 and BBCH13 overlapped the most, which indicated that it was difficult to distinguish between BBCH12 and BBCH13.

TABLE 7 | Confusion matrix for EfficientnetB4 evaluated on the test set.

		Predicted			
		BBCH11	**BBCH12**	**BBCH13**	**Recall**
Observed	BBCH11	563	0	0	100
	BBCH12	3	696	10	98.17%
	BBCH13	0	0	1,148	100%
Precision		99.47%	100%	99.14%	99.47%

The lower right cell shows the accuracy.

To further verify the robustness of the HOG-SVM model, different numbers of images were used to train the SVM model, and 1,000 images, 1,500 images, 2,000 images, and 2,500 images in each growth stage were randomly selected and formed the classification datasets. Training, validation, and test sets were divided according to the ratio of 6:2:2. Medium Gaussian kernel and HOG feature of cell size of 16 and block size of 2 were used. The performance of SVMs is shown in **Table 5**. The accuracy, average precision, average recall, and F1 score of different number of training images varied from 79.8 to 84.9, 80.8 to 85.9, 79.8 to 84.9, and 80.3 to 85.4%, respectively. As the number of training image increased, the performance of the SVM classifiers dropped slightly. However, the lowest F1 score was above 80%, which indicated that the overall performance was reasonably robust.

Results of Deep Learning-Based Rice Seedling Growth Stages Detection

Results of Rice Seedling Growth Stages Detection Based on Efficientnet

All Efficientnets B0-B7 were trained on the training set and then validated and tested on validation set and test set, respectively. The performance evaluations of accuracy, average precision,

TABLE 8 | Evaluation performance on the validation sets for different CNN models.

	Accuracy	Average precision	Average recall	F1 score	Total training time (min)	Total test time (sec)
EfficientnetB4	99.47%	99.53%	99.39%	99.46%	62.5	19
Densenet121	99.06%	98.79%	99.11%	98.95%	114.2	35
Resnet50	98.97%	98.74%	98.92%	98.83%	104.2	30
VGG16	94.84%	94.55%	94.83%	94.69%	116.7	33

average recall, and F1score obtained from all Efficientnet models on validation datasets are provided in **Table 6**. For all eight models, classification results showed good performance. Accuracies were recorded in the range of 97.52–99.47%, whereas the average precision, average recall, and F1 score varied in the ranges of 97.54 to 99.53, 97.71 to 99.39, and 97.62 to 99.46%, respectively. The classification accuracy got better as the version of Efficientnet increased; however, there were slight decreases after EfficientnetB4. **Table 6** shows that EfficientnetB4 outperformed other Efficientnet models and achieved the best values in four evaluation indexes. **Table 7** shows the confusion matrix of EfficientnetB4 evaluated on the test datasets. The EfficientnetB4 showed satisfactory results. Precision and recall in each growth stage got height values, which were above 98.17%. Among them, precision in BBCH12 and recall in BBCH11 and BBCH13 obtained 100%. The classifier incorrectly recognized 3 and 10 out of 709 images (0.4 and 1.4%) of BBCH12 as BBCH11 and BBCH13, respectively. Thus, the recall rate in BBCH12 was less lower than the BBCH11 and BBCH13.

The precision-recall curves plot the precision rate against the recall rate. The under-area values of precision–recall curves indicate the reliability of the model from 0 to 1. The under-area values close to 1 indicates that the model can differentiate multiple classes with higher accuracy; otherwise, the smaller under-area values are, the poorer performance of the model suffers when distinguishing classes. It can be seen from **Figure 7A** that the under area of precision–recall curve of EfficientnetB4 is the largest, which indicates that it performs the best.

In terms of processing time of Efficientnets, as the version of Efficientnet increased from B1 to B7, the time consumption of training and test increased from 40.9 to 220.9 min, 12 to 66 s, respectively.

Comparison With Other State-of- Art Deep Learning Models

To verify the effectiveness of the EfficientnetB4, three other popular CNN models, VGG16, Resnet50, and Densenet121, were trained for rice seedling growth stage recognition and classification and compared with the EfficientnetB4. The mentioned CNN models were trained on the same dataset (with same hardware configuration) that were used in Efficientnet. **Table 8** and **Figure 7B** show the performance comparison of the four models. As we can see in the table, the performance of the EfficientnetB4 achieved the best results in terms of accuracy, average precision, average recall, and F1 score. Densenet121 is close to the EfficientnetB4 model. VGG16 presented the lowest accuracy value of 94.84%. The validation results revealed that overall EfficientnetB4 performed better than the other three CNN models. In terms of processing time of popular CNNs, the time consumption of training and test of Efficientnet B4 was the lowest.

DISCUSSION AND CONCLUSION

Comparison With Traditional Machine Learning Algorithms and Deep Learning Algorithms

In this paper, automatic approaches of rice seedling growth stages recognition and classification have been presented using both the traditional machine learning algorithm and deep learning algorithm. Compared with HOG-SVM-based algorithm, the performance of deep learning algorithm far exceeds that of the traditional machine learning in growth stages classification. For instance, as the best deep learning models, the EfficientnetB4 achieved the best performance, with 99.47, 99.53, 99.39, and 99.46% in accuracy, average precision, average recall, and F1 score, respectively. Meanwhile, the best HOG-SVM model obtained the performance with 84.9, 85.9, 84.9, and 85.4% in accuracy, average precision, average recall, and F1 score, respectively. In **Tables 4**, **7**, we can notice from the confusion matrix that errors in HOG-SVM algorithm and the EfficientnetB4 mostly occur on adjacent growth stages. These are situations where even human eyes that inspect from the canopy of the seedling can have uncertainty to decide the exact growth stages from one stage to the next one. Remaining errors are low and can thus be considered as reasonable errors.

The construction of multiple layers for automatically image features learning from training data instead of complex manual feature extraction contributes to high performance of the deep learning algorithms. The phase of manual feature extraction in traditional machine learning is affected to a greater or lesser extent by many other factors and thus can sometimes result in low prediction performance. In the HOG-SVM-based rice seedling growth stage recognition, HOG features, consisting of the orientation of edges found through the computation of the image gradient, are manually selected to describe the texture feature of the seedling canopy structure. The hyperparameters of HOG, the cell size and block size, affect the number of occurrences of edges within given orientation ranges that constitute a locally spaced histogram and thus have effects on the classification performance. In addition, it can be noticed that the performance of the SVM model varies more obviously than the EfficientnetB4 does as the number of training images increases. In **Tables 5**, **9**, as the

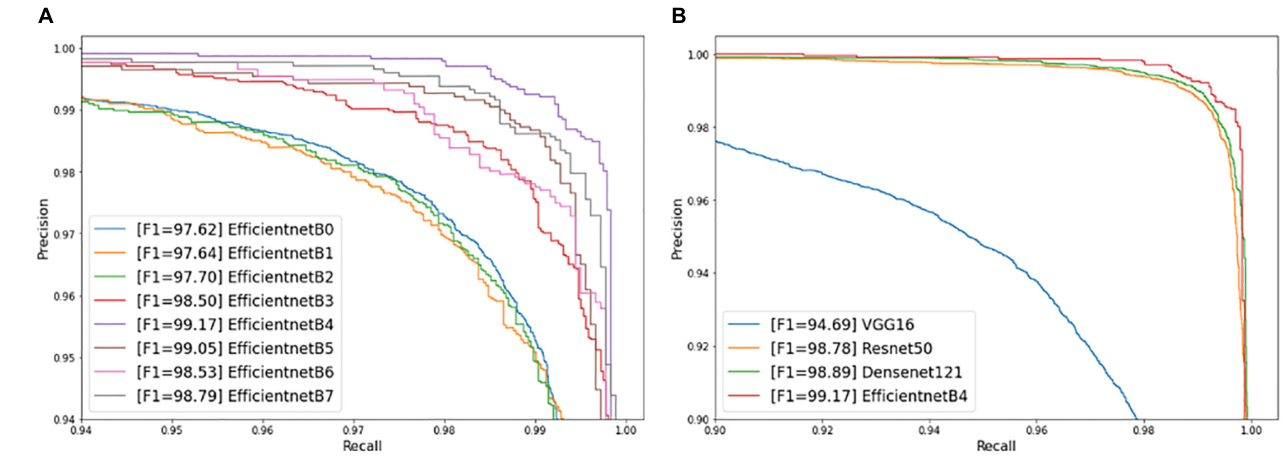


FIGURE 7 | Precision–Recall curves of different deep learning models: **(A)** EfficientnetB0-B7 **(B)** State-of-art CNN models.

TABLE 9 | Evaluation performance on test sets for EfficientnetB4 with different numbers of training images in each growth stage.

Model	Number of training images in each growth stage	Accuracy	Average precision	Average recall	F1 score	Time of training (min)	Time of testing (sec)
EfficientnetB4	1,000	98.17	98.20	98.17	98.18	15.8	5
EfficientnetB4	2,000	98.75	98.78	98.75	98.77	30.8	9
EfficientnetB4	2,500	99.15	99.15	99.16	99.16	45.0	13

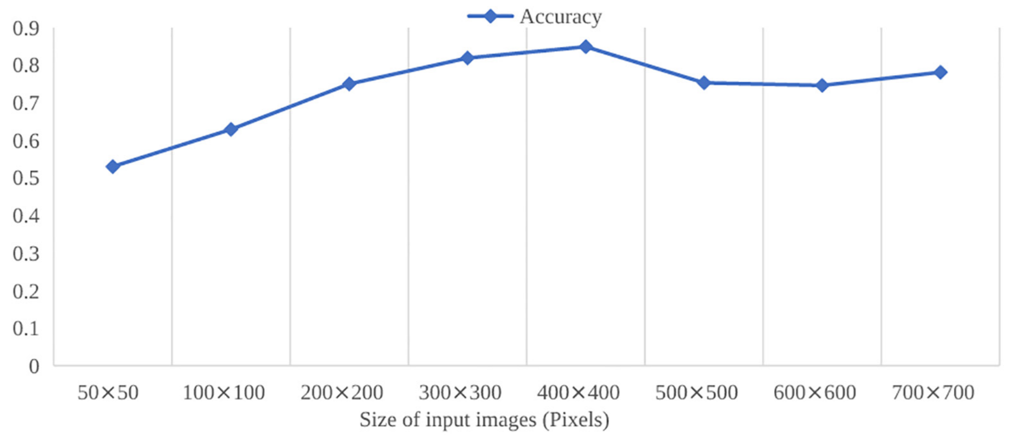


FIGURE 8 | Effects of different input image size on HOG-SVM classification.

number of training images in each growth stage increases from 1,000 to 2,500, the accuracy of HOG-SVM dropped from 84.9 to 79.8%, whereas the accuracy of EfficientnetB4 increased from 98.17 to 99.15%. Furthermore, the size of input images had effect on the performance of HOG-SVM classification. In **Figure 8**, the performance of classification shows the difference as the size of input images varies. The accuracy rises quickly with small input image sizes. When the input image size reaches 400 × 400 pixels, the highest value is obtained. However, the accuracy drops slightly as the image size becomes larger.

On the other hand, the computational and processing time is a crucial aspect in machine learning algorithms. Compared to the processing time presented in **Tables 5**, **6**, the training time of deep learning models took much longer than the HOG-SVM models. Training time of deep learning models mostly depends on the number of images used, the batch size, the learning rate, and the hardware used, among other factors. In the aspect of test time, deep learning models were faster than the HOG-SVM models. When it comes to practical application, researchers pay more attention to the test time.

In general, deep learning models exhibit satisfactory performance in rice seedling growth stage recognition. Furthermore, the datasets of rice seedling in three growth stages presented in this paper differed in genotype, sowing density, and sowing dates. A total of three cultivars and five nursing seedling densities are included in the dataset, which constitute a comprehensive seedling phenotyping. From this point of view, the traditional machine learning algorithms show reasonable discriminative ability in rice seedling growth stages.

Limitation and Further Applications

The presented results show that the machine learning algorithms are robust on rice genotypes, sowing density, and sowing dates. However, crop phenotyping based on UAV images is also sensitive to sensor-target angles, overlap among leaves, and field conditions. In our study, all images were acquired by UAV from a vertical downward angle at a height of 3 m, producing images with similar statistical structure. To make the machine learning algorithms broadly useful across many situations, a variety of reasonable flight heights and resultant image resolutions are needed to take into account. Additionally, the minimum required image resolution (i.e., maximum flight height) that delivers quality results should be determined, because a higher flight height would allow the data to be collected more rapidly.

It has been previously reported that computer vision and machine learning techniques can help to identify the growth stages of individual seedling (Yu et al., 2013; Samiei et al., 2020). However, the issue of plant overlapping each other would decrease the detection accuracy and become a limitation. Rice is densely planted crop, and few studies have been carried out to recognize the growth stages of rice seedling. This study investigated the feasibility of developing machine learning algorithms for rice seedling growth stages detection with different canopy phenotypes. Rich information automatically learned or extracted from canopy phenotype (structural and textural information) makes it possible for the machine learning-based data analytics to achieve decision-making in a way much closer to how human brains work. It will be interesting to extend the approach to a range of crops of agricultural interest, such as oat, wheat, and sorghum, to investigate quantitatively how, by similarity in shape of different crops, the knowledge learned on rice seedlings could be transferred to others *via* transfer learning. Moreover, during the whole growth cycle, more fine growth stages, not only stages of seedlings but also stages after seedling transplantation, could also be added to extend the investigation of crop growth stage discriminative ability of machine learning algorithms.

Conclusion

Recognizing rice seedling growth stages to timely do field operations, such as temperature control, fertilizer, irrigation, cultivation, and disease control, is of great significance of crop management, provision of standard and well-nourished seedlings for mechanical transplanting, and increase of yield. Specifically, when raising rice seedlings in paddy field, it is inefficient to manually inspect on growth stages, and environmental factors, such as rain, solar radiation have great impact on the growth stage variation. Thus, timely recognizing rice seedling growth stages become more and more important. In this study, automatic approaches using machine learning algorithms on UAV images were developed to determine three key growth stages of rice seedling, BBCH11, BBCH12, and BBCH13. In the traditional machine learning algorithm, HOG was selected as the texture feature to represent the canopy structure of the seedlings and combine with SVM classifier to recognize the growth stages. The best HOG-SVM showed reasonable discriminative ability in the classification task. Compared with the HOG-SVM algorithm, the deep learning algorithms showed outstanding performance in detection of seedling growth stages. Generally speaking, the machine learning algorithms proposed in this paper could be used to estimate the growth stages of rice seedlings in the BBCH11 to BBCH13, and they provide a basis for timely seedling supplements and subsequent crop management. Future research should include experiments employing more cultivars, different crops, and more growth stages recognition and investigate other factors to further verify and optimize the algorithms in this paper.

DATA AVAILABILITY STATEMENT

The datasets, models and code used in this manuscript are available at the following locations: Datasets and models: https://drive.google.com/drive/folders/1AY-ro3HID9nodrk2aIcTmGgjCzTggkks?usp=sharing. Code: https://github.com/imagevision-lab/rice_seedling_growth_stages_detection.

AUTHOR CONTRIBUTIONS

ST: conceptualization, field data acquisition, data curation, formal analysis, investigation, methodology, software, validation, visualization, and writing—original draft, review, and editing. JL: data curation, validation, software, and writing—original draft. HL: data curation, formal analysis, software, and visualization. ML: data curation, formal analysis, writing—original draft and visualization. JY and GL: data curation and visualization. YW: data curation and field data acquisition. ZL and LQ: investigation, methodology, project administration, and supervision. XM: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, visualization, and writing—review and editing. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Research and Development Projects in Key Areas of Guangdong Province (grant no. 2019B020221003), the National Natural Science Foundation of China (grant no. 52175226), the National Key Research and Development Program of China Subproject (grant no. 2021YFD2000702), and the Natural Science Foundation of Guangdong Province (grant no. 2020A1515010634).

REFERENCES

Abid, M., Khan, I., Mahmood, F., Ashraf, U., Imran, M., and Anjum, S. A. (2015). Response of hybrid rice to various transplanting dates and nitrogen application rates. *Philipp. Agric. Sci.* 98, 98–104.

Abouzahir, S., Sadik, M., and Sabir, E. (2021). Bag-of-visual-words-augmented histogram of oriented gradients for efficient weed detection. *Biosyst. Eng.* 202, 179–194. doi: 10.1016/j.biosystemseng.2020.11.005

Bai, X., Cao, Z., Zhao, L., Zhang, J., Lv, C., Li, C., et al. (2018). Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. *Agric. For. Meteorol.* 259, 260–270. doi: 10.1016/j.agrformet.2018.05.001

Biswas, J. C., Ladha, J. K., Dazzo, F. B., Yanni, Y. G., and Rolfe, B. G. (2000). Rhizobial inoculation influences seedling vigor and yield of rice. *Agron. J.* 92, 880–886. doi: 10.2134/agronj2000.925880x

Chen, R., Chu, T., Landivar, J. A., Yang, C., and Maeda, M. M. (2018). Monitoring cotton (*Gossypium hirsutum* L.) germination using ultrahigh-resolution UAS images. *Precis. Agric.* 19, 161–177. doi: 10.1007/s11119-017-9508-7

Cheng, S. R., Ashraf, U., Zhang, T. T., Mo, Z. W., Kong, L. L., Mai, Y. X., et al. (2018). Different seedling raising methods affect characteristics of machine-transplanted rice seedlings. *Appl. Ecol. Environ. Res.* 16, 1399–1412. doi: 10.15666/aeer/1602_13991412

Chowdhury, M. E. H., 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

Chu, Z., and Yu, J. (2020). An end-to-end model for rice yield prediction using deep learning fusion. *Comput. Electron. Agric.* 174:105471. doi: 10.1016/j.compag.2020.105471

Cutler, J. M., Steponkus, P. L., Wach, M. J., and Shahan, K. W. (1980). Dynamic aspects and enhancement of leaf elongation in rice. *Plant Physiol.* 66, 147–152. doi: 10.1104/pp.66.1.147

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection", in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005*, San Diego, CA, 886–893. doi: 10.1109/CVPR.2005.177

Farman, H., Ahmad, J., Jan, B., Shahzad, Y., Abdullah, M., and Ullah, A. (2022). Efficientnet-based robust recognition of peach plant diseases in field images. *Comput. Mater. Continua* 71, 2073–2089. doi: 10.32604/cmc.2022.018961

Han, J., Shi, L., Yang, Q., Huang, K., Zha, Y., and Yu, J. (2021). Real-time detection of rice phenology through convolutional neural network using handheld camera images. *Precis. Agric.* 22, 154–178. doi: 10.1007/s11119-020-09734-2

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

Hiraoka, H., Nishiyama, I., and Suzuki, Y. (1987). Thermal reaction in growth of rice plants at the vegetative growth stage-comparison among different ecotypical groups. *Jpn. J. Crop Sci.* 56, 302–312. doi: 10.1626/jcs.56.302

Huang, G., Liu, Z., and Weinberger, K. Q. (2017). "DenseNet: densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2261–2269. doi: 10.1109/CVPR.2017.243

Jin, X., Che, J., and Chen, Y. (2021). Weed identification using deep learning and image processing in vegetable plantation. *IEEE Access* 9, 10940–10950. doi: 10.1109/ACCESS.2021.3050296

Kanno, K., Mae, T., and Makino, A. (2009). High night temperature stimulates photosynthesis, biomass production and growth during the vegetative stage of rice plants. *Soil Sci. Plant Nutr.* 55, 124–131. doi: 10.1111/j.1747-0765.2008.00343.x

Kargbo, M. B., Pan, S., Mo, Z., Wang, Z., Luo, X., Tian, H., et al. (2016). Physiological basis of improved performance of super rice (*Oryza sativa*) to deep placed fertilizer with precision hill-drilling machine. *Int. J. Agric. Biol.* 18, 797–804. doi: 10.17957/IJAB/15.0173

Lancashire, P. D., Bleiholder, H., van den Boom, T., Langelüddeke, P., Stauss, R., Weber, E., et al. (1991). A uniform decimal code for growth stages of crops and weeds. *Ann. Appl. Biol.* 119, 561–601. doi: 10.1111/j.1744-7348.1991.tb04895.x

Li, H., Li, Z., Dong, W., Cao, X., Wen, Z., Xiao, R., et al. (2021). An automatic approach for detecting seedlings per hill of machine-transplanted hybrid rice utilizing machine vision. *Comput. Electron. Agric.* 185:106178. doi: 10.1016/j.compag.2021.106178

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. (2018). Machine learning in agriculture: a review. *Sensors* 18:2674. doi: 10.3390/s18082674

Makino, A., Nakano, H., and Mae, T. (1994). Effects of growth temperature on the responses of ribulose-1,5-bisphosphate carboxylase, electron transport components, and sucrose synthesis enzymes to leaf nitrogen in rice, and their relationships to photosynthesis. *Plant Physiol.* 105, 1231–1238. doi: 10.1104/pp.105.4.1231

Maruyama, S., and Nakamura, Y. (1997). Photosynthesis, dark respiration and protein synthesis of rice leaves at low temperature – analysis of ribulose-1,5-bisphosphate carboxylase. *Jpn. J. Crop Sci.* 66, 85–91. doi: 10.1626/jcs.66.85

Nagai, T., and Makino, A. (2009). Differences between rice and wheat in temperature responses of photosynthesis and plant growth. *Plant Cell Physiol.* 50, 744–755. doi: 10.1093/pcp/pcp029

Ohsumi, A., Furuhata, M., and Matsumura, O. (2012). Varietal differences in biomass production of rice early after transplanting at low temperatures. *Plant Prod. Sci.* 15, 32–39. doi: 10.1626/pps.15.32

Pandey, P., Dakshinamurthy, H. N., and Young, S. N. (2021). Frontier: autonomy in detection, actuation, and planning for robotic weeding systems. *Trans. ASABE* 64, 557–563. doi: 10.13031/TRANS.14085

Pasuquin, E., Lafarge, T., and Tubana, B. (2008). Transplanting young seedlings in irrigated rice fields: early and high tiller production enhanced grain yield. *Field Crops Res.* 105, 141–155. doi: 10.1016/j.fcr.2007.09.001

Rasti, S., Bleakley, C. J., Silvestre, G. C. M., Holden, N. M., Langton, D., and O'Hare, G. M. P. (2021). Crop growth stage estimation prior to canopy closure using deep learning algorithms. *Neural Comput. Appl.* 33, 1733–1743. doi: 10.1007/s00521-020-05064-6

Ruiz-Sánchez, M., Aroca, R., Muñoz, Y., Polón, R., and Ruiz-Lozano, J. M. (2010). The arbuscular mycorrhizal symbiosis enhances the photosynthetic efficiency and the antioxidative response of rice plants subjected to drought stress. *J. Plant Physiol.* 167, 862–869. doi: 10.1016/j.jplph.2010.01.018

Samiei, S., Rasti, P., Ly Vu, J., Buitink, J., and Rousseau, D. (2020). Deep learning-based detection of seedling development. *Plant Methods* 16:103. doi: 10.1186/s13007-020-00647-9

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA.

Tan, K., Lee, W. S., Gan, H., and Wang, S. (2018). Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. *Biosyst. Eng.* 176, 59–72. doi: 10.1016/j.biosystemseng.2018.08.011

Tan, M., and Le, Q. V. (2019). "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML*, San Diego, CA.

Tan, S., Ma, X., Mai, Z., Qi, L., and Wang, Y. (2019). Segmentation and counting algorithm for touching hybrid rice grains. *Comput. Electron. Agric.* 162, 493–504. doi: 10.1016/j.compag.2019.04.030

Taylor, S. D., and Browning, D. M. (2022). Classification of daily crop phenology in phenocams using deep learning and hidden markov models. *Remote Sens.* 14:286. doi: 10.3390/rs14020286

Tong, J. H., Li, J. B., and Jiang, H. Y. (2013). Machine vision techniques for the evaluation of seedling quality based on leaf area. *Biosyst. Eng.* 115, 369–379. doi: 10.1016/j.biosystemseng.2013.02.006

Velumani, K., Madec, S., de Solan, B., Lopez-Lozano, R., Gillet, J., Labrosse, J., et al. (2020). An automatic method based on daily in situ images and deep learning to date wheat heading stage. *Field Crops Res.* 252:107793. doi: 10.1016/j.fcr.2020.107793

Weiss, K., Khoshgoftaar, T. M., and Wang, D. D. (2016). A survey of transfer learning. *J. Big Data* 3:9. doi: 10.1186/s40537-016-0043-6

Yang, Q., Shi, L., Han, J., Zha, Y., and Zhu, P. (2019). Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images. *Field Crops Res.* 235, 142–153. doi: 10.1016/j.fcr.2019.02.022

Yu, Z., Cao, Z., Wu, X., Bai, X., Qin, Y., Zhuo, W., et al. (2013). Automatic image-based detection technology for two critical growth stages of maize: emergence and three-leaf stage. *Agric. For. Meteorol.* 174, 65–84. doi: 10.1016/j.agrformet.2013.02.011

Zhang, Y., Zhong, W., and Pan, H. (2021). Identification of stored grain pests by modified residual network. *Comput. Electron. Agric.* 182:105983. doi: 10.1016/j.compag.2021.105983

Zhao, B., Zhang, J., Yang, C., Zhou, G., Ding, Y., Shi, Y., et al. (2018). Rapeseed seedling stand counting and seeding performance evaluation at two early growth stages based on unmanned aerial vehicle imagery. *Front. Plant Sci.* 9:1362. doi: 10.3389/fpls.2018.01362

Zhao, L., Guo, W., Wang, J., Wang, H., Duan, Y., Wang, C., et al. (2021). An efficient method for estimating wheat heading dates using uav images. *Remote Sens.* 13:3067. doi: 10.3390/rs13163067

Zhu, Y., Cao, Z., Lu, H., Li, Y., and Xiao, Y. (2016). In-field automatic observation of wheat heading stage using computer vision. *Biosyst. Eng.* 143, 28–41. doi: 10.1016/j.biosystemseng.2015.12.015

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: 5 July 2022
doi: 10.3389/fpls.2022.910878

Effects of Image Dataset Configuration on the Accuracy of Rice Disease Recognition Based on Convolution Neural Network

*Huiru Zhou, Jie Deng, Dingzhou Cai, Xuan Lv and Bo Ming Wu**

College of Plant Protection, China Agricultural University, Beijing, China

OPEN ACCESS

Edited by:
Wanneng Yang,
Huazhong Agricultural University,
China

Reviewed by:
Yang Lu,
Heilongjiang Bayi Agricultural University, China
Gongfa Li,
Wuhan University of Science and Technology, China
Liang Wanjie,
Jiangsu Academy of Agricultural Sciences (JAAS), China

****Correspondence:***
Bo Ming Wu
bmwu@cau.edu.cn

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *01 April 2022*
Accepted: *10 May 2022*
Published: *5 July 2022*

Citation:
Zhou H, Deng J, Cai D, Lv X and Wu BM (2022) Effects of Image Dataset Configuration on the Accuracy of Rice Disease Recognition Based on Convolution Neural Network. Front. Plant Sci. 13:910878. doi: 10.3389/fpls.2022.910878

In recent years, the convolution neural network has been the most widely used deep learning algorithm in the field of plant disease diagnosis and has performed well in classification. However, in practice, there are still some specific issues that have not been paid adequate attention to. For instance, the same pathogen may cause similar or different symptoms when infecting plant leaves, while the same pathogen may cause similar or disparate symptoms on different parts of the plant. Therefore, questions come up naturally: should the images showing different symptoms of the same disease be in one class or two separate classes in the image database? Also, how will the different classification methods affect the results of image recognition? In this study, taking rice leaf blast and neck blast caused by *Magnaporthe oryzae*, and rice sheath blight caused by *Rhizoctonia solani* as examples, three experiments were designed to explore how database configuration affects recognition accuracy in recognizing different symptoms of the same disease on the same plant part, similar symptoms of the same disease on different parts, and different symptoms on different parts. The results suggested that when the symptoms of the same disease were the same or similar, no matter whether they were on the same plant part or not, training combined classes of these images can get better performance than training them separately. When the difference between symptoms was obvious, the classification was relatively easy, and both separate training and combined training could achieve relatively high recognition accuracy. The results also, to a certain extent, indicated that the greater the number of images in the training data set, the higher the average classification accuracy.

Keywords: deep learning, convolutional neural network, rice diseases, image recognition, crop disease dataset, model fitting

INTRODUCTION

Rice production is facing many threats, especially many diseases caused by fungi, bacteria, and environmental factors (Zhang et al., 2018). Timely and accurate diagnosis of rice diseases is critical to the management of these diseases. Traditionally, disease diagnosis was mainly done by experienced personnel based on visible symptoms and laboratory identification (Sethy et al., 2020). However, experienced personnel are in short supply at grass-roots plant protection stations in China and many other developing countries. Besides, the identification of crop diseases using laboratory technology is often laborious and time-consuming (Feng et al., 2020). Therefore, efforts have been made to develop alternative techniques, including image recognition based on machine learning for its timely feedback and low cost (Coulibaly et al., 2019; Abade et al., 2021; Bari et al., 2021).

Early automatic diagnoses of crop diseases were mainly done *via* image recognition based on traditional machine learning (Li et al., 2020). Many traditional machine learning algorithms, including self-organizing maps (Phadikar and Sil, 2008), back propagation neural network (Xiao et al., 2018), Naive Bayes (Islam et al., 2018), K-means clustering (Ghyar and Birajdar, 2017), and support vector machine (Yao et al., 2009), have been applied to the recognition of rice disease images. These algorithms achieved classification accuracy ranging from 92 to 97.2% in these studies, but the small training dataset and the huge feature extraction engineering have been two huge obstacles to the practical application of traditional machine learning algorithms in the field of rice diseases recognition (DeChant et al., 2017; Lu J. et al., 2017).

Deep learning, with the advantages of automatic feature extraction and efficient processing of big data, triggered a boom of research on image recognition these years (Min et al., 2017). Among many deep learning algorithms, the convolutional neural network (CNN) is most widely used in the field of computer vision (Voulodimos et al., 2018). The CNN automatically learns the features of the image through convolution and pooling operations, mimicking the processes of image recognition by the cerebral perception cortex (Yamins and DiCarlo, 2016), which suggested that CNN could perform like the human visual nerves in some way (Cadieu et al., 2014).

Recently, many researchers all over the world have also paid attention to apply deep learning, especially CNN, in the diagnosis of rice diseases. Some researchers trained existing CNN models with rice disease images (Ghosal and Sarkar, 2020; Deng et al., 2021; Krishnamoorthy et al., 2021), some built their own CNN models (Lu Y. et al., 2017), and some modified the classical CNN models such as DenseNet by adding inception module (Chen et al., 2020). Lightweight models, such as simple CNN in which model parameters were greatly reduced without precision loss, have also been developed for application with mobile devices (Rahman et al., 2020). As CNN is excellent in extracting features, Liang et al. (2019) also used a traditional SVM classifier for subsequent image classification based on image features extracted by CNN from images of rice leaf blast and achieved a significantly better classification accuracy by combining SVM with CNN than by combining SVM with two traditional feature extraction methods, namely, LBPH and Haar-WT.

The existing research results suggested that deep learning-based image recognition has become more and more mature and achieved high performance in the recognition of rice diseases, both in accuracy and efficiency. Therefore, instead of building new models or improving algorithms, more attention has been paid to solve specific and practical issues in training existing models by some researchers recently. For example, Mohanty et al. (2016) found that the image type used in model training and the image allocation ratio between the training set and test set would have effects on the diagnosis accuracy of the resulted model. Picon et al. (2019) proved that training a model for multi-crops performed slightly better than developing specific models for individual crops. Lee et al. (2020) proved that if a model was trained with datasets containing plant diseases that were not associated with a specific crop, the model would be more suitable for a wider range of uses, especially for images obtained in different fields and images from unseen crops.

Similarly, automatic diagnosis of rice diseases has encountered some practical problems because of the high complexity of rice disease symptoms under field conditions. For example, similar or different symptoms can develop at different stages, under different weather conditions, or on different plant parts. Previous studies on the diagnosis of rice diseases concentrated on the recognition of typical symptoms of different rice diseases, but rarely addressed how the images of different symptoms caused by the same disease should be tagged in the construction of the training dataset. Should they be divided into different classes or combined into a single class? How will the different data configurations affect the accuracy of models? This has become an urgent problem to be solved before the automatic disease diagnosis can really be applied to field conditions.

Therefore, taking rice blast and rice sheath blight as examples in this study, experiments were conducted to explore how the split or merged disease classes in the configuration of training databases affect the recognition accuracy of the model. The specific objectives of this study were as follows:

(1) To select an appropriate model from 5 common CNN models for the subsequent investigation;
(2) To evaluate the effects of three training data configuration methods on the performance of CNN models during the training and test processes;
(3) To identify where the misclassifications lie *via* constructing a normalized confusion matrix for each method; and
(4) To explore the possible causes for misclassification by visualizing the recognition process.

MATERIALS AND METHODS

Choosing Crop Diseases and Construction of Datasets

Collection of Disease Images

Images of healthy rice leaves (HRL), and rice leaves or sheaths with symptoms of the three common diseases, rice blast (RB), rice brown spots (RBS), and rice sheath blight (RSB) were collected mainly from experimental rice fields in Panjin and Dandong cities of Liaoning Province, China. In addition, images were also collected from the greenhouse on the campus of China Agricultural University (CAU), from CAU experimental fields in Haidian District, Beijing, and from commercial fields in Wuyuan County of Jiangxi Province and in Lu'an City of Anhui Province. These images were photographed using smartphones or cameras following three rules: (1) avoid overexposure caused by direct sunlight; (2) ensure that the targeted lesion was in the center of the picture; and (3) avoid different disease symptoms in a single picture.

The rice leaves, necks, heads, and whole plants with no visible symptoms were photographed and regarded as healthy rice plants. For rice leaf blast, images of chronic (RLBC), and acute (RLBA) leaf lesions were collected in this study at the early growth stage of rice because of their importance and prevalence under field conditions, while the other two less common symptom types, namely, white spot and brown spot, were not included in this study. According to Kato (2001), the chronic type leaf symptoms were defined as spindle-shaped leaf lesions with a yellow outside halo, a brown inner ring, and a gray white center (**Figure 1A**), while acute type symptoms were defined as the leaf lesions that are nearly round or oval in shape, which often become irregular, and look like water stains with a layer of dark green mold on the surface (**Figure 1B**). Besides, images of rice neck blast (RNB), the most economically important symptom of rice blast, were also collected at the late growth stages of rice in this study. According to Kumar et al. (1992), neck blast was defined as the symptoms that appeared around the neck of rice panicles as light brown spots at the initial stage and then gradually expand up and down, leading to a white gray color of the whole rice ear, and sometimes the death of whole ear (**Figure 1C**).

Since rice brown spot caused by *Bipolalaris oryzae* has a similar shape and yellow halo to those of rice blast leaf lesions, images of rice leaves with brown spots were collected and used to test the recognition accuracy of the outcome models. According to Quintana et al. (2017), the infected leaves with sesame-like oval dark brown spots surrounded by yellow halos were considered as typical symptoms of rice brown spot (**Figure 1D**).

Another important disease, rice sheath blight caused by *Rhizoctonia solani*, which can cause similar symptoms on leaves (RSBL) and sheaths (RSBS), was also included in this study to illustrate how the classification of similar symptoms on different plant parts caused by the same pathogen would affect the accuracy of recognition. According to Lee and Rush (1983), the typical symptoms of this disease are cloud-shaped lesions on the leaf sheaths and leaves, with brown to dark brown edges and grayish green to grayish white middle parts (**Figures 1E,F**).

Preprocessing of Images

As CNN requires squared input images, in order to avoid image deformation caused by the forced compression of non-squared images during input, the automatic clipping method was used to cut each image into a square, with the side length equal to the length of the short side of the original image and using the original image center as the clipping center. The clipped images were then compressed to 500 × 500 pixels. Subsequently, normalization was applied on each image by dividing all pixel values with 255 to accelerate the convergence of models during the subsequent training procedure.

As the number of acquired images in some classes was inadequate for model training and validation, more images in these classes were generated to meet the requirement by image augmentation (**Table 1**). The methods used in augmentation included flip, translocation, rotation, and zoom (Francois, 2018).

Experimental Scheme

Three experiments were designed to investigate the effects of dataset configuration on rice disease images recognition. In each experiment, two symptoms of one disease were selected for training and testing together with the other three diseases. In experiment 1, training datasets with separate and combined classes of RLBC and RLBA were compared. In experiment 2, training datasets with separate and combined classes of RLBC and RNB were compared. In experiment 3, training datasets with separate and combined classes of RSBL and RSBS were compared.

In each experiment, a method using two separated classes and two methods with one combined classes were compared. Considering that the imbalance of data may affect the training results, two methods were used in the construction of the combined class, directly combining all the images of two classes into one class, and randomly selecting half images from each class and combining them into one class.

Construction of Datasets

Images of each class were randomly numbered after preprocessing, with a unique ID for each image. For example, the first image of RSB was named “RSB (0).” For each class, the first 500 or 1,000 images were used in training and validation datasets as required, and the images 1,001–1,099 were used to build test sets.

There were three independent datasets for each experiment. In experiment 1, 1,000 images of each class were divided into training set and validation set according to a ratio of 8:2 for method A. In method B, 1,000 images of RLBC and RLBA were directly merged into one class, with twice as many images as the other classes. In method C, 500 images were randomly taken from RLBC and RLBA, respectively, to form a combined class. The same ratio of 8:2 was used dividing image data into constructing training and validation sets in both methods B and C. In addition to the 1,000 images, other 100 images of each class were randomly selected to form a 500-image test set. These 500 images were used to test all three methods A, B, and C, but classes of RLBC and RLBA would be merged into one class for testing methods B and C. In the same way, training, validation, and test datasets were constructed in experiment 2 and experiment 3 (**Table 2**).

Hardware and Software

Keras/Tensorflow backend framework based on Anaconda3 platform was used in this study (version: keras 2.2.4, tensorflow 1.15.0), and the training and validation processes were coded using Python 3.7 programming language. The computer was equipped with 32 g memory module and GTX 1080Ti graphics card. The computer operation system was the 64-bit Windows 10 professional edition. The programs were all run on a single graphic processing unit (GPU) because the training speed on GPU is much faster than that on the central processing unit (CPU).

Training Parameter Setting

Instead of starting from scratch, transfer learning was applied in all model training experiments to saving

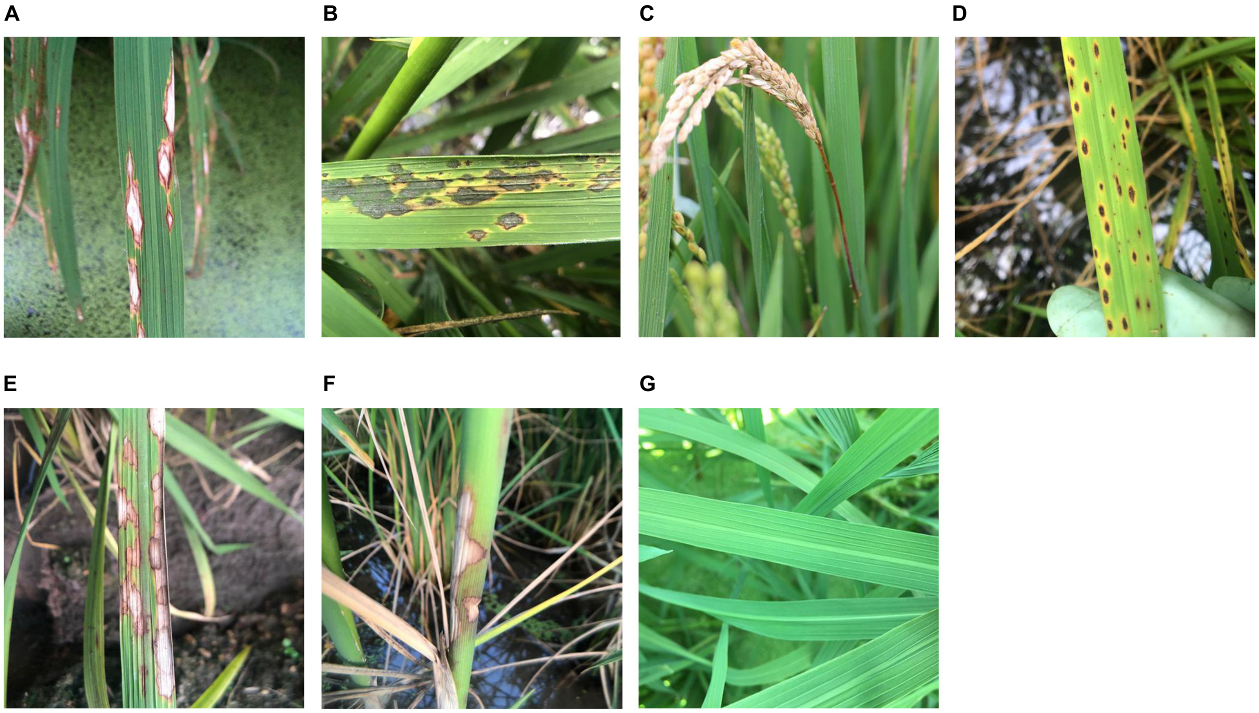


FIGURE 1 | Sample images of healthy rice leaves and rice diseases. **(A)** Chronic lesions of rice leaf blast, **(B)** acute lesions of rice leaf blast, **(C)** rice neck blast, **(D)** rice brown spots, **(E)** rice sheath blight on leaves, **(F)** rice sheath blight on sheaths, and **(G)** healthy rice leaves.

TABLE 1 | The number of images within each disease class obtained in this study.

	Rice blast			Rice sheath blight				
	Leaf		Neck	Leaf	Sheath			
	Acute (RLBA)	Chronic (RLBC)	(RNB)	(RSBL)	(RSBS)	Rice brown spots (RBS)	Healthy rice leaves (HRL)	Total
Initial number	1,146	1,186	599	1,193	598	1,143	1,146	7,011
Number after augmentation	1,146	1,186	1,198	1,193	1,196	1,143	1,146	8,208

TABLE 2 | The number of images in each disease class in training experiments using different methods.

	Experiment 1			Experiment 2			Experiment 3		
Disease classes	Method A	Method B[b]	Method C	Method J	Method K	Method L	Method X	Method Y	Method Z
HRL[a]	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
RBS	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
RLBA	1,000	$\begin{bmatrix}1,000\\1,000\end{bmatrix}$	$\begin{bmatrix}500\\500\end{bmatrix}$	/	/	/	/	/	/
RLBC	1,000			1,000	$\begin{bmatrix}1,000\\1,000\end{bmatrix}$	$\begin{bmatrix}500\\500\end{bmatrix}$	1,000	1,000	1,000
RNB	/	/	/	1,000			/	/	/
RSBL	1,000	1,000	1,000	1,000	1,000	1,000	1,000	$\begin{bmatrix}1,000\\1,000\end{bmatrix}$	$\begin{bmatrix}500\\500\end{bmatrix}$
RSBS	/	/	/	/	/	/	1,000		

[a] *HRL, healthy rice leaves; RBS, rice brown spot; RLBA, rice leaf blast-acute lesions; RLBC, rice leaf blast-chronic lesions; RNB, rice neck blast; RSBL, rice sheath blight on leaves; RSBS, rice sheath blight on sheaths.*
[b] *The images from the two classes within the braces were combined into one single class for training.*

training time by carrying the weights from the training on ImageNet dataset (Russakovsky et al., 2015). The learning rate was set as 0.001, and the training was run for 50 epochs with a momentum of 0.9, an optimization function of stochastic gradient descent (SGD), and a mini-batch size of 32.

Model Selection

Different algorithms have their own purposes or specific application scenarios when designing or modifying. For example, a multi-stream residual network (MResLSTM) was designed for dynamic hand movement recognition (Yang et al., 2021), and a modified YOLO v3 algorithm was applied to detect helmet wearing by construction personnel (Huang et al., 2021). At present, however, there is no widely used model for the diagnosis of rice diseases, so we conducted a preliminary experiment to select from five representative CNN models for subsequent experiments on the construction of datasets.

The VGG series (Simonyan and Zisserman, 2014), first developed by the VGG group of Oxford University, were CNN models with stacked 3 × 3 convolution kernels for extracting complex features with a manageable number of parameters. Considering the moderate size of our disease data, VGG16 (16 layers) was selected as the representative of this model series. Compared with the VGG series, some CNN models used more network layers to extract higher dimension features and took different approaches to handle the gradient dispersion problem associated with deeper networks (Gao et al., 2019). Inception v3 was chosen as a candidate model in this study for its deep depths and its inception module, which uses convolution kernels of different sizes in the same layer to realize feature fusion of different scales and batch normalization to speed up the learning rate (Szegedy et al., 2015). ResNet50 (50 layers) was included as a representative of ResNet series, in which a residual module was introduced for a shortcut connection in the network allowing the original input information to be directly transmitted to the later layer (He et al., 2015). In addition, MobileNet v2 (Howard et al., 2017) and NASNetMobile (Zoph et al., 2018), two representatives of the current lightweight models in the application scenarios of mobile terminals or embedded devices, were also selected for their relatively excellent performance and small number of parameters (Wang et al., 2020).

A pre-experiment was conducted to compare the performances of the five CNN models in recognition of the three rice leaf diseases and healthy rice leaves (**Figure 1G**). The 1,000 training images from each of the five classes, namely, RLBC, RLBA, RBS, RSBL, and HRL, were divided into a training dataset and a validation dataset according to the ratio of 8:2. The models were trained for 50 epochs using the transfer learning method, and the initial weights of five models were all set as the shared weights from training on ImageNet as described in the "Training parameter setting" section. The size of models, speed of training (in seconds per epoch), the highest validation accuracy, the final validation accuracy, the average validation accuracy, and standard deviation of validation accuracy were used to evaluate the models.

Experiments and Statistical Analysis

Subsequently, 3 experiments were done using the best model selected from the pre-experiment. Due to the random input order of mini-batches, the results of training could vary at each run. To estimate this variation and assess the reliability of the results, each of the three experiments was repeated three times. The final validation accuracy, final validation loss, test accuracy, and test loss were analyzed using the GLM procedure in SAS (version 9. 4, SAS Institute Inc., Cary, NC, United States) to determine whether the effects of the training dataset configuration were statistically significant.

Over the 50 epochs of the training processes, the average validation accuracy and average validation loss of three repeated experiments were calculated every four epochs. As the performance of each method fluctuated over epochs in the training process, to better express the whole trend during the process, regression was performed to fit a negative exponential decay model to the average validation accuracy and an exponential decay model to average validation loss over the training processes for each method using the non-linear regression procedure in SAS (Version 9.4, SAS Institute Inc., Cary, NC, United States).

For validation accuracy, the following model was used:

$$A = A_{max} - (A_{max} - A_0)\, e^{(-r_a \cdot x)}$$

where A was the validation accuracy and x was the epoch number in training, while A_{max}, A_0, and r_a were parameters to be estimated in model fitting. A_{max} reflects the highest validation accuracy that the method can reach, A_0 reflects the initial validation accuracy, and r_a can reflect the increase rate of A or improvement rate of validation accuracy over epochs.

For validation loss, the following model was used in regression:

$$L = L_{min} + e^{(-r_l \cdot x + b)}$$

where L was the validation loss and x was the epoch number in training, while L_{min}, r_l, and b were parameters to be estimated during model fitting. L_{min} represents the lowest validation loss rate obtained by this method after unlimited epochs, $L_{min} + e^{(-r_l + b)}$ reflects the initial validation loss at epoch #1, and r_l is related to the decline rate of validation loss.

After model fitting, the parameters were compared between different methods using Student's t-test (Steel and Torrie, 1980) to characterize the disparity of the three methods in the training process.

Normalized Confusion Matrix

Confusion matrix, which was widely used in the evaluation of classification accuracy in many areas, was constructed for comparison of different training dataset configuration methods based on test results. As the image numbers of the classes to be tested in this study varied among different training dataset configuration methods, to better reflect their difference in classification accuracy, the normalized confusion matrix was used. For any classification with c classes, the confusion matrix consisted of c rows × c columns, and the element in the i-th row and j-th column was calculated by dividing the number of images that belonged to the i class and were classified into the j class with the total number of images in the row.

Heatmap

To understand which parts of the input image, such as the lesion edge, the center, or other areas, had contributed more to the automatic classification by the models, for each representative image with a high frequency of misclassification in recognition, a heatmap of class activation was generated using the GRAD-CAM algorithm (Selvaraju et al., 2020), in which the pixels that contributed heavily to the final classification will be presented as yellow to red colors and those that contributed less will be presented in green to purple colors. The heatmap generated this way serves as a tool for visualization of the feature extraction process of deep neural network.

RESULTS

Performance of the Five Models in Pre-experiments

The results from the pre-experiment demonstrated that the five CNN models performed differently in the classification of these images (**Table 3**). VGG16 and Inception v3 all achieved a validation accuracy higher than 99%, but ResNet50 had the highest average validation accuracy and a smaller standard deviation among these models, suggesting that its convergence speed was the fastest and its performance was the most stable. Considering ResNet50's excellent performance in training, including good speed (38 s/epoch), the highest final validation accuracy, the highest average validation accuracy, and the smallest standard deviation, it was selected for the subsequent training experiments.

Experiment 1: Different Symptoms of the Same Disease on the Same Part

The results from experiment 1 revealed that the training curves of validation accuracy and validation loss using method A differed from those using methods B and C (**Figures 2A,B**). Method A consistently had lower validation accuracy and higher validation loss than methods B and C did during the whole training process over 50 epochs (**Figures 2A,B**). Differences also existed between method B and method C in the early epochs of the training process, but the difference gradually decreased to an ignorable level with the increase of training epochs. Regardless of methods used in training dataset configuration, the trends of validation accuracy over training epochs could be fitted well to the negative exponential decay model (**Table 4A**) and those of validation loss fitted well to the exponential decay model (**Table 4B**). The t-test indicated that the highest accuracy (A_{max}) obtained using method A was lower than those using the other two methods, while the lowest validation loss (L_{min}) using method A was significantly greater than those using methods B and C (**Tables 4A,B**). The growth rate r_a of validation accuracy and the decline rate r_l of validation loss were significantly faster for method B than for the other two methods.

The ANOVA and multiple mean comparison revealed that on both validation and test datasets, the validation accuracy and test accuracy obtained using method A were significantly lower than those obtained using methods B and C, and the validation loss and test loss obtained using method A were significantly greater than those obtained using methods B and C (**Table 5**).

The confusion matrix of test results using method A revealed that the class with the lowest accuracy was RLBC, and the main classification errors came from the misclassification of RLBC images into RLBA by the model (**Figure 3A**). When combining the two classes into one for training, the test accuracy ranged from 96 to 99% in every class with little variation among classes (**Figures 3B,C**). To understand why the misclassifications occurred, the original images of these misclassified RLBC images were visually examined again. It was found that although the leaf lesions in these images were nearly spindle shaped, the edges and corners were not obvious enough. When there were many lesions on leaves, they connected into pieces that were more like water stains, and the surfaces of some lesions were even gray green, which were typical symptoms of RLBA at the early stage of developing into RLBC (Kumar et al., 1992).

Experiment 2: Different Symptoms of the Same Disease at Different Parts

The validation accuracy obtained using method K was highest among the three methods at the beginning of the training processes, and the lowest accuracy was gained using method L, but the accuracy increase rates r_a were higher for methods L (0.6564) and J (0.5377) than for method K (0.2800), and as a result, the three methods differed very less in accuracy after 20 training epochs (**Figure 2C** and **Table 4A**), and the maximum accuracy gained after 50 epochs varied from 0.9935 to 0.9940,

TABLE 3 | Performance of five CNN models in the classification of rice disease images.

CNN models	Seconds/epoch	Highest validation accuracy[a]	Final validation accuracy[b]	Average validation accuracy[c]	Standard deviation of validation accuracy[d]
VGG16	36	0.9900	0.9890	0.8758	0.2473
Inception v3	60	0.9980	0.9920	0.9803	0.4020
ResNet50	**38**	**0.9940**	**0.9920**	**0.9851**	**0.0107**
MobileNet v2	37	0.9830	0.9520	0.7138	0.2199
NASNetMobile	57	0.9870	0.9870	0.9693	0.0352

[a] *The highest validation accuracy achieved within the first 50 epochs.*
[b] *The final validation accuracy after 50 epochs.*
[c] *The average validation accuracy over the first 50 epochs.*
[d] *The standard deviation of validation accuracy over the first 50 epochs, which reflects the convergence rate of five models.*
The training results of the selected model ResNet50 were shown in bold.

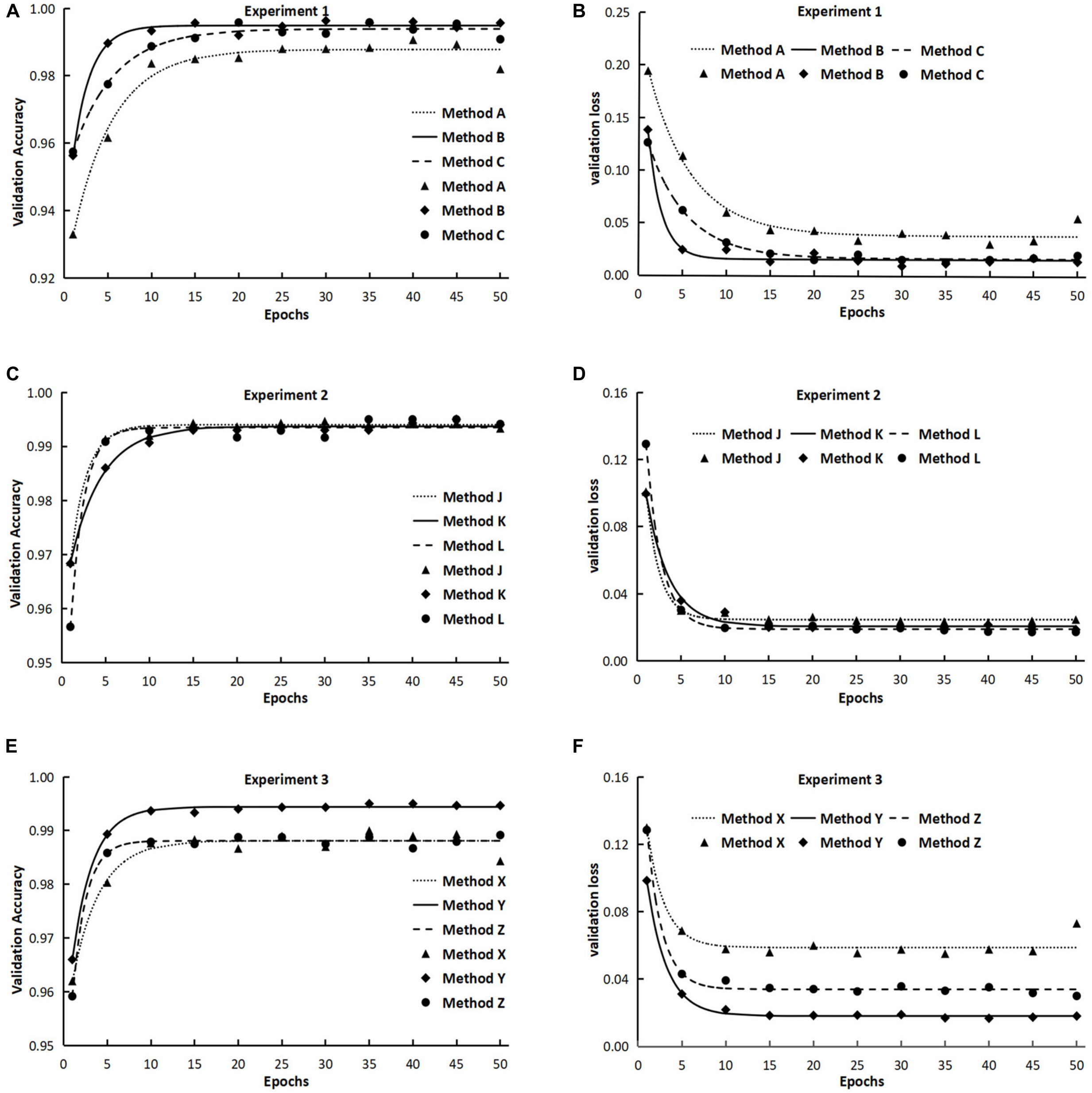


FIGURE 2 | The validation accuracy and validation loss during the training processes in experiments 1, 2, and 3. (The points in the figures were means from three repeated runs, and the lines represented the fitted models of validation accuracy and validation loss.) Method A: Training with two separate classes, namely, acute type of rice leaf blast (RLBA) and chronic type of rice leaf blast (RLBC). Method B: Combining RLBA and RLBC as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method C: Combining RLBA and RLBC as one class for training and the total number of images in the combined class was equal to those in the other three classes. Method J: Training with two separate classes of RLBC and rice neck blast (RNB). Method K: Combining RLBC and RNB as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method L: Combining RLBC and RNB as one class for training and the total number of images in the combined class was equal to those in the other three classes. Method X: Training with two separate classes of rice sheath blight on leaves (RSBL) and rice sheath blight on sheath (RSBS). Method Y: Combining RSBL and RSBS as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method Z: Combining RSBL and RSBS as one class for training and the total number of images in the combined class was equal to those in the other three classes.

showing no significant difference among the three methods (**Table 4A**). On the contrary, the validation loss using method L was the highest among the three methods early in the training, but it declined quickly as the training progressed and ended the training with a loss value that was very close to the other two methods (**Figure 2D** and **Table 5**).

Interestingly, although the accuracy using methods J, K, and L differed slightly (insignificantly) on validation data, the test

TABLE 4A | Parameters and determinant coefficients of models [$A = A_{max} - (A_{max} - A_0)\, e^{-r_a \cdot x}$] fitted to the validation accuracy over training epochs in 3 experiments using different methods.

Experiment-method	A_0	A_{max}	r_a	R^2
1-method A	0.9191 ± 0.0047^{b}	0.9878 ± 0.0011^{b}	0.2154 ± 0.0289^{b}	0.977
1-method B	0.9319 ± 0.0050^{b}	0.9949 ± 0.0005^{a}	0.4896 ± 0.0667^{a}	0.985
1-method C	0.9489 ± 0.0028^{a}	0.9939 ± 0.0007^{a}	0.2077 ± 0.0254^{b}	0.985
2-method J	0.9507 ± 0.0035^{b}	0.9940 ± 0.0003^{a}	0.5377 ± 0.0718^{a}	0.991
2-method K	0.9603 ± 0.0014^{a}	0.9937 ± 0.0003^{a}	0.2800 ± 0.0230^{b}	0.992
2-method L	0.9223 ± 0.0099^{c}	0.9935 ± 0.0005^{a}	0.6564 ± 0.1332^{a}	0.992
3-method X	0.9520 ± 0.0037^{a}	0.9881 ± 0.0006^{b}	0.3206 ± 0.0629^{b}	0.960
3-method Y	0.9510 ± 0.0015^{a}	0.9944 ± 0.0002^{a}	0.4244 ± 0.0260^{b}	0.997
3-method Z	0.9340 ± 0.0053^{b}	0.9881 ± 0.0003^{b}	0.6274 ± 0.0921^{a}	0.992

R^2: Degree of coincidence between test data and fitting function. The closer the value of R^2 is to 1, the higher the degree of coincidence is.
Within each experiment, the fitted value followed by different letters in the same column differed significantly at confidence level p = 0.05.

TABLE 4B | Parameters and determinant coefficients of models [$L = L_{min} + e^{(-r_l \cdot x + b)}$] fitted to the validation loss over training epochs in 3 experiments using different methods.

Experiment-method	L_{min}	r_l	b	R^2
1-method A	0.0382 ± 0.0030^{a}	–0.2020 ± 0.0242^{b}	−1.6449 ± 0.0591^{a}	0.980
1-method B	0.0156 ± 0.0016^{b}	–0.6331 ± 0.1274^{a}	−1.4649 ± 0.1325^{a}	0.985
1-method C	0.0165 ± 0.0009^{b}	–0.2226 ± 0.0126^{b}	−1.9856 ± 0.0281^{b}	0.991
2-method J	0.0246 ± 0.0006^{a}	–0.6433 ± 0.0788^{a}	−1.9319 ± 0.0816^{b}	0.996
2-method K	0.0206 ± 0.0005^{b}	–0.3797 ± 0.0360^{b}	–2.1610 ± 0.0499^{c}	0.991
2-method L	0.0189 ± 0.0005^{b}	–0.5632 ± 0.0353^{a}	−1.6410 ± 0.0382^{a}	0.991
3-method X	0.0587 ± 0.0019^{a}	–0.5010 ± 0.1536^{a}	–2.1412 ± 0.1754^{ab}	0.948
3-method Y	0.0181 ± 0.0004^{c}	–0.4453 ± 0.0216^{a}	–2.0777 ± 0.0266^{b}	0.998
3-method Z	0.0339 ± 0.0009^{b}	–0.5686 ± 0.0682^{a}	−1.7911 ± 0.0734^{a}	0.994

Within each experiment, the fitted value followed by different letters in the same column differed significantly at confidence level p = 0.05.

accuracy obtained using method L was significantly lower, and the test loss was significantly greater than those obtained using method J and method K (**Table 5**).

The confusion matrix for method J in experiment 2 revealed that the model can distinguish RNB from other classes well, and the accuracy of RLBC was the lowest among the five classes, with majority of misclassification errors between RLBC and RSBL, but its recognition accuracy of RNB was relatively high (**Figure 3D**). When a combined class of RNB with RLBC was used in methods K and L, the accuracy of the combined RNB/RLBC class was between those of the two separate classes (**Figures 3E,F**). It was also noted that considerable errors existed in misclassifying RSBL into RLBC or combined class of RLBC with RNB regardless of the methods used (**Figures 3D–F**). This revealed that the identification of different plant parts is an indispensable part of classification by CNN models, and this identification could help to distinguish diseases on different plant parts, but similar symptoms on the same plant parts could not use this information and therefore become a more difficult task. It was also very interesting to note that method L had lower accuracy on combined RLBC/RNB class than method K. This might have been because method K had been trained with more images of the combined class than method L.

TABLE 5 | The accuracy and loss obtained with validation and test datasets in three experiments using different methods.

Experiment-method	Val_acc	Val_loss	Test_acc	Test_loss
1-Method A	0.9831^{b}	0.0530^{a}	0.9573^{b}	0.1680^{a}
1-Method B	0.9928^{a}	0.0211^{b}	0.9767^{a}	0.0685^{b}
1-Method C	0.9912^{a}	0.0254^{b}	0.9727^{a}	0.0957^{b}
2-Method J	0.9928^{a}	0.0282^{a}	0.9593^{a}	0.1223^{c}
2-Method K	0.9918^{a}	0.0251^{a}	0.9547^{a}	0.1586^{b}
2-Method L	0.9919^{a}	0.0236^{a}	0.9420^{b}	0.1884^{a}
3-Method X	0.9867^{b}	0.0610^{a}	0.9400^{a}	0.2174^{a}
3-Method Y	0.9927^{a}	0.0225^{c}	0.9593^{a}	0.1060^{a}
3-Method Z	0.9871^{b}	0.0376^{b}	0.9527^{a}	0.1263^{a}

Each of the values in the table was the mean from 3 repeated runs. Within each experiment, the means followed by different letters in the same column differed significantly at confidence level p = 0.05.

Experiment 3: Similar Symptoms of the Same Disease at Different Parts

The initial validation accuracy of method Y was the highest among the three methods, and with the increase in training epochs, its validation accuracy remained highest all way to the end (**Figure 2E**). The results from *t*-test on model parameters A_{max} and r_a revealed that the highest validation accuracy A_{max} from method Y was significantly higher than those from the other two methods, but no significant difference in r_a was detected

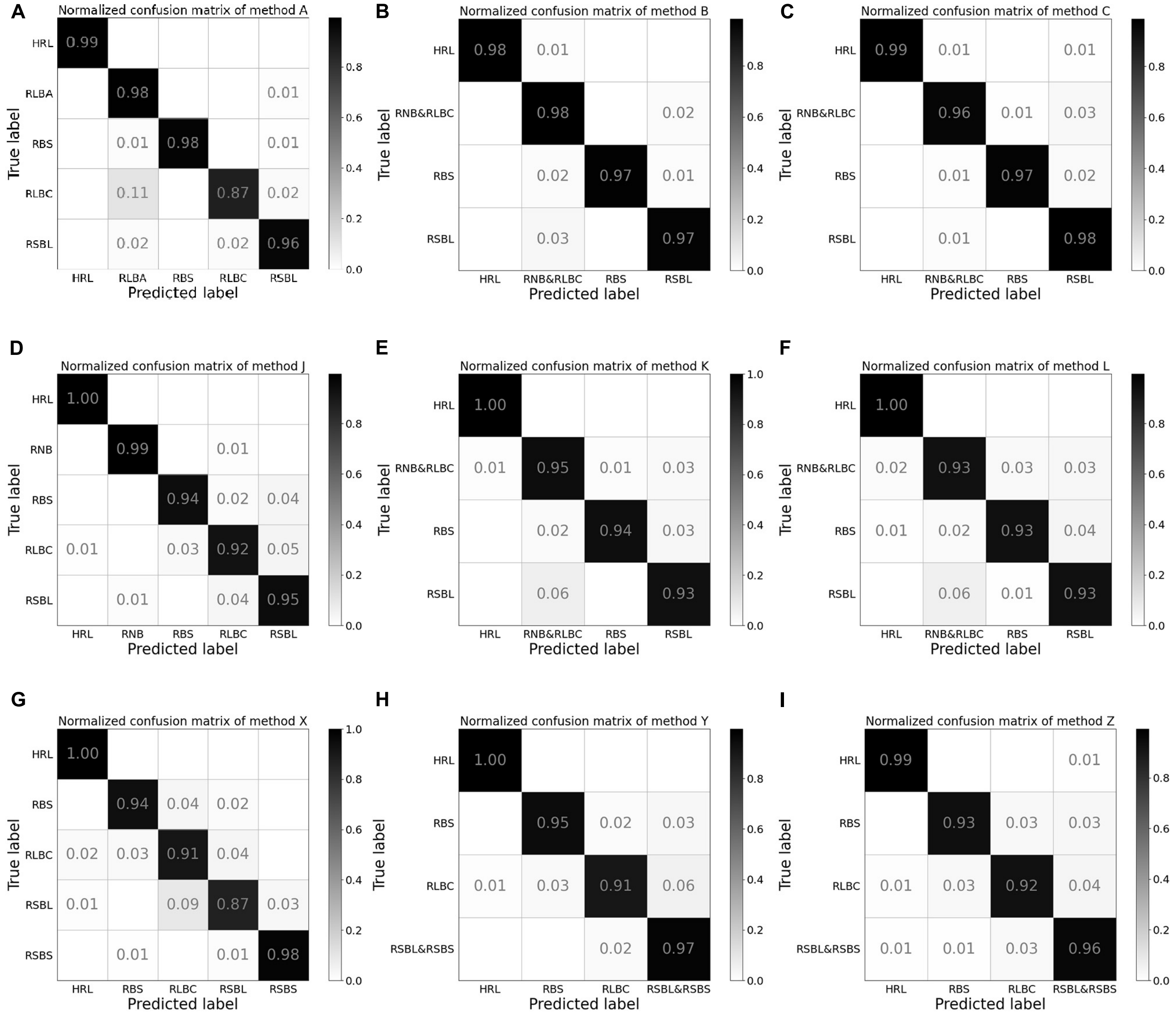


FIGURE 3 | The normalized confusion matrix for the test results from experiments 1, 2, and 3. Method A **(A)**: Training with two separate classes, namely, acute type of rice leaf blast (RLBA) and chronic type of rice leaf blast (RLBC). Method B **(B)**: Combining RLBA and RLBC as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method C **(C)**: Combining RLBA and RLBC as one class for training and the total number of images in the combined class was equal to those in the other three classes. Method J **(D)**: Training with two separate classes of RLBC and rice neck blast (RNB). Method K **(E)**: Combining RLBC and RNB as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method L **(F)**: Combining RLBC and RNB as one class for training and the total number of images in the combined class was equal to those in the other three classes. Method X **(G)**: Training with two separate classes of rice sheath blight on leaves (RSBL) and rice sheath blight on sheath (RSBS). Method Y **(H)**: Combining RSBL and RSBS as one class for training and the total number of images in the combined class was two times as those in the other three classes. Method Z **(I)**: Combining RSBL and RSBS as one class for training and the total number of images in the combined class was equal to those in the other three classes.

between method Y and method Z (**Table 4A**). The validation loss curves obtained with three methods displayed trends reverse to validation accuracy, in that no significant difference in the decline rate r_l of validation loss was detected among three methods (**Figure 2F**), but method Y had the lowest validation loss among the three methods, and method X had the highest validation loss.

The test results showed that the average test accuracy of method Y was much higher than that of method X and slightly higher than that of method Z (**Table 5**), although the ANOVA detected no significant difference between the three methods (see **Supplementary Material**). The confusion matrix of the test results illustrated that the model trained with method X misclassified 4% RLBC images as RSBL and 9% RSBL images as RLBC (**Figure 3G**). When the model was trained using method Y, with a combined class of RSBL&RSBS, its accuracy was greatly improved that it misclassified 6% of RLBC images into the combined class, but only misclassified 2% RSBL&RSBS images into RLBC (**Figure 3H**). Similar results were gained with method Z (**Figure 3I**). Once again, classifying between RLBC and RSBL was a difficult task, and the accuracy for the combined class was higher for method Y than for method Z, where more combined class images were used in training for method Y than method Z.

To further explore the reasons why differentiating RLBC and RSBL was difficult and easy to be misclassified for the outcome models, heatmaps of RLBC samples correctly classified, RLBC images misclassified as RSBL, RSBL samples correctly classified, and RSBL images misclassified as RLBC were compared (**Figure 4**). For those correctly classified RLBC, the areas with hot color were concentrated around the disease lesions, suggesting an excellent feature extraction by the model (**Figures 4.1–4.4**). However, it was observed that in most of the misclassified RLBC samples, the hot loci were not well overlapped with the disease lesions, suggesting the model didn't extract important lesion features for decision-making, interfered either by other leaf damages (**Figures 4.5**,**4.6**) or field background (**Figures 4.7**,**4.8**). The existence of RLBC that directly led to the test results of the three methods of experiment 3 was not significantly different. Unlike RLBC, for all RSBL images, regardless of whether correctly classified or misclassified, the classification areas were mainly concentrated on the disease lesion area (**Figures 4.9–4.16**). It can be seen that compared with the typical symptoms of RSBL, in most of the misclassified samples, lesions were relatively small, had gray center areas, and were surrounded by brown halos, which was, to a certain degree, similar to the atypical RLBC, except for the subtle difference in lesion shapes (**Figures 4.15**,**4.16**). This may be one of the reasons why more rice sheath blight images were identified as RLBC than RBS.

DISCUSSION AND CONCLUSION

In this study, we explored some specific problems encountered in dataset configuration for automatic recognition of rice diseases. The results from this study demonstrated that whether a combined class or several separated classes should be used depend on the similarity of these classes. For example, our results from experiment 1 demonstrated that using a combined class

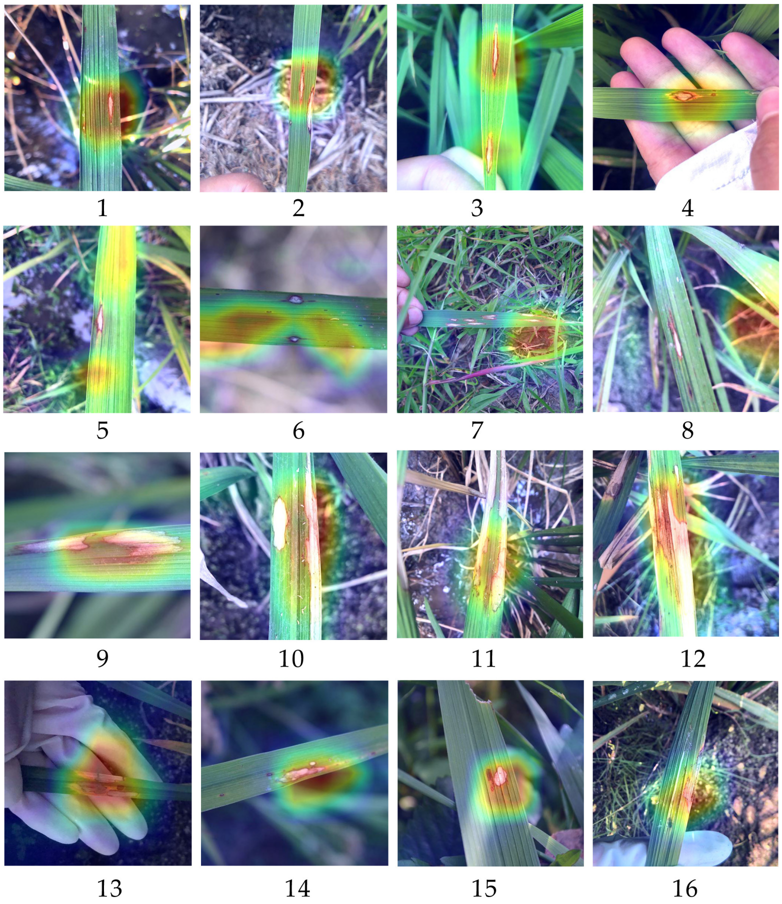


FIGURE 4 | Heatmaps generated based on the classification by models trained with methods X for some rice leaf blast samples and rice sheath blight samples. **(1–4)** The samples of chronic type of rice leaf blast (RLBC) that were correctly recognized as RLBC; **(5–8)** The samples of RLBC that were mistakenly recognized as RSB. **(9–12)** The samples of rice sheath blight (RSB) that were correctly recognized as RSB. **(13–16)** The samples of RSB that were mistakenly recognized as RLBC. The red part has the highest contribution to the final prediction results. On the contrary, the purple part has the lowest contribution to the final prediction results of an image).

for RLBA and RLBC, two very similar symptoms on rice leaves, could achieve better recognition performance (higher accuracy and lower loss) than using two separate classes. A possible explanation might be that similar lesions sometimes are difficult to differentiate even for human experts because acute lesions often gradually develop into chronic lesions in the later stage (Kumar et al., 1992). This was also supported by the high misclassification rate between these two classes by method A using separate classes, but relatively low misclassification rates between any of these two classes and other class by method A. Similarly, for RSBL and RSBS in experiment 3, using a combined class in the training dataset could achieve a better performance than using two separate classes. A possible explanation is that using two separate classes of RSBL and RSBS will require the model to differentiate the similar cloud-shaped lesions on leaves and on sheaths and therefore will increase the possibility for the model to make mistakes in recognition of the background plant parts. However, using a combined class and using separate classes for RLBC and RNB had no significant impact on the performance of resulted models in experiment 2 where two symptoms were on different plant parts. A possible explanation for this was that with information from areas surrounding lesions, it is relatively easy for CNN model to differentiate two different symptoms, and thus, using a combined class or two separated classes didn't have any significant impact on the final recognition as illustrated among methods J, K, and L in experiment 2 of this study.

The results from this study illustrated that a large number of images were required for training to achieve a high and repeatable recognition accuracy. As revealed in experiment 2, method L, in which the model was trained with half as many images of RLBC/RNB class as in methods J and K, although gained very high validation accuracy, performed significantly worse than methods J and K when tested with unseen images. So, for deep learning model, how many images are required to achieve best recognition effect? Is the more the number of images, the better the result will be? So far, few experts have explored this issue, and the number of images used in the existing literature varied from dozens to thousands. Rangarajan et al. (2018) discussed the influence of different number of images on the accuracy of the model, but the total number of images was small, and a scientific validation process has not been established yet. More in-depth studies are needed to answer this question in the future.

Through this study, we further prove the excellent ability of CNN in feature extraction. Based on the results of three experiments, it can be seen that the main features affecting the decision-making of rice disease classification models came from the disease lesion area, then from the area of plant organs, and finally from the image background. This is also consistent with the logic of human beings when classifying crop diseases. It can be seen from the heatmaps that for most samples, whether by correctly classified or by misclassified, the main feature areas that affect the model decision-making were still concentrated on the lesion area, and the areas were covered with red or yellow. The nearby areas of rice organs were also yellow, while the less important background areas were covered with blue or purple. The results of experiment 3 showed that when the disease lesions of RSB were similar, even if they existed on different organs, there would be confusion between RSBL and RSBS to some extent, indicating that the main distinguishing features still came from the lesion. At the same time, the results of experiment 2 showed that when the symptoms and organs were different, the model could extract more favorable information except the features of disease lesions, and that is why it can well distinguish the two classes when separately training RLBC and RNB. Does this mean that the image background is not important? Studies have shown that although the targets on the simple indoor background image and the complex field image were the same, the models trained by the two image sets could not be universal (Ferentinos, 2018). From the heatmaps of RLBC images misclassified as RSBL, it can be seen that although the error rate was low, the main factor causing the wrong model decision was the feature extraction of the field background. This also showed that the recognition of the background played an auxiliary role for the model. Therefore, it is very important to collect disease images under different conditions to improve the generalization ability of the models.

The results of three experiments showed that if the data configuration scheme was correct, the overall accuracy could be effectively improved. In experiment 1, combining the two similar leaf symptoms of rice and training, the validation accuracy was improved from 0.9831 to 0.9928, and the test accuracy was improved from 0.9573 to 0.9767, which was statistically significant at the confidence level of 0.05. Similarly, in experiment 3, the average accuracy of two symptoms of rice sheath blight was improved to 0.9700 from 0.9250 by using a combined class for similar symptoms on different plant parts. However, the results of experiment 2 revealed that for disparate symptoms on different plant parts, training with one combined class or two separate classes makes no difference, and the amount of data is a key factor affecting the overall accuracy. The average test accuracy of method L with a smaller data set was significantly lower (at a confidence level of 0.05) than that of the other two methods with larger datasets.

This study proposed a database configuration scheme among different symptoms of the same rice disease. Similar problems are often encountered in the diagnosis of other crop diseases (Barbedo, 2016). If our goal is to achieve a high overall classification accuracy, the findings from this study provide a reference. However, if the purpose is to differentiate multiple similar disease symptoms on different plant parts or at different stages, even if the symptoms are similar, they should be separately trained. Hopefully, the findings from this study can inspire researchers to put more efforts in automatic crop disease identification and think about the problems of disease identification from more different perspectives.

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

HZ led the whole experimental process and wrote the manuscript. JD provided suggestions about experimental design and guided the operation of the algorithms. DC participated in the construction of the dataset and collated data. XL participated in the image collection process. BW supervised the project and wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This research was funded by the National Natural Science Foundation of China (No. 31471727) and the National Key Research and Develop Program of China (#2016YFD0300702).

ACKNOWLEDGMENTS

We would like to thank Donggang Demonstration Farm, Panjin Saline Alkali Land Utilization, and Research Institute for providing experimental fields in collecting rice disease images. We would also like to thank Huayuan Yao, Kai Yuan, Meina Hu and Hui Liu for their help in the process of collecting images.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.910878/full#supplementary-material

REFERENCES

Abade, A., Ferreira, P. A., and Vidal, F. D. B. (2021). Plant diseases recognition on images using convolutional neural networks: a systematic review. *Comput. Electron. Agric.* 185:106125. doi: 10.1016/j.compag.2021.106125

Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst. Eng.* 144, 52–60. doi: 10.1016/j.biosystemseng.2016.01.017

Bari, B. S., Islam, M. N., Rashid, M., Hasan, M. J., Razman, M. A. M., Musa, R. M., et al. (2021). A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Comput. Sci.* 7:e432. doi: 10.7717/peerj-cs.432

Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., et al. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* 10:e1003963. doi: 10.1371/journal.pcbi.1003963

Chen, J., Zhang, D., Nanehkaran, Y. A., and Li, D. (2020). Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agric.* 100, 3246–3256. doi: 10.1002/jsfa.10365

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

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., et al. (2017). Automated identification of northern leaf Blight-Infected maize plants from field imagery using deep learning. *Phytopathology* 107, 1426–1432. doi: 10.1094/PHYTO-11-16-0417-R

Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., et al. (2021). Automatic diagnosis of rice diseases using deep learning. *Front. Plant Sci.* 12:701038. doi: 10.3389/fpls.2021.701038

Feng, L., Wu, B., Zhu, S., Wang, J., Su, Z., Liu, F., et al. (2020). Investigation on data fusion of multisource spectral data for rice leaf diseases identification using machine learning methods. *Front. Plant Sci.* 11:577063. doi: 10.3389/fpls.2020.577063

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

Francois, C. (2018). *Deep Learning with Python*. New Jersey: Manning Publications.

Gao, Q., Liu, J., Ju, Z., and Zhang, X. (2019). Dual-Hand detection for human–robot interaction by a parallel network based on hand detection and body pose estimation. *IEEE Trans. Ind. Electron*. 66, 9663–9672. doi: 10.1109/TIE.2019.2898624

Ghosal, S., and Sarkar, K. (2020). "Rice leaf diseases classification using CNN with transfer learning," in *Proceedings of 2020 IEEE Calcutta Conference*. (Piscataway: IEEE), 230–235.

Ghyar, B. S., and Birajdar, G. K. (2017). "Computer vision based approach to detect rice leaf diseases using texture and color descriptors," in *Proceedings of the International Conference on Inventive Computing and Informatics(ICIC)*. (Piscataway: IEEE), 1074–1078.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *arXiv* [preprint]. doi: 10.48550/arXiv.1512.03385

Howard, A. G., Menglong, Z., Chen, B., Kalenichenko, D., Weijun, W., Weyand, T., et al. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv* [preprint]. doi: 10.48550/arXiv.1704.04861

Huang, L., Fu, Q., He, M., Jiang, D., and Hao, Z. (2021). Detection algorithm of safety helmet wearing based on deep learning. *Concurr. Comput. Pract. Exp*. 33:e6234. doi: 10.1002/cpe.6234

Islam, T., Sah, M., Baral, S., and RoyChoudhury, R. (2018). "A Faster Technique on Rice Disease Detection using Image Processing of Affected Area in Agro-Field," in *Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (Coimbatore, India: IEEE), 62–66. doi: 10.1109/ICICCT.2018.8473322

Kato, H. (2001). Rice blast disease - an introduction. *Pestic. Outlook* 12, 23–25. doi: 10.1039/b100803j

Krishnamoorthy, N., Prasad, L. V. N., Kumar, C. S. P., Subedi, B., Abraha, H. B., and Sathishkumar, V. E. (2021). Rice leaf diseases prediction using deep neural networks with transfer learning. *Environ. Res.* 198:111275. doi: 10.1016/j.envres.2021.111275

Kumar, J., Chaube, H. S., Singh, U. S., and Mukhopadhyay, A. N. (1992). *Plant Diseases of International Importance*. New Jersey: Prentice Hall, Inc.

Lee, F. N., and Rush, M. C. (1983). Rice sheath blight: a major rice disease. *Plant Dis*. 67, 829–832. doi: 10.1094/PD-67-829

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

Li, Y., Nie, J., and Chao, X. (2020). Do we really need deep CNN for plant diseases identification? *Comput. Electron. Agric.* 178:105803. doi: 10.1016/j.compag.2020.105803

Liang, W. J., Zhang, H., Zhang, G. F., and Cao, H. X. (2019). Rice blast disease recognition using a deep convolutional neural network. *Sci. Rep.* 9:2869. doi: 10.1038/s41598-019-38966-0

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

Lu, Y., Yi, S. J., Zeng, N. Y., Liu, Y. R., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Brief. Bioinform.* 18, 851–869. doi: 10.1093/bib/bbw068

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

Phadikar, S., and Sil, J. (2008). "Rice Disease Identification using Pattern Recognition Techniques," in *Proceedings of the 2008 11th International*

Conference on Computer and Information Technology. (Khulna, Bangladesh: IEEE), 420–423.

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., and Echazarra, J. (2019). Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Comput. Electron. Agric.* 167:105093. doi: 10.1016/j.compag.2019.105093

Quintana, L., Gutierez, S., Arriola, M., Morinigo, K., and Ortiz, A. (2017). Rice brown spot Bipolaris oryzae (Breda de Haan) Shoemaker in Paraguay. *Trop. Plant Res.* 4, 419–420. doi: 10.22271/tpr.2017.v4.i3.055

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

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

Russakovsky, O., Deng, J., and Su, H. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* 128, 336–359. doi: 10.1007/s11263-019-01228-7

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

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

Steel, R. G. D., and Torrie, J. H. (1980). *Principles and Procedures of Statistics*. New York: McGraw-Hill Book Company.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *arXiv* [preprint]. doi: 10.48550/arXiv.1512.00567

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: a brief review. *Comput. Intel. Neurosci.* 2018:7068349. doi: 10.1155/2018/7068349

Wang, W., Hu, Y., Zou, T., Liu, H., Wang, J., and Wang, X. (2020). A new image classification approach via improved MobileNet models with local receptive field expansion in shallow layers. *Comput. Intel. Neurosci*. 2020:8817849. doi: 10.1155/2020/8817849

Xiao, M., Ma, Y., Feng, Z., Deng, Z., Hou, S., Shu, L., et al. (2018). Rice blast recognition based on principal component analysis and neural network. *Comput. Electron. Agric.* 154, 482–490. doi: 10.1016/j.compag.2018.08.028

Yamins, D. L. K., and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yang, Z., Jiang, D., Sun, Y., Tao, B., Tong, X., Jiang, G., et al. (2021). Dynamic gesture recognition using surface EMG signals based on Multi-Stream residual network. *Front. Bioeng. Biotechnol.* 9:779353. doi: 10.3389/fbioe.2021.779353

Yao, Q., Guan, Z., Zhou, Y., Tang, J., Hu, Y., and Yang, B. (2009). "Application of support vector machine for detecting rice diseases using shape and color texture features," in *Proceedings of the 2009 International Conference on Engineering Computation*. (Hong Kong, China: IEEE), 79–83. doi: 10.1109/ICEC.2009.73

Zhang, J., Yan, L., and Hou, J. (2018). "Recognition of rice leaf diseases based on salient characteristics," in *Proceedings of the 2018 13th World Congress on Intelligent Control and Automation (WCICA)*. (Changsha, China: IEEE), 801–806.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). "Learning transferable architectures for scalable image recognition," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Salt Lake City, USA: IEEE), 8697–8710. doi: 10.1109/CVPR.2018.00907

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 13 September 2022
DOI 10.3389/fpls.2022.991487

OPEN ACCESS

EDITED BY
Kioumars Ghamkhar,
AgResearch Ltd, New Zealand

REVIEWED BY
Changji Wen,
Jilin Agricultural University, China
Longsheng Fu,
Northwest A&F University, China

*CORRESPONDENCE
Chenglin Wang
wangchenglin055@163.com
Lixue Zhu
zhulixue@zhku.edu.cn

SPECIALTY SECTION
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

RECEIVED 11 July 2022
ACCEPTED 15 August 2022
PUBLISHED 13 September 2022

CITATION
Lin G, Wang C, Xu Y, Wang M, Zhang Z and Zhu L (2022) Real-time guava tree-part segmentation using fully convolutional network with channel and spatial attention. *Front. Plant Sci.* 13:991487. doi: 10.3389/fpls.2022.991487

Real-time guava tree-part segmentation using fully convolutional network with channel and spatial attention

Guichao Lin[1,2], Chenglin Wang[1,2]*, Yao Xu[1], Minglong Wang[1], Zhihao Zhang[1] and Lixue Zhu[1,2]*

[1]School of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou, China, [2]Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou, China

It is imminent to develop intelligent harvesting robots to alleviate the burden of rising costs of manual picking. A key problem in robotic harvesting is how to recognize tree parts efficiently without losing accuracy, thus helping the robots plan collision-free paths. This study introduces a real-time tree-part segmentation network by improving fully convolutional network with channel and spatial attention. A lightweight backbone is first deployed to extract low-level and high-level features. These features may contain redundant information in their channel and spatial dimensions, so a channel and spatial attention module is proposed to enhance informative channels and spatial locations. On this basis, a feature aggregation module is investigated to fuse the low-level details and high-level semantics to improve segmentation accuracy. A tree-part dataset with 891 RGB images is collected, and each image is manually annotated in a per-pixel fashion. Experiment results show that when using MobileNetV3-Large as the backbone, the proposed network obtained an intersection-over-union (IoU) value of 63.33 and 66.25% for the branches and fruits, respectively, and required only 2.36 billion floating point operations per second (FLOPs); when using MobileNetV3-Small as the backbone, the network achieved an IoU value of 60.62 and 61.05% for the branches and fruits, respectively, at a speed of 1.18 billion FLOPs. Such results demonstrate that the proposed network can segment the tree-parts efficiently without loss of accuracy, and thus can be applied to the harvesting robots to plan collision-free paths.

KEYWORDS

tree-part segmentation, MobileNetV3, attention mechanism, neural network, harvesting robot

Introduction

Fruit harvesting is time-sensitive and labor-intensive, making manual picking expensive. In order to reduce the cost burden of manual picking, it is of great significance to develop intelligent harvesting robots. In structured environments, fruit trees are often planted in a V shape (Chen et al., 2021) or plane shape (Zhang et al., 2018), and fruit detection and localization are key problems facing the robots, which have been well-addressed. However, in unstructured environments, the fruit trees have complex three-dimensional structures, and therefore a major problem facing the robots is how to recognize tree parts (including fruits, branches, and backgrounds) for the robots to plan collision-free paths (Lin et al., 2021a). Due to the complex shape and uneven thickness of the branches, the tree parts are difficult to identify (Barth et al., 2018; Lin et al., 2021b). Guava is a fruit widely grown in Guangdong Province, China. In this study, a real-time and accurate guava tree-part segmentation method is investigated to enable the guava-harvesting robots to work in unstructured environments.

Tree-part segmentation can be accomplished by traditional image analysis methods, requiring manual design of classifiers *via* feature engineering (Amatya et al., 2016; Ji et al., 2016). Such methods are usually limited to specific environments and fruit trees. Currently state-of-the-art tree-part segmentation are dominated by fully convolutional networks (FCN). Our previous study used a VGG16-based FCN to segment guava branches with an intersection-over-union (IoU) of 47.3% and an average running time of 0.165 s (Lin et al., 2019). Furthermore, we employed Mask R-CNN to detect and segment guava branches simultaneously, and obtained 51.8% F1 score at a speed of 0.159 s per image (Lin et al., 2021b). Unfortunately, slender branches were found difficult to recognize. Li et al. deployed DeepLabV3 with Xception65 as the backbone to recognize litchi branches and fruits, and accomplished a mean IoU (mIoU) of 78.46% at a speed of 0.6 s (Li et al., 2020). Majeed et al. (2020) used a VGG16-based SegNet to segment tree trunk, branch and trellis wire, and achieved a boundary-F1 score of 0.93, 0.89, and 0.91, respectively. Zhang et al. employed DeepLabV3+ with a lightweight backbone ResNet18 to identify apple tree trunks and branches. The IoUs for trunks and branches were 63 and 40%, respectively, and the average running time was 0.35 s per image (Zhang et al., 2021). Chen et al. (2021) applied a ResNet50-based DeepLabV3, a ResNet34-based U-Net and Pix2Pix to segment occluded branches, respectively, and found that DeepLabV3 outperformed the other models in terms of mIoU, binary accuracy and boundary F1 score. Boogaard et al. (2021) segmented cucumber plants into eight parts by using a point cloud segmentation network PointNet++ and obtained 95% mIoU. Wan et al. developed an improved YOLOV4 to detect branch segments, applied a thresholding segmentation method to remove background, and used a polynomial fit to reconstruct the branches. The detection F1 score was 90%, and the running speed was 22.7 frames per second (FPS) (Wan et al., 2022). Because manually annotating a large empirical dataset is time-consuming and costly, Barth et al. trained DeepLabV2 with VGG16 as the backbone on a large synthetic dataset and then fine-tuned DeepLabV2 on a small empirical dataset. The final network categorized pepper plants into seven different parts with a mIoU of 40% (Barth et al., 2019). Furthermore, Barth et al. (2020) deployed a cycle generative adversarial network to generate realistic synthetic images to train DeepLabV2 and obtained 52% mIoU. Although the approaches mentioned above produce encouraging results, they are typically computationally inefficient since they employ very deep backbones to encode both low-level and high-level features. How to strike a balance between real-time performance and accuracy is a key problem that needs to be solved.

Recently, some efforts have been made to develop real-time segmentation networks. These efforts can be roughly divided into two categories. The first category uses existing lightweight backbones to reduce computation. Howard et al. (2019) developed a shallow segmentation head and appended it to the top of MobileNetV3, and achieved a mIoU of 72% with only 1.98 million multiply-accumulate operations on Cityscapes dataset. Hu et al. proposed a fast spatial attention module to enhance the features encoded by ResNet34, used a simple decoder to merge the features, and achieved 75.5 mIoU at 58 FPS on the Cityscapes dataset (Hu P. et al., 2020). Another category uses customized lightweight backbones to speed up the network inference. Yu et al. proposed a novel network termed BiSeNetV2, which uses a semantic branch with narrow channels and deep layers to generate high-level semantics, applies a detail branch with wide channels and shallow layers to obtain low-level details, and combines these features to predict a segment map. It achieves 72.6% mIoU on the Cityscapes dataset with a speed of 156 PFS (Yu et al., 2021). Gao (2021) proposed a fast backbone that consists of many dilated block structures and used a shallow decoder to output the segmentation. The network achieves 78.3 mIoU at 30FPS on the Cityscapes dataset. Overall, the first category is more attractive, because it utilizes exiting backbones to extract semantic features and hence allows us to focus on more important modules such as decoder.

The objective of this study is to develop a real-time and accurate tree-part segmentation network so that the robots can avoid the obstacles during harvesting. Specifically, a state-of-the-art lightweight backbone is deployed to capture the low-level and high-level features. And then, an attention module is proposed to enhance informative channels and locations in the above features. Subsequently, these features are fused together by a feature aggregation module. The final feature is processed by a segmentation head to output a segment map. A comprehensive experiment is performed to evaluate the proposed tree-part segmentation network.

The contribution of the study is listed as follows:

(1) A tree-part dataset containing 891 RGB images is provided, where each image is annotated on a per-pixel level manually.
(2) A real-time tree-part segmentation is proposed by improving an FCN with channel and spatial attention.
(3) The developed network achieves impressive results. Specifically, when using MobileNetV3-Large as the backbone, the network achieves an IoU of 63.33, 66.25, and 93.12% for the branches, fruits and background, respectively, at a speed of 36 FPS.

Materials and methods

In this section, the data used for this research, including data acquisition, split and annotation, is presented in section 2.1. The developed tree-part segmentation network is introduced in section 2.2. Section 2.3 explains the evaluation criteria used to measure the performance of the developed network.

Data

Data acquisition

The data acquisition site is located in a commercial guava orchard on Haiou Island, Guangzhou, China. The guava species is carmine. There is 3.1 m between two neighboring rows and 2.5 m between two neighboring trees in each row. A low-cost depth camera RealSense D435i is used to capture images, which can simultaneously generate RGB and depth images. This study only uses RGB images, which have a resolution of 480 pixels by 640 pixels. The images were taken on September 24, 2021 between 12:00 and 16:00, just in time for the guava harvest. The day was sunny with a temperature range of 30–34°C. During image acquisition, the camera was held by hand and moved along the path between two rows. The distance between camera and guava tree was about 0.6 m. A total of 41,787 images were acquired. Because adjacent images look similar and may have little effect on network training, a subset of the images were sampled uniformly which comprises 891 images. **Figure 1A** shows a captured image.

Data split

These 891 RGB images were divided into a test and training set. The test set contains the first 30% of the images, and the training set contains the last 70% of the images. This partitioning approach keeps the data sets independent and therefore better examines the generalization performance of the network.

Data annotation

Because branches and fruits will prevent the robots from getting close to the targets, they should be annotated to enable the network to recognize them. Each pixel on the images in the training and test sets was annotated as a branch, fruit, or background class using the open-source annotation program LabelMe (Russell et al., 2008). A visual example is shown in **Figure 1B**. It is worth noting that per-pixel label annotation is very time-consuming, and we spent almost 2 months to accomplish the annotation task.

Tree-part segmentation network

This section illustrates the proposed tree-part segmentation network in detail. An efficient network backbone for capturing low-level and high-level features is introduced in Section 2.2.1. The proposed channel and spatial attention module for boosting meaningful features is elaborated in Section 2.2.2. Section 2.2.3 describes the multi-level feature aggregation module for

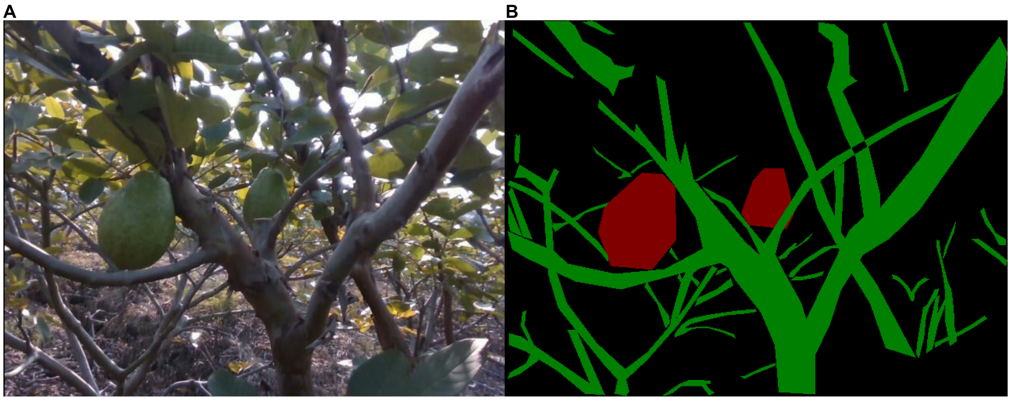


FIGURE 1
Image example. **(A)** A guava tree. **(B)** Different parts of the guava tree, where the red, green, and black regions represent the fruit, branch, and background, respectively.

fusing low-level details and high-level semantics. Section 2.2.4 introduces the segmentation head, and Section 2.2.5 presents the network architecture.

Backbone

To realize real-time segmentation and thus enable the harvesting robots to work efficiently, an efficient neural network MobileNetV3 (Howard et al., 2019) is employed as the segmentation network backbone. MobileNetV3 builds on the latest techniques such as depth-wise separable convolution, inverted bottleneck (Sandler et al., 2018) and squeeze-excitation network (Hu J. et al., 2020), and has been widely deployed in mobile applications. There are many layers outputting feature maps of the same resolution, and these layers are considered to be at the same stage. MobileNetV3 has five stages. Let $\{C_2, C_3, C_4, C_5\}$ denote the outputs of the last layer of stage 2, stage 3, stage 4, and stage 5. Typically, the output of shallow stage such as C_2 contains low-level information but with limited semantics, while that of deep stage such as C_5 contains high-level semantics but with low resolution. These low-level details and high-level semantics can be combined to achieve high accuracy segmentation (Yu et al., 2021). Therefore, they are utilized in this study.

Because MobileNetV3 is primitively designed to output 1,000 classes for ImageNet (Russakovsky et al., 2015), the last few layers have many channels, which may be redundant for our task. In this study, the last layer in stage 5 is directly excluded. We discover that this modification can improve the segmentation accuracy and speed. Additionally, it is a common practice to place atrous convolution in the last few stages of the backbone to generate dense feature maps, which can effectively increase the segmentation accuracy (Chen et al., 2018; Howard et al., 2019). However, when we developed the network model in this paper, we found that the atrous convolution harmed the performance of our network. Hence, we do not use it in the backbone.

Global context information can reduce the probability of misclassification. Pyramid pooling module (PPM) (Zhao et al., 2016) is a practical technique to generate global context information, which uses four different scales of global average pooling layers to enlarge the network receptive fields, up-samples the resulting feature maps so that they have the same size as the original feature map by bilinear interpolation, and then concatenates them as the final global context information. PPM is attached at the top of MobileNetV3.

Channel and spatial attention module

Formally, $\{C_2, C_3, C_4, C_5\}$ encode different levels of channel and spatial information. Not every channel offers useful information. Channel attention mechanism (Roy et al., 2018; Woo, 2018; Hu J. et al., 2020) can be used to recalibrate these feature maps to focus on useful channels, thereby increasing the representation power. Note that the squeeze and excitation attention block of MobileNetV3 serves to refine some intermediate layers, whereas the channel attention mechanism here only serves to refine the output of the last layer of each stage. Besides, the pixel-wise spatial information is more important for semantic segmentation. Therefore, the feature maps can be further recalibrated along space using spatial attention mechanism, making them more informative spatially (Roy et al., 2018; Woo, 2018). To this effect, a channel and spatial attention module (CSAM) is proposed, which consists of a channel attention module and a spatial attention module. CSAM is detailed as follows.

The channel attention module is developed by the inspiration of Howard et al. (2019) to strengthen useful channels and weaken useless channels. Let $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ denote a feature map, where H and W are the spatial height and width, and C is the number of channels. A global average pooling layer is first performed on $\boldsymbol{X}$, resulting a vector $\mathbf{u} \in \mathbb{R}^{C}$ with its k^{th} element:

$$u_k = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{u}(h, w, k) \tag{1}$$

Vector $\mathbf{u}$ is then used to generate a gate vector $\boldsymbol{g}$ by employing a gating mechanism:

$$\mathbf{g} = \sigma(\mathbf{W}_1 \mathbf{u}) \tag{2}$$

where σ refers to the sigmoid function, $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ is a learnable tensor, and r is a reduction ratio using for limiting model complexity. Gate vector $\boldsymbol{g}$ measures the usefulness of the channels, which is used to recalibrate $\boldsymbol{X}$:

$$\mathbf{X}_c = \mathbf{g} \bigotimes \delta(\mathbf{W}_2 * \mathbf{X}) \tag{3}$$

where $\bigotimes$ denotes the channel-wise multiplication, δ is the ReLu function, $*$ refers to convolution, $\mathbf{W}_2 \in \mathbb{R}^{1 \times 1 \times C \times \frac{C}{r}}$ denotes the filter kernel, and $\mathbf{X}_c \in \mathbb{R}^{H \times W \times \frac{C}{r}}$ is the projection of $\boldsymbol{X}$. Equation 3 not only depicts the interdependencies between the channels of X, but also highlights the useful channels while downplaying the useless ones.

In order to fully exploit the spatial information of the feature map, the spatial attention module developed by Roy et al. (2018) is deployed. Specifically, a gate map $\mathbf{G} \in \mathbb{R}^{H \times W}$ is first generated *via* squeezing the feature map along its channel dimension and employing a sigmoid function:

$$\mathbf{G} = \sigma(\mathbf{W}_3 * \mathbf{X}_c) \tag{4}$$

where $\mathbf{W}_3 \in \mathbb{R}^{1 \times 1 \times \frac{C}{r} \times 1}$ is the filter kernel. Then gate map $\boldsymbol{G}$ is used to rescale the feature map:

$$\mathbf{X}_s = \mathbf{G} \bigotimes \mathbf{X}_c \tag{5}$$

where $\bigotimes$ denotes the element-wise multiplication. Equation 5 makes the network focus on important spatial locations and ignore useless ones.

The architecture of CSAM is illustrated in **Figure 2**. CSAM is appended on C_2, C_3, C_4 and the output of PPM, and

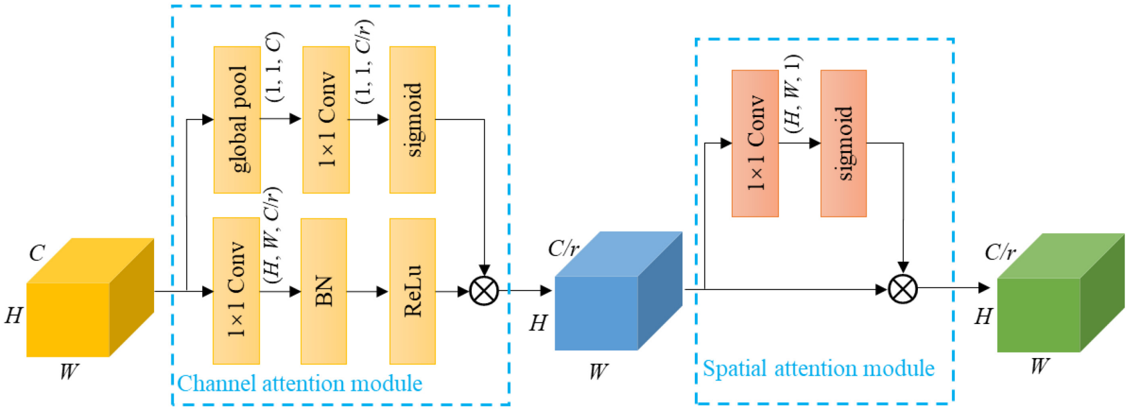


FIGURE 2
Design details of CSAM. Note that *Conv* is convolutional operation, and BN is batch normalization; 1 × 1 represents the kernel size, *H* × *W* × *C* and *H* × *W* × *C*/*r* denote the tensor shape (height, width, and depth); the first ⊗ refers to channel-wise multiplication, and the second ⊗ is element-wise multiplication.

the corresponding reduction ratios are set to {1, 1, 2, 4} for MobileNetV3-Large and {1, 1, 1, 2} for MobileNetV3-Small. The resulting feature maps are denoted as $\{G_2, G_3, G_4, G_5\}$. It is worth noting that CASM is attached to PPM and not C_5 simply because PPM itself contains C_5. The work (Roy et al., 2018) also proposes a similar attention module. CSAM differs in introducing a reduction ratio to reduce the module complexity, and information goes through the two modules in an orderly manner, which progressively filters out useless information.

Let us consider an input feature map of C channels. The channel attention module introduces $\frac{2C^2}{r}$ new weights, while the spatial attention module introduces $\frac{C}{r}$ weights. So, a CASM brings a total of $\frac{2C^2+C}{r}$ parameters. Because the feature maps of MobileNetV3 have relatively few channels, these extra parameters only add a small amount of computation to the backbone.

Feature aggregation module

Typically, thin branches are harder to segment than thick branches, because detailed information is easily lost when the output stride is increased. This problem can be alleviated by fusing feature maps from different layers, such as $\{G_2, G_3, G_4, G_5\}$. A simple variant of feature pyramid network (FPN) (Lin et al., 2016) is used to gradually up-samples and merges the feature maps from deepest feature maps to shallow ones. As shown in **Figure 3**, our FPN variant first appends a 1 × 1 convolutional layer on the coarsest feature map G_5 to reduce its channel dimension, up-samples G_5 by a factor of 2, and then merges G_5 with its corresponding bottom-up map G_4 by element-wise addition. This process is repeated until the finest feature map is generated. A 3 × 3 convolutional layer is appended on each merged feature map to generate the final feature map with a fixed output dimension of 48. Here, batch normalization and ReLu are adopted after each convolution, which are omitted for simplifying notations. On this basis, these feature maps are concatenated. Because lower-level feature maps may have large values than higher-level ones, which probably destabilizes network training, the concatenated features should be normalized carefully. To this effect, a L_2 normalization layer (Liu et al., 2015) is performed on the concatenated features. Specifically, let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_C)$ be the concatenated features, and C is the number of channels. $\boldsymbol{X}$ is normalized with the following equation:

$$\mathbf{x}_c = \gamma_c \frac{\mathbf{x}_c}{||\mathbf{x}_c||_2} \tag{6}$$

where $||\cdot||_2$ means the L_2 norm; $c = 1, \ldots, C$; and γ_c is a learnable scaling parameter, which can avoid the resulting features being too small and hence promotes network learning. In experiments, the initial value of γ_c is set to 1. Subsequently, a CSAM with reduction ratio of K is attached after the L_2 normalization layer to further refine the feature map, where K refers to the number of feature maps fused. **Figure 3** shows the architecture of the proposed FAM.

Segmentation head

The segmentation head is used to output a segment map of the same size as the input RGB image, which is N-channeled with N being the number of classes. In this study, N equals to 3. **Figure 4** shows the segmentation head, which consists of a 3 × 3 convolution layer, a batch normalization layer, a ReLU activation, a 1 × 1 convolution layer and an up-sampling operation *via* bilinear interpolation.

Network architecture

The overall architecture is shown in **Figure 5**. MobileNetV3 forms the backbone network with PPM attached on the top to capture global contextual information. Feature maps from the

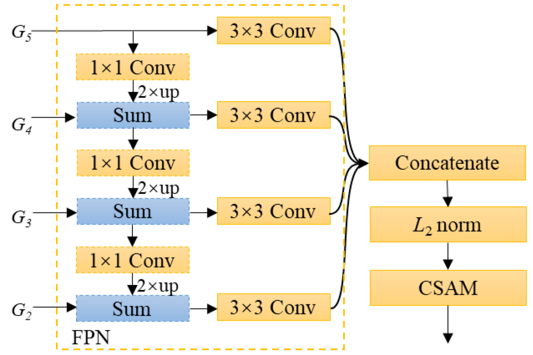


FIGURE 3
Design details of FAM. Note that *up* refers to up-sampling by bilinear interpolation.

last layers of stage 2, stage 3, stage 4, and PPM are refined by CSAM and then used as input to FAM to produce a feature map containing low-level details and high-level semantics. The output of FAM is processed by the segmentation head to make the final semantic segmentation.

The tree-part segmentation network is trained in an end-to-end manner to minimize a cross-entropy loss defined on the output of the segmentation head. To stabilize network training, an auxiliary segmentation head is inserted after the output of stage 3, and an auxiliary cross-entropy loss with weight 0.4 is added to the final loss (Zhao et al., 2016), as shown in **Figure 5**. This auxiliary segmentation head is only used in the training phase and removed in the inference phase. Furthermore, a L_2 regularization with weight 5e^{-4} on the parameters of the network except the backbone are added to the final loss to alleviate network over-fitting. Note that because this study uses a pre-trained MobileNetV3 on ImageNet as the backbone, we do not place the L_2 regularization on the parameters of the backbone.

Segmentation evaluation

To evaluate the accuracy performance of the tree-part segmentation network, three commonly used metrics are used: IoU, mIoU, and pixel accuracy (PA). For the sake of explanation, let N denotes the total number of classes, and p_{ij} denote the number of pixels that belong to class i but are predicted to be class j. Obviously, p_{ii}, p_{ij} and p_{ji} represent the number of true positives, false negatives, and false positives, respectively. IoU is the ratio between the intersection and union of the ground true and predicted segmentation, and can be calculated by dividing true positives by the sum of false positives, false negatives and true positives. For class i, its IoU is computed as follows:

$$IoU_i = \frac{p_{ii}}{\sum_{j=0}^{N-1} p_{ij} + \sum_{j=0}^{N-1} p_{ji} - p_{ii}} \tag{7}$$

mIoU is an improved IoU which computes the IoU value for each class and then averages them:

$$mIoU = \frac{1}{N}\sum_{i=0}^{N-1} \frac{p_{ii}}{\sum_{j=0}^{N-1} p_{ij} + \sum_{j=0}^{N-1} p_{ji} - p_{ii}} \tag{8}$$

PA measures the network recall ability. It calculates a ratio between the amount of true positives and the total number of pixels:

$$PA = \frac{\sum_{i=0}^{N-1} p_{ii}}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p_{ij}} \tag{9}$$

To measure the real-time performance of the developed network, three metrics are utilized: floating point operations per second (FLOPs), FPS, and number of parameters. Note that FPS is determined by counting how much RGB images can be processed per second in the inference phase.

Experimental setup

Implementation details

The developed network is programmed in Pytorch and runs on a computer with Windows 10 system, 32 GB RAM, Intel i9-11900K CPU, and NVIDIA GeForce RTX 3080 GPU. The backbone is pre-trained on ImageNet, and other parameters are initialized using the default initialization method in Pytorch. Standard Adam is used to minimize the loss function, and "cosine" learning scheduler (Loshchilov and Hutter, 2016) is

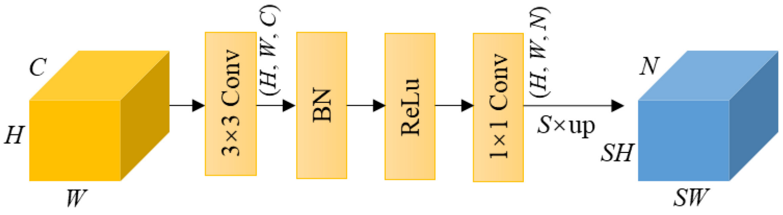


FIGURE 4
Illustration of the segmentation head. Note that *S* is the scale ratio of up-sampling, and *N* is the number of classes.

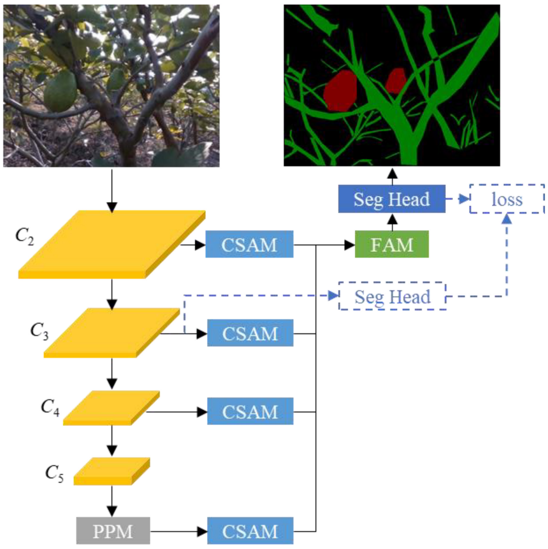


FIGURE 5
Overview of the tree-part segmentation network, where three-dimensional blocks represent feature maps and two-dimensional blocks refer to convolutional modules.

used to adjust learning rate, where initial learning rate is set to $1e^{-4}$. The network is trained on the train set, and 150 training epochs are used with a mini-batch size of 12. To avoid network over-fitting, the following data augmentation methods are implemented during training: horizontal flipping, vertical flipping, random rotation within the range of [−45°, 45°], random scale within the rage of [0.8, 1.2], and randomly changing the hue, saturation and value of the input image.

Ablation study

This section performs the ablation study to validate the effectiveness of each module in our network. In the following experiments, MobileNetV3-Large is used as the backbone, and the segmentation models are trained on our training set and evaluated on our test set. The ablation study is detailed as follows:

(1) Ablation for backbone. Placing atrous convolution in the last stage of the backbone can preserve the details, which has been widely utilized in semantic segmentation (Chen et al., 2018; Howard et al., 2019). However, it is unclear whether atrous convolution can improve the segmentation accuracy of our network. In addition, whether removing the last layer of stage 5 of the backbone network will improve efficiency and accuracy. Experiments are conducted to answer these questions.

(2) Ablation for feature aggregation. High-level features contain semantic information but with limited details, while low-level features contain detailed information but with limited semantics. Fusing these features can improve segmentation accuracy. However, it is unclear which low-level and high-level features should be fused. We re-implement the network with different combinations of the low-level and high-level features, and find the best combination through experiments.

(3) Ablation for auxiliary segmentation head. Auxiliary segmentation head has been widely used in semantic segmentation (Zhao et al., 2016; Yu et al., 2021). We insert the auxiliary segmentation head to different stages of the backbone in the training phase and reveal which position is most important.

Comparison with existing methods

To evaluate the accuracy and real-time performance of the developed network, a comparison experiment is performed. MobileNetV3-Large and MobileNetV3-Small are used as the backbone of our network. Four state-of-the-art networks are used for comparisons: DeepLabV3 (Chen et al., 2017), DeepLabV3+ (Chen et al., 2018), LR-ASPP (Howard et al., 2019), and FANet (Hu P. et al., 2020). For the sake of

TABLE 1 Ablations on the backbone and feature aggregation module.

Row	AC	R	NF	IoU (%)			mIoU (%)	PA (%)	FPS	#Params	FLOPs
				Branch	Fruit	Background					
1	✓	x	4	63.37	66.67	93.05	74.03	93.76	32.84	6.9M	3.48B
2	✓	✓	4	62.51	67.05	93.18	74.25	93.87	33.85	5.7M	3.08B
3	x	x	4	63.40	66.03	93.26	74.23	93.95	33.80	6.9M	2.44B
4	x	✓	4	63.33	66.25	93.12	74.23	93.84	36.00	5.7M	2.36B
5	x	✓	3	58.72	63.14	92.20	71.35	92.96	34.67	5.7M	1.66B
6	x	✓	2	49.74	61.16	90.72	67.21	91.49	34.36	5.7M	1.46B

AC, Apply atrous convolution in the last block of the backbone; R, Remove the last layer in stage 5 of the backbone; NF, Number of feature maps fused in FAM. When NF = 4, $\{G_2, G_3, G_4, G_5\}$ are fused. When NF = 3, $\{G_3, G_4, G_5\}$ are fused. When NF = 2, $\{G_4, G_5\}$ are fused. M and B represent million and billion, respectively.

TABLE 2 Ablations on the auxiliary segmentation head, which is inserted after the output of different stages in the backbone.

Stage	IoU (%)			mIoU (%)	PA (%)
	Branch	Fruit	Background		
2	62.45	65.32	92.95	73.58	93.68
3	63.33	66.25	93.12	74.23	93.84
4	64.04	61.96	93.33	73.11	94.02
5	63.07	61.98	93.23	72.76	93.92

comparison, DeepLabV3, DeepLabV3+ and LR-ASPP use MobileNetV3-Large as the backbone, and apply the atrous convolution to the last block of MobileNetV3-Large to generate denser feature maps. FANet uses ResNet18 as the backbone as suggested by Hu P. et al. (2020). All of the comparison networks are implemented in Pytorch and trained according to the strategy described in section 3.1. Our network and the comparison networks are evaluated on the test set, and quantitative results including IoU, mIoU, PA, FPS, and FLOPs are reported and discussed.

Results and discussion

Ablation study

Table 1 lists the results of different configurations of the backbone. As shown in the table, we observed that (1) when not employing the atrous convolution in the last block of the backbone to extract dense features, the mIoU and PA slightly improved by 0.20 and 0.19%, respectively, while being faster (row 1 vs. row 3), (2) removing the last layer in stage 5 of the backbone did not decrease the IoU and PA while being slightly faster (row 1 vs. row2, row 3 vs. row 4), and (3) when not employing the atrous convolution and removing the last layer in stage 5, the network obtained similar accuracies while being significant faster than its variants (row 4 vs. row 1, 2, and 3). These results indicate that the atrous convolution was not necessary for our task, and the MobileNetV3 backbone contained redundant layers which should be excluded.

Aggregating different levels of features has varying effects on the network performance, as shown in Table 1. Fusing $\{G_2, G_3, G_4, G_5\}$ performed better than fusing $\{G_3, G_4, G_5\}$ and $\{G_4, G_5\}$ by 2.88 and 7.02%, respectively, in terms of mIoU, and only required a few more computation. This illustrates that the network performance could benefit from fusing as many features as possible. In this study, we fused $\{G_2, G_3, G_4, G_5\}$ to improve the network accuracy.

Table 2 shows the effect of different positions to place the auxiliary segmentation head. As can be seen, inserting the auxiliary segmentation head into the output of stage 3 outperformed that of stage 2, stage 4 and stage 5 by 0.65, 1.12, and 1.47%, respectively, in terms of mIoU, and slightly underperformed that of stage 4 and stage 5 by 0.17 and 0.08%, respectively, in terms of PA. Therefore, we chose to attach the auxiliary segmentation head to the output of stage 3.

Comparison with existing methods

Table 3 lists the accuracy and real-time performance of the proposed network and comparison methods. Overall, our network with MobileNetV3-Large as the backbone outperformed LR-ASPP, DeepLabV3, DeepLabV3+, and FANet in terms of the accuracy metrics, which validated the effectiveness of the proposed modules. Furthermore, our network performed faster than DeepLabV3, DeepLabV3+ and FANet in terms of FLOPs, likely because DeepLabV3 and DeepLabV3+ applied a very time-consuming atrous spatial pyramid pooling module to encode context information, and FANet used a relatively large backbone. Surprisingly, there was little difference in FPS between our network and the comparison networks, probably because the depth-wise convolution in MobileNets and the multi-branch design in ResNet increased the memory access cost, affecting the inference speed (Ding et al., 2021). Conclusively, the proposed network with MobileNetV3-Large as the backbone was more accurate than the comparison methods while being fast.

Additionally, our network with MobileNetV3-Small as the backbone had slightly lower accuracy than DeepLabV3+, but higher accuracy than LR-ASPP, DeepLabV3, and FANet. Moreover, this network achieved the best real-time performance. In other words, when MobileNetV3-Small was

TABLE 3 Accuracy and real-time performance of the proposed network and comparison methods on test set.

Methods	Backbone	IoU (%)			mIoU (%)	PA (%)	FPS	#Params	FLOPs
		Branch	Fruit	Background					
Ours	MobileNetV3-Large	63.33	66.25	93.12	74.23	93.84	36.00	5.7M	2.36B
Ours	MobileNetV3-Small	60.62	61.05	92.82	71.50	93.52	37.91	2.7M	1.18B
LR-ASPP	MobileNetV3-Large	60.05	58.60	92.85	70.50	93.52	36.67	5.7M	2.37B
DeepLabV3	MobileNetV3-Large	56.34	58.82	92.14	69.11	92.85	35.78	13.5M	11.58B
DeepLabV3+	MobileNetV3-Large	62.59	61.05	93.36	72.33	94.00	31.52	14.2M	35.73B
FANet	ResNet18	54.71	57.57	92.25	68.17	92.97	36.65	13.8M	6.93B

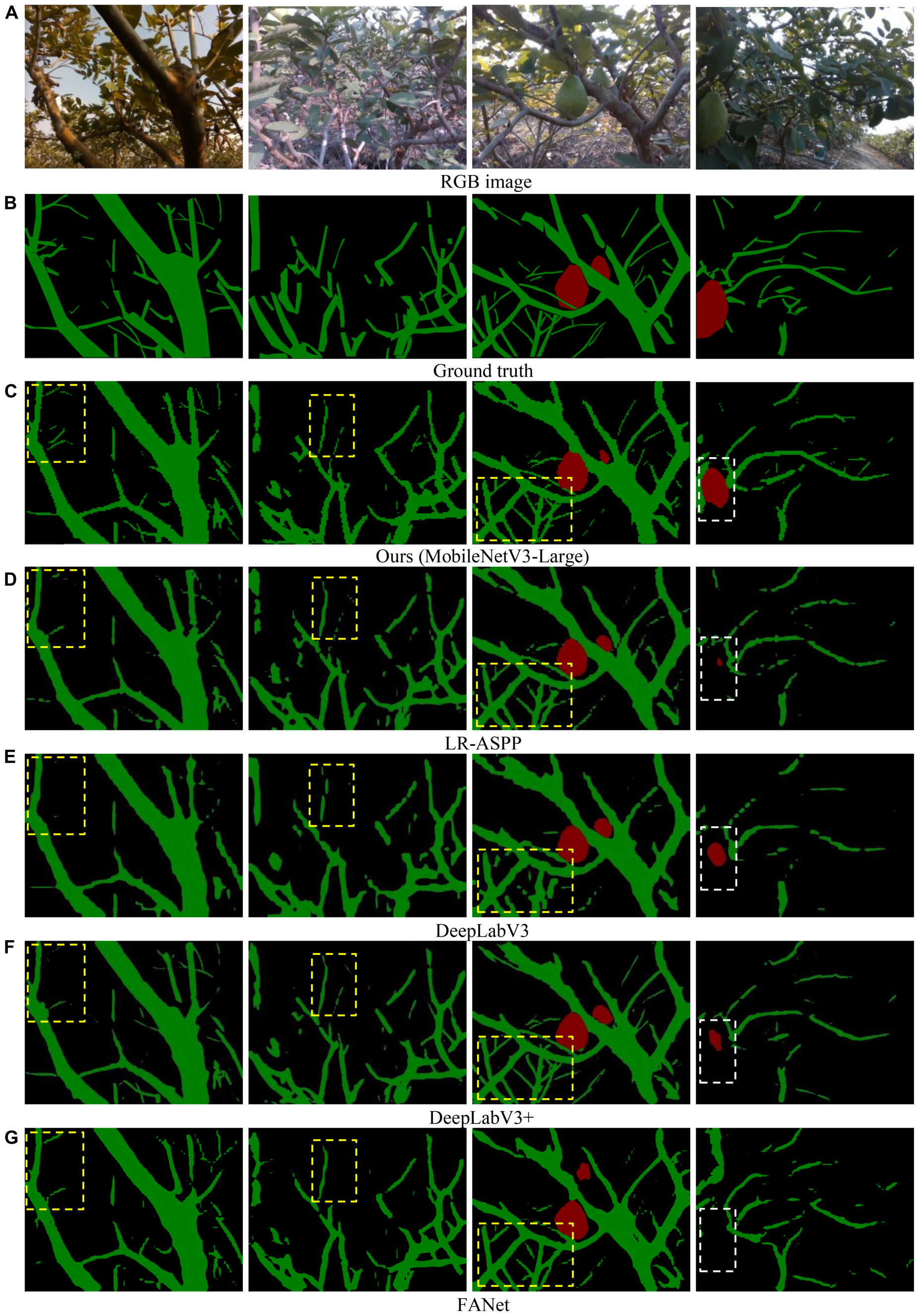


FIGURE 6
Visual examples illustrating results of our network and comparison networks. **(A)** RGB image. **(B)** Ground truth. **(C)** Ours (MobileNetV3-Large). **(D)** LR-ASPP. **(E)** DeepLabV3. **(F)** DeepLabV3+. **(G)** FANet.

used as the backbone, the proposed network was the fastest among the comparison networks, but somewhat less accurate.

Our network achieved a large IoU value for the background, probably because the background dominated the images, making the network pay more attention to the background. This problem can be alleviated by reshaping the loss function by down-weighting the background and up-weighting other objects (Ronneberger et al., 2015). Besides, the IoU value of the branch class was lower than that of the fruit class. A possible reason was that some branches were very thin and hence their detailed information was easy to be lost, making them hard to segment. Although we have fused multi-layer features to solve such a problem, MobileNetV3 was too lightweight to provide enough features. Future work will consider adding a detail branch (Yu et al., 2021) to the backbone to extract detailed information.

Some qualitative results were shown in **Figure 6**. Visually, our network was more accurate in tree-part segmentation. Specifically, the developed network could capture the details of most thin branches, whereas the comparison networks struggled to segment the thin branches, as shown in the yellow boxes in columns 1–3 of **Figure 6**. Besides, our network outperformed the comparison networks in the recognition ability of fruits, as shown in the while boxes in column 4 of **Figure 6**. The results validate the effectiveness of the developed attention module and feature aggregation module. Although most of the branches were identified, some thin branches seemed to be difficult to identify. In robotic harvesting, the thin branches might clog the end effector, causing shear failure. Therefore, future work will focus on improving the segmentation accuracy of thin branches. A relevant video can be found at: https://www.bilibili.com/video/BV1nS4y147wa/?vd_source=d082953b9cfe065d2d003486f259e84f.

Conclusion

This study aimed to develop a tree-part segmentation network that can segment fruits and branches efficiently and accurately for harvesting robots to avoid obstacles. Experimental results validated that the proposed network can accomplish the research objective. Some specific conclusions drawn from the study were given as follows:

(1) A tree-part dataset was collected. The dataset consists of 891 RGB images captured in the fields. Each image is manually annotated in a per-pixel fashion, which took us almost 2 months to label. To the best of our knowledge, this is the first tree-part dataset used to help harvesting robots avoid obstacles.

(2) A tree-part segmentation network was developed, which consists of four components: a lightweight backbone, CASM, FAM, and segmentation head. Here, CASM was used to enhance informative channels and locations in the feature maps, and FAM was designed to fuse multi-layer feature maps to improve the segmentation accuracy. Experiments on the test set shows that when using MobileNetV3-Large as the backbone, the network achieved an IoU of 63.33, 66.25, and 93.12% for the branches, fruits and background, respectively, at a speed of 2.36 billion FLOPs. These performance values validates that the network could segment tree parts efficiently and quite accurately. However, the IoU value of the branch class was the lowest, probably because the max-pooling operations in the backbone lost the detailed information of the thin branches, thus making the thin branches difficult to segment.

The proposed network could be transferred to segment other fruits by fine-tuning on new datasets. Future research will add two more classes (soft branch and hard branch) to the current dataset to allow harvesting robots to push away soft branches and avoid hard ones for better fruit picking. Furthermore, future work will attempt to add a detailed path in the backbone to preserve the detailed information of the input image, thus improving the accuracy.

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 authors.

Author contributions

GL: methodology, investigation, and writing—original draft. CW: investigation, methodology, and writing—review and editing. YX and MW: writing—review and editing. ZZ: conceptualization and data curation. LZ: methodology and supervision. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Laboratory of Lingnan Modern Agriculture Project (Grant No. NZ2021038), the National Natural Science Foundation of China (Grant No. 32101632), the Basic and Applied Basic Research Project of Guangzhou Basic Research Plan (Grant No. 202201011310), and the Science and Technology Program of Meizhou, China (Grant No. 2021A0304004).

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

Amatya, S., Karkee, M., Gongal, A., Zhang, Q., and Whiting, M. D. (2016). Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. *Biosyst. Eng.* 146, 3–15. doi: 10.1016/j.biosystemseng.2015.10.003

Barth, R., Hemming, J., and Van Henten, E. J. (2020). Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation. *Comput. Electron. Agric.* 173:105378. doi: 10.1016/j.compag.2020.105378

Barth, R., IJsselmuiden, J., Hemming, J., and Henten, E. J. V. (2018). Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Comput. Electron. Agric.* 144, 284–296. doi: 10.1016/j.compag.2017.12.001

Barth, R., IJsselmuiden, J., Hemming, J., and Van Henten, E. J. (2019). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Comput. Electron. Agric.* 161, 291–304. doi: 10.1016/j.compag.2017.11.040

Boogaard, F. P., van Henten, E. J., and Kootstra, G. (2021). Boosting plant-part segmentation of cucumber plants by enriching incomplete 3d point clouds with spectral data. *Biosyst. Eng.* 211, 167–182. doi: 10.1016/j.biosystemseng.2021.09.004

Chen, L. C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv* [Preprint]. arXiv:1706.05587.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). *Encoder-decoder with atrous separable convolution for semantic image segmentation*, eds V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer International Publishing). doi: 10.1007/978-3-030-01234-2_49

Chen, Z., Ting, D., Newbury, R., and Chen, C. (2021). Semantic segmentation for partially occluded apple trees based on deep learning. *Comput. Electron. Agric.* 181:105952. doi: 10.1016/j.compag.2020.105952

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021). "Repvgg: Making vgg-style convnets great again," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville). doi: 10.1109/CVPR46437.2021.01352

Gao, R. (2021). Rethink dilated convolution for real-time semantic segmentation. *arXiv* [Preprint]. arXiv:2111.09957.

Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., et al. (2019). Searching for mobilenetv3. *arXiv* [Preprint]. arXiv:1905.02244. doi: 10.1109/ICCV.2019.00140

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, P., Perazzi, F., Heilbron, F. C., Wang, O., Lin, Z., Saenko, K., et al. (2020). Real-time semantic segmentation with fast attention. *arXiv* [Preprint]. arXiv:2007.03815. doi: 10.1109/LRA.2020.3039744

Ji, W., Qian, Z., Xu, B., Tao, Y., Zhao, D., and Ding, S. (2016). Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot. *Optik* 127, 11173–11182. doi: 10.1016/j.ijleo.2016.09.044

Li, J., Tang, Y., Zou, X., Lin, G., and Wang, H. (2020). Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. *IEEE Access* 8, 117746–117758. doi: 10.1109/ACCESS.2020.3005386

Lin, G., Tang, Y., Zou, X., Xiong, J., and Li, J. (2019). Guava detection and pose estimation using a low-cost rgb-d sensor in the field. *Sensors* 19:428. doi: 10.3390/s19020428

Lin, G., Zhu, L., Li, J., Zou, X., and Tang, Y. (2021a). Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning. *Comput. Electron. Agric.* 188:106350. doi: 10.1016/j.compag.2021.106350

Lin, G., Tang, Y., Zou, X., and Wang, C. (2021b). Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis. *Comput. Electron. Agric.* 184:106107. doi: 10.1016/j.compag.2021.106107

Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2016). Feature pyramid networks for object detection. *arXiv* [Preprint]. arXiv:1612.03144. doi: 10.1109/CVPR.2017.106

Liu, W., Rabinovich, A., and Berg, A. C. (2015). Parsenet: Looking wider to see better. *arXiv* [Preprint]. arXiv:1506.04579.

Loshchilov, I., and Hutter, F. (2016). SGDR: Stochastic gradient descent with restarts. *arXiv* [Preprint]. arXiv:1608.03983.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., et al. (2020). Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* 170:105277. doi: 10.1016/j.compag.2020.105277

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *arXiv* [Preprint]. arXiv:1505.04597. doi: 10.1007/978-3-319-24574-4_28

Roy, A. G., Navab, N., and Wachinger, C. (2018). Concurrent spatial and channel squeeze & excitation in fully convolutional networks. *arXiv* [Preprint]. arXiv:1803.02579. doi: 10.1007/978-3-030-00928-1_48

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* 115, 211–252. doi: 10.1007/s11263-015-0816-y

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. Vision* 77, 157–173. doi: 10.1007/s11263-007-0090-8

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv* [Preprint]. arXiv:1801.04381. doi: 10.1109/CVPR.2018.00474

Wan, H., Fan, Z., Yu, X., Kang, M., Wang, P., and Zeng, X. (2022). A real-time branch detection and reconstruction mechanism for harvesting robot via convolutional neural network and image segmentation. *Comput. Electron. Agric.* 192:106609. doi: 10.1016/j.compag.2021.106609

Woo, S. A. P. J. (2018). *CBAM: Convolutional block attention module*. Cham: Springer. doi: 10.1007/978-3-030-01234-2_1

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., and Sang, N. (2021). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vision* 129, 3051–3068. doi: 10.1007/s11263-021-01515-2

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., and Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (r-cnn). *Comput. Electron. Agric.* 155, 386–393. doi: 10.1016/j.compag.2018.10.029

Zhang, X., Karkee, M., Zhang, Q., and Whiting, M. D. (2021). Computer vision-based tree trunk and branch identification and shaking points detection in dense-foliage canopy for automated harvesting of apples. *J. Field Robot.* 38, 476–493. doi: 10.1002/rob.21998

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2016). Pyramid scene parsing network. *arXiv* [Preprint]. arXiv:1612.01105. doi: 10.1109/CVPR.2017.660

TYPE Original Research
PUBLISHED 28 September 2022
DOI 10.3389/fpls.2022.993244

Detection method of wheat spike improved YOLOv5s based on the attention mechanism

Hecang Zang[1,2], Yanjing Wang[3]*, Linyuan Ru[4], Meng Zhou[1,2], Dandan Chen[1,2], Qing Zhao[1,2], Jie Zhang[1,2], Guoqiang Li[1,2]* and Guoqing Zheng[1,2]

[1]Institute of Agricultural Economics and Information, Henan Academy of Agricultural Sciences, Zhengzhou, China, [2]Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology, Ministry of Agriculture and Rural Affairs, Zhengzhou, China, [3]College of Life Sciences, Zhengzhou Normal University, Zhengzhou, China, [4]College of Computer and Information Engineering, Henan Normal University, Xinxiang, China

OPEN ACCESS

EDITED BY
Wanneng Yang,
Huazhong Agricultural University,
China

REVIEWED BY
Weijuan Hu,
Institute of Genetics
and Developmental Biology (CAS),
China
Gongfa Li,
Wuhan University of Science
and Technology, China
Yang Lu,
Heilongjiang Bayi Agricultural
University, China

*CORRESPONDENCE
Yanjing Wang
wangyanjing1230@163.com
Guoqiang Li
moonlgq1984@163.com

SPECIALTY SECTION
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

RECEIVED 13 July 2022
ACCEPTED 30 August 2022
PUBLISHED 28 September 2022

CITATION
Zang H, Wang Y, Ru L, Zhou M,
Chen D, Zhao Q, Zhang J, Li G and
Zheng G (2022) Detection method
of wheat spike improved YOLOv5s
based on the attention mechanism.
Front. Plant Sci. 13:993244.
doi: 10.3389/fpls.2022.993244

In wheat breeding, spike number is a key indicator for evaluating wheat yield, and the timely and accurate acquisition of wheat spike number is of great practical significance for yield prediction. In actual production; the method of using an artificial field survey to count wheat spikes is time-consuming and labor-intensive. Therefore, this paper proposes a method based on YOLOv5s with an improved attention mechanism, which can accurately detect the number of small-scale wheat spikes and better solve the problems of occlusion and cross-overlapping of the wheat spikes. This method introduces an efficient channel attention module (ECA) in the C3 module of the backbone structure of the YOLOv5s network model; at the same time, the global attention mechanism module (GAM) is inserted between the neck structure and the head structure; the attention mechanism can be more Effectively extract feature information and suppress useless information. The result shows that the accuracy of the improved YOLOv5s model reached 71.61% in the task of wheat spike number, which was 4.95% higher than that of the standard YOLOv5s model and had higher counting accuracy. The improved YOLOv5s and YOLOv5m have similar parameters, while RMSE and MEA are reduced by 7.62 and 6.47, respectively, and the performance is better than YOLOv5l. Therefore, the improved YOLOv5s method improves its applicability in complex field environments and provides a technical reference for the automatic identification of wheat spike numbers and yield estimation. Labeled images, source code, and trained models are available at: https://github.com/228384274/improved-yolov5.

KEYWORDS

wheat, spike number detection, attention mechanism, deep learning, YOLOv5s

Introduction

Wheat is an important food crop in our country. In 2021, the planting area of wheat will be 22.911 million hectares, and the output will be 134 million tons in our country; China is the largest wheat producer in the world (Sreenivasulu and Schnurbusch, 2012; Ge et al., 2018; Chen et al., 2021; Wen et al., 2022). However, the current COVID-19 epidemic is raging, the domestic and foreign environments are complex and changeable, abnormal weather and natural disasters are frequent, and food security is facing severe challenges (Laborde et al., 2020; FAO, 2021; Ministry of Emergency Management of the People's Republic of China, 2022). The spike number is an important indicator for wheat yield estimation (Zhang et al., 2007; Gou et al., 2016; Zhou et al., 2021). Therefore, wheat spike number detection is the key to predicting and evaluating wheat yield. Timely and accurate acquisition of wheat spike numbers has always been the focus of wheat breeding and cultivation research.

In actual production, the acquisition of wheat spikes mainly includes low-throughput artificial field investigation and high-throughput remote sensing image processing. Artificial field surveys have the disadvantages of strong subjectivity, strong randomness, and lack of uniform standards, which lead to the shortcomings of time-consuming, labor-intensive, and low-efficiency researchers. They cannot obtain statistical results of wheat spikes efficiently and quickly (Kamilaris and Prenafeta-Boldú, 2018). The high-throughput remote sensing image processing is based on the feature fusion of different textures (Ganeva et al., 2022), color features (Grillo et al., 2017), spectral reflectance, and uses machine learning to detect targets in wheat spike images to extract the number of wheat spikes. Zhao et al. (2021) proposed a method based on an improved YOLOv5, which can accurately detect the number of wheat spikes in unmanned aerial vehicle (UAV) images; the average accuracy (AP) of wheat spike detection in UAV images is 94.1%, which is 10.8% higher than the standard YOLOv5, and solves the problem of the wrong detection and missed detection of wheat spikes due to occlusion conditions. Gong et al. (2021) proposed a method of wheat-head detection based on a deep neural network to enhance the speed and accuracy of detection; the mean average precision of the proposed method is 94.5%, and the detection speed is 71 FPS. Li et al. (2022) used a deep-learning algorithm (Faster R-CNN) on red green blue (RGB) images to explore the possibility of image-based detection of spike numbers and its application to identify the loci underlying spike numbers. Xiong et al. (2019) proposed a simple yet effective contextual extension of TasselNet–TasselNetv2, which simultaneously addresses two important use cases in plant counting.

Alkhudaydi et al. (2019a) developed a deep-learning-based analysis pipeline to segment spike regions from complicated backgrounds. Zhao et al. (2022) proposed a deep learning method for oriented and small wheat spike detection (OSWSDet); the AP is 90.5%. Wang Y. D. et al. (2021) proposed an improved EfficientDet-D0 object detection model for wheat ear counting; the counting accuracy of the improved EfficientDet-D0 model reaches 94%, which is about 2% higher than the original model and focuses on solving occlusion. Wang et al. (2019) proposed a field-based high-throughput phenotyping approach using deep learning that can directly measure morphological and developmental phenotypes in genetic populations from field-based imaging. David et al. (2020, 2021) built the Global Wheat Head Detection (GWHD) dataset and released in 2021 a new version of the GWHD dataset, which is bigger, more diverse, and less noisy than the GWHD_2020 version. Yang et al. (2021) proposed an improved YOLOv4 with a spatial and channel attention model was proposed that could enhance the feature extraction capabilities of the network by adding receptive field modules. Fernandez-Gallego et al. (2018) proposed an automatic algorithm for the number of wheat spikes to estimate the number of wheat spikes under field conditions. Lu et al. (2017) developed a smartphone application software to complete the detection and collection of wheat diseased spikes, with an accuracy of 96.6%. Pound et al. (2017) used the deep learning method to calculate the number of wheat spikes through the images of wheat spikes taken under greenhouse conditions. Hasan et al. (2018) and Li et al. (2021) use the R-CNN method to detect, count, and analyze wheat spikes, which has high recognition accuracy, but the detection speed is slow and cannot be deployed in real-time detection equipment. Compared with the above methods, our proposed method has a faster detection speed while improving accuracy than the two-stage target detection method. Compared with other improved YOLO algorithms, we introduce the attention mechanism into the YOLO model to improve the network's ability to extract the target features, rather than relying on data sets. Compared with the traditional image processing methods, the deep learning technology can automatically extract the target features, while the traditional methods mainly rely on manual design features, and the algorithm has no generalization. The extraction ability of unknown features is poor. Therefore, we introduce the attention mechanism into the YOLO model to ensure accuracy and faster detection speed, which lays the foundation for future deployment on mobile devices.

In recent years, with the rapid development of artificial intelligence, deep learning algorithms have been widely used in the industrial field. Huang et al. (2021) determined whether workers meet the standard of wearing helmets by improving the YOLOv3 algorithm. The final result is that the mAP reaches 93.1%. Huang et al. (2022) used the improved single shot multiBox detector (SSD) algorithm to verify the effectiveness of multi-scale feature fusion for small targets. Sun et al. (2022) solved the problems of poor image quality, loss of detail information, and excessive brightness enhancement in the image enhancement process in a low-light environment by improving the multi-scale Retinex and ABC algorithms. Bai et al. (2022) improved the network by combining the target frame recommendation strategy in the SSD algorithm with the frame regression algorithm to improve the detection accuracy of small targets. Weng et al. (2021) proposed an angle network model to accurately estimate the robot picking angle, which improves the accuracy and real-time detection. Gao et al. (2019) applied deep neural networks to hand detection and achieved good results. The deep learning object detection model has made remarkable progress in wheat spike image detection (Madec et al., 2019; He et al., 2020), which is the main technical means for wheat spike recognition and detection counting, and has reached top performance in detection accuracy and speed (Zhou et al., 2018a; Khoroshevsky et al., 2021; Lu et al., 2021; Wang D. et al., 2021). Single-stage algorithms for object detection include SSD (Liu et al., 2016) and the YOLO family, which includes YOLO (Redmon et al., 2016), YOLO9000 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), and YOLOv5 (Ultralytics, 2021). The single-stage detection algorithm is also known as the target detection algorithm based on regression analysis, which regards the target detection problem as a regression analysis problem on target location and category information, which can directly output the detection results through a neural network model. Considering the cost and observational limitations of satellites, ground-based remote sensing, and drones according to the needs of researchers, the use of smartphones has significantly improved the efficiency of wheat spike surveys. However, in the detection of wheat spike images, due to the high density of wheat spike, serious occlusion, and serious cross-overlapping, detection errors and missed detection of the wheat spike are caused. At the same time, due to the large morphological differences between individual wheat spikes and the fact that the color of the wheat spike is consistent with the background, the difficulty and accuracy of wheat spike detection are further increased.

In order to solve the above problems, this paper proposes an improved YOLOv5s target detection method using an attention mechanism for the accurate detection of wheat spikes. This method introduces ECA into the C3 module of the backbone structure of the YOLOv5s network model; GAM is inserted between the neck structure and the head structure; the attention mechanism can more effectively extract feature information and suppress useless information. This method improves the applicability of the YOLOv5s method in complex field environments, which can accurately detect the number of small-scale wheat spikes and better solve the problem of occlusion and overlap of a wheat spike.

Materials and methods

Overview of the test site

The experimental site is located in the regional wheat experiment of the Henan Modern Agriculture Research and Development Base of the Henan Academy of Agricultural Sciences. It is located at 35°0′44″ north latitude and 113°41′44″ east longitude, as shown in **Figure 1**. The climate type is a warm temperate continental monsoon climate, with an annual average temperature of 14.4°C, annual average rainfall of 549.9 mm, and annual sunshine hours of 2300–2600 h. The wheat-corn rotation is the main planting pattern in this area.

The experiment adopted a completely randomized block design; the sowing date was 9 October 2020, the planting density was 1.95 million plants/hm^2, and there were 501 plots in total. Each plot was planted with six rows of new winter wheat varieties, repeated three times, and the plot area was 12 m^2. The management measures of the experimental field are higher than those of the ordinary field.

Data collection

Global wheat open dataset

The wheat spike image data is a public dataset provided by the Global Wheat Challenge 2021 International Conference on Computer Vision 2021.[1] The dataset consists of sample_submission.csv, test.zip, and train.zip, which each contain 3,655 images; the resolution of each image is 1024 × 1024.

Image data collection

The images were collected at 10:00 a.m. on 19 and 20 April 2021. The weather was clear and cloudless. The smartphone Huawei Honor 20 Pro was used to obtain the wheat heading stage images. The photographer fixed the smartphone on the handheld shooting pole, which shot vertically 50 cm above the wheat canopy. A total of 560 images were taken, and each image has a resolution of 960 × 720. An example of some images at the heading stage of wheat is shown in **Figure 1**.

Dataset construction and labeling

According to the number of images, the wheat heading date image is used as the dataset to construct the wheat spike number YOLOv5s detection model. The training dataset used in this paper is from train.zip provided by global wheat challenge 2021, where train.zip contains 3,655 images of wheat spikes and anchor box files. According to the number of wheat spikes in

[1] https://www.aicrowd.com/challenges/global-wheat-challenge-2021/dataset_files, 6 July 2021, day download.

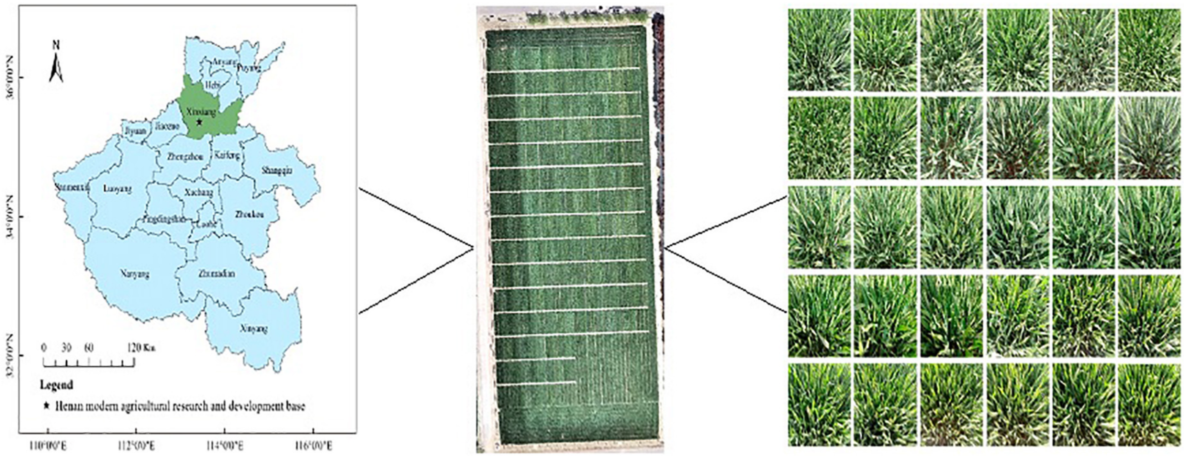


FIGURE 1
Geographical location of the study area.

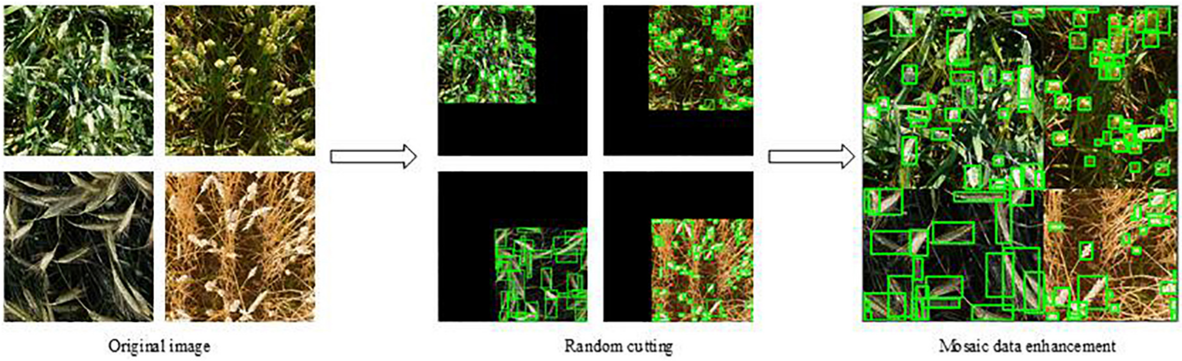


FIGURE 2
Data enhancement.

each image, 500 clear and unobstructed original images of the wheat heading stage were selected as the test set. According to the format requirements of the Pascal VOC dataset, labeling is used to label and generate the dataset XML type annotation file. Cut the original collected image into 640 × 640-pixel images.

Data enhancement

In order to improve the generalization ability of the training model, we mainly chose mosaic data enhancement, adaptive anchor box calculation, and adaptive image scaling as data enhancement methods. The details are as follows:

Mosaic data enhancement

Mosaic data enhancement uses four images and stitches them together in the form of random scaling, random clipping, and random arrangement. Each image has its own corresponding annotation box. After stitching the four images, a new image is obtained, and the corresponding annotation box of the image is also obtained. Then the image is transferred to a neural network for learning, which is equivalent to transferring four images for learning, making the model recognize the target in a smaller range. Figure 2 shows the workflow of wheat spikes enhanced with mosaic data.

Adaptive anchor box calculation

YOLOv5 network model does not only use the anchor box that has been labeled. Before starting training, it will check the labeled information in the dataset and calculate the best recall rate of the labeled information in this dataset for the default anchor box. When the best recall rate is greater than or equal to 0.98, there is no need to update the anchor box; If the optimal recall rate is less than 0.98, the anchor box that conforms to this data set needs to be recalculated. This function is embedded in the code in YOLOv5. For each training, the best anchor box is adaptively calculated according to the name of the data set. Users can turn off or turn on the image preprocessing function according to their own needs. This paper uses this image preprocessing method before training data.

Adaptive image scaling

Due to the different aspect ratios of most images, the size of black edges at both ends is different after using the traditional image scaling method to scale and fill. However, if too much filling is used, there will be a lot of information redundancy, affecting the algorithm's reasoning speed. In order to further improve the reasoning speed of YOLOv5, this method can adaptively add the fewest black edges to the scaled image.

Field measurement data collection

Consistent with the acquisition time of the image data, the measured value of the number of the wheat spikes was collected by an image-based manual counting method. Based on the unified wheat spike counting standard, people with relevant agronomic backgrounds were selected to count, and the average value was taken as the measured value of the wheat spike number corresponding to the image.

Network model construction

YOLOv5s network model

YOLOv5 is the latest product of the YOLO series, which is improved based on YOLOv4, and the running speed has been greatly improved (Chen and Chen, 2022). The YOLOv5 network model structure is mainly divided into four versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. In practical applications, a model of an appropriate size can be selected according to different specific scenarios. YOLOv5 is an improved version based on YOLOv4, which is a one-stage detection network with excellent accuracy and detection speed. After absorbing the advantages of the previous version and other networks, YOLOv5 has changed the previous YOLO target detection algorithm's characteristics of fast detection speed but low accuracy. YOLOv5 has improved the detection accuracy and real-time performance, meeting the real-time detection needs of video images, and the structure is also more compact. YOLOv5s have the least number of parameters, but the accuracy is low. YOLOv5s have a small depth and width while ensuring high accuracy. The other three versions continue to deepen and widen on this basis, especially when enhancing the extraction of image semantic information. YOLOv5s have the characteristics of fast running speed and high flexibility and have strong advantages in the rapid deployment of models. The network structure is shown in Figure 3. The network consists of four parts: input, backbone, neck, and head. The size of the input image at the input end is 640 × 640 × 3, and the images are preprocessed using strategies such as mosaic data enhancement, adaptive anchor box calculation, and image scaling. The role of the backbone network is to extract rich semantic features from the input image. It includes the Focus module, the Conv module, the C3 module, and the SPP module. In YOLOv5, CSPDarknet53 is used as the backbone network of the model. The neck adopts FPN and PAN to generate feature pyramids, which are used to enhance the detection of multi-scale objects.

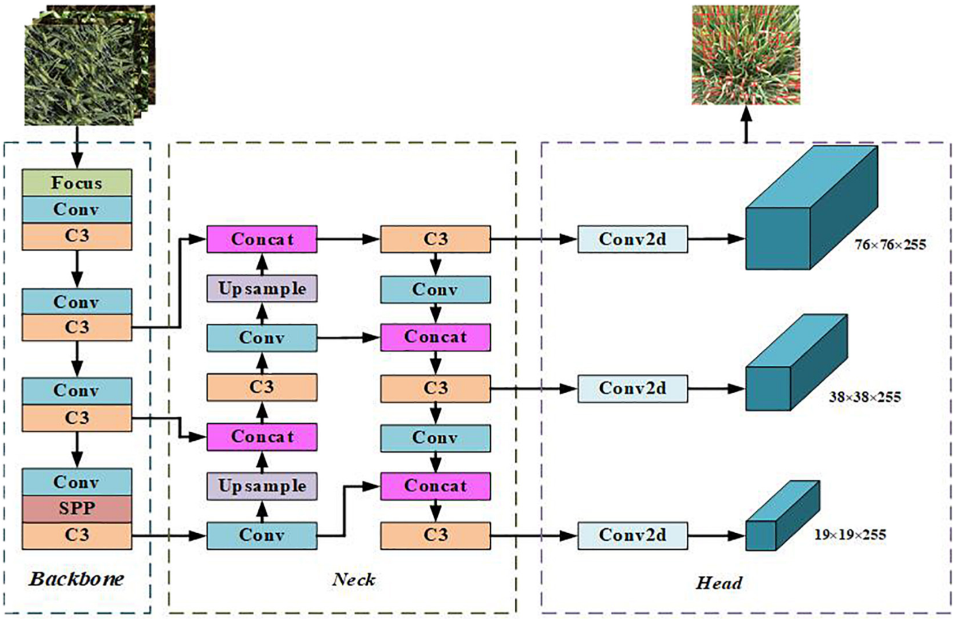


FIGURE 3
Network structure of YOLOv5s algorithm.

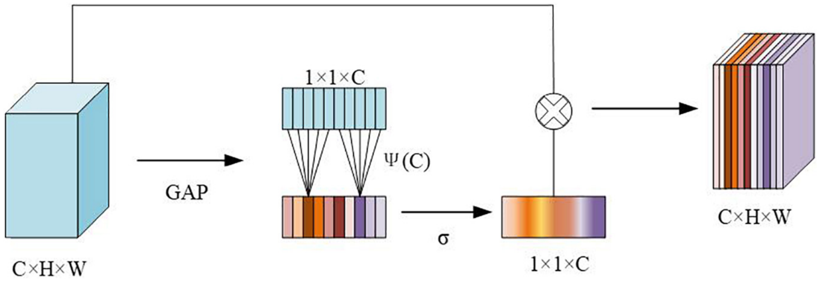


FIGURE 4
Structure of efficient channel attention (ECA) module.

The head is predicted from the features passed from the neck, and three different scaled feature maps are generated.

Improved YOLOv5s network model

Among the five models of the YOLOv5 network, the YOLOv5s model has high accuracy, fewer parameters, and fast detection speed, which can be deployed on hardware devices. The research on wheat spike detection and counting is based on the YOLOV5s network model, and the attention mechanism is added to YOLOV5s to improve the robustness of the network model.

Attention mechanism

The introduction of an attention mechanism into convolutional neural networks shows great potential for improving network performance. In the field of computer vision, attention mechanisms are widely used in natural scene segmentation, medical image segmentation, and object detection. Among them, the most representative is the Squeeze-and-Excitation (SE) (Hu et al., 2018), followed by the Convolutional Block Attention Module (CBAM) (Woo et al., 2018) module. Although the SE module can improve the network performance, it will increase the complexity and computational complexity of the model. The CBAM module ignores the channel-space interaction, which leads to the loss of cross-dimensional information. Therefore, this paper selects a more lightweight Efficient Channel Attention (ECA) (Wang et al., 2020) module and a Global Attention Mechanism (GAM) (Liu et al., 2021) that can amplify cross-dimensional interactions. In view of a large number of wheat spikes, dense distribution, occlusion, and overlap in the wheat spike image, the direct use of pre-trained YOLOv5x has high prediction accuracy, but the inference speed of the network is slow, and the number of parameters of the model is 168 M, which is difficult to use in hardware devices Deploy on. The reasoning speed of the YOLOv5s network model is fast, and the number of parameters is small, but the accuracy of YOLOv5s is low. The direct use of the YOLOv5s network model to detect and count wheat spikes is not satisfactory.

Introduce the improved C3 module of the efficient channel attention module

The ECA module structure is shown in Figure 4. The size of the input feature map is $C \times H \times W$, and then the size of the feature map is obtained through Global Average Pooling (GAP). The aggregated features obtained after GAP generate channel weights through a weight-sharing one-dimensional convolution. Among them, the one-dimensional convolution involves the hyperparameter ψ(C), which is the size of the convolution kernel determined by the mapping of the channel dimension C. Then, after the obtained feature map is operated, the output size is $1 \times 1 \times C$, and it is multiplied by the corresponding channel of the original input feature, and the final output feature size is $C \times H \times W$. Among them, the calculation method is shown in the following formula 1:

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \quad (1)$$

C represents the channel dimension, $|t|_{odd}$ represents the nearest odd number closest to it t, γ is set to 2, and b is set to 1.

In this study, the ECA module was introduced into the C3 module of the backbone part of the YOLOv5s network model so as to improve useful features, suppress unimportant features, and improve the accuracy of network model detection without additional model parameters. The improved C3 module is named the ECA-C3 module, and its structure is shown in Figure 5.

Introduce the YOLOV5s model improved by the global attention mechanism module

The purpose of the GAM module is to design an attention mechanism that can reduce information dispersion while amplifying the interactive features of the global dimension. Figure 6 shows the whole process of the GAM module. Given an input feature map, the intermediate states and outputs are defined as follows:

$$F_2 = M_c(F_1) \otimes F_1 \quad (2)$$

$$F_3 = M_s(F_2) \otimes F_2 \quad (3)$$

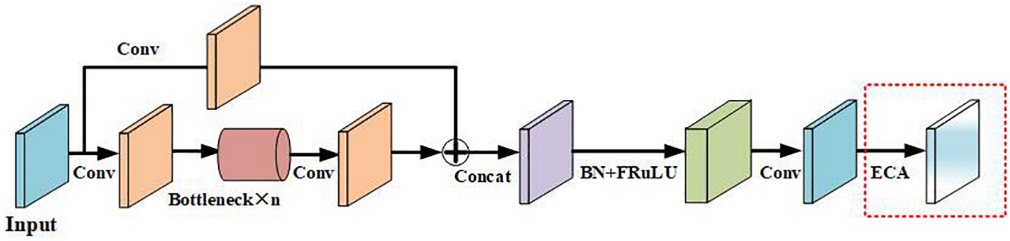


FIGURE 5
Structure of improved C3 module.

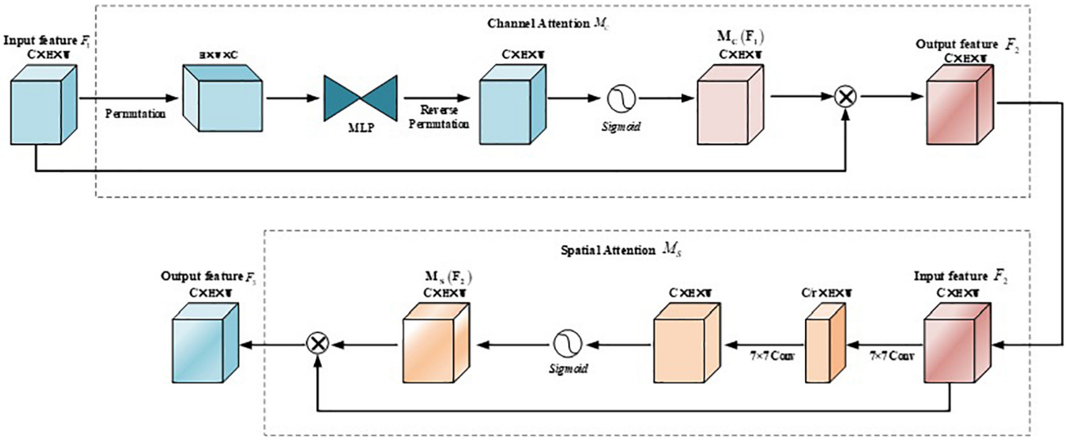


FIGURE 6
Structure of global attention mechanism (GAM) module.

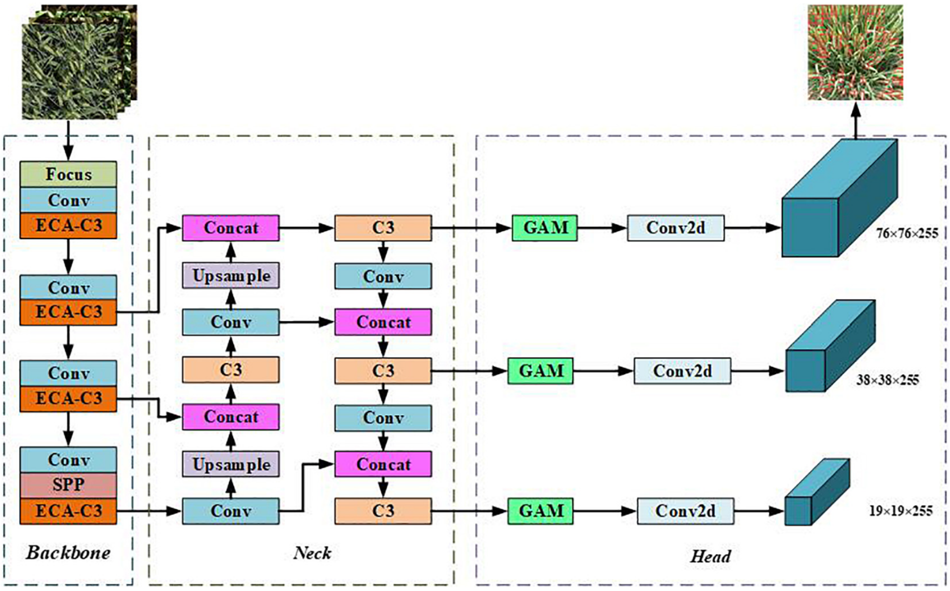


FIGURE 7
Network structure of improved YOLOv5s algorithm.

Among them, F_1 is the input feature map, F_2 is the feature map obtained after channel attention, F_3 is the final feature map after GAM M_c and M_S represents the channel attention map and the spatial attention map, respectively; $\otimes$ it represents element-wise multiplication.

The channel attention submodule maintains features in three dimensions using a three-dimensional arrangement and then amplifies the spatial dependencies across dimensions in a two-layer Multi-Layer Perceptron (MLP). In the spatial attention sub-module, first, two convolution operations with a kernel size of 7 × 7 are used for spatial information fusion. At the same time, in order to eliminate the feature loss caused by pooling, the pooling operation is removed here to maintain the feature map further.

YOLOv5s network model with attention mechanism

The improved YOLOv5s network model is shown in Figure 7. When different from the standard YOLOv5s, the improved model replaces the C3 module of the backbone part with the proposed ECA-C3 module so that the network can effectively extract the target features; GAM is added before the 2D convolution between the neck and head module, and the added GAM will increase the number of parameters of the network model, but it can make the network capture important features like the three-dimensional channel, space width, and space height. The size of the improved YOLOV5s input image is 3 × 640 × 640, and the first prediction branch of the head is used as an example to illustrate. The algorithm structure of the improved YOLOv5s model is shown in Table 1. Among them, “from” refers to the input layer corresponding to the layer module, and −1 refers to the previous layer.

Channel attention modeling

First, a feature map with a size of 256 × 80 × 80 is obtained through the C3 module, and a feature map of 80 × 80 × 256 is obtained through dimension transformation; the feature map is passed through a two-layer MLP, and the channel scaling rate is set to 4. The dimension of the feature map is reduced to 80 × 80 × 64, and then the dimension is increased to 80 × 80 × 256; the feature map is restored to the original shape and size of 256 × 80 × 80 through dimension transformation; the *sigmoid* function is used to obtain the size of 256 × 80 × 80 channel attention map; multiplies the original input feature map F and $M_C(F_1)$ to get a feature map of size 256 × 80 × 80.

Spatial attention modeling

First, F_1 pass a 7 × 7 convolution, and set the same channel scaling rate as the channel attention, and the size of the obtained feature map is 64 × 80 × 80; then go through a 7 × 7 convolution again to restore the feature map to 256 × 80 × 80. After using the *sigmoid* function, a spatial attention map $M_S(F_2)$ with a size of 256 × 80 × 80 is obtained; multiply with F_1 and $M_S(F_2)$, an output feature map with a size of 256 × 80 × 80 is obtained.

TABLE 1 Algorithm structure of improved YOLOv5s.

Number of layers	From	Parameter quantity	Module name
0	−1	3520	Focus
1	−1	18560	Conv
2	−1	18819	ECA-C3
3	−1	73984	Conv
4	−1	115715	ECA-C3
5	−1	295424	Conv
6	−1	625155	ECA-C3
7	−1	1180672	Conv
8	−1	656896	SPP
9	−1	1182723	ECA-C3
10	−1	131584	Conv
11	−1	0	Upsample
12	[−1,6]	0	Concat
13	−1	361984	C3
14	−1	33024	Conv
15	−1	0	Upsample
16	[−1,4]	0	Concat
17	−1	90880	C3
18	−1	147712	Conv
19	[−1,14]	0	Concat
20	−1	296448	C3
21	−1	590336	Conv
22	[−1,10]	0	Concat
23	−1	1182720	C3
24	[17,20,23]	8622262	Detect

Experimental results and analysis

Experimental equipment and parameter settings

The experiment is based on the deep learning framework built by Pytorch1.10 and CUDA11.2, using Linux Ubuntu18.04 LTS operating system, Intel$^{\circledR}$ Core $^{\mathrm{TM}}$i7-8700 CPU @3.70GHZ processor, Tesla T4 16G for experiments. The size of the images for training, verification, and testing in this experiment is 640 × 640 pixels, the input batch size is set to 8, and the training process is set to 60 epochs. The training process uses the stochastic gradient descent (SGD) optimizer; the initial learning rate is 0.01, the momentum factor is 0.937, and the weight decay rate is 0.0005.

Evaluation index and loss function

YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and improved YOLOv5s are validated on the validation set randomly divided into the public data set Global wheat challenge 2021, and the evaluation indicators Precision, Recall, mAP@0.5, and

TABLE 2 Test performance comparison of different models.

Methods	RMSE	MAE	Recall	mAP@.0.5	Map@.0.5:0.95
YOLOv5s	53.23	41.24	0.887	0.949	0.526
YOLOv5m	51.56	40.83	0.894	0.949	0.522
YOLOv5l	49.71	38.87	0.888	0.947	0.525
YOLOv5x	44.51	33.62	0.913	0.950	0.541
Improved YOLOv5s	43.94	34.36	0.911	0.951	0.545
Faster R-CNN	94.57	87.10	0.819	0.862	0.355

mAP@0.5:0.95 are similar, it showed that all three models could achieve the best performance in the detection task of the Global wheat challenge 2021, so the above four evaluation indicators are not selected to evaluate the model. This study mainly evaluates the performance of the model when the wheat spike data collected in the field is used as the test set for wheat spike counting. Therefore, the accuracy (Accuracy, ACC) is selected as the evaluation index for YOLOv5s counting, using the number of parameters and the amount of calculation (GFLOPs) and inference speed to evaluate model performance. The calculation formula of accuracy is as follows:

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

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$\mathrm{mAp} = \int_0^1 P \cdot R \, \mathrm{d}R \tag{6}$$

Among them, *TP* they represent true positives, *TN* represents true negatives, *FP* represents false positives, and *FN* represents false negatives. The larger the ACC value, the better the detection effect of the model.

In this study, CIoU is selected as the loss function to calculate the localization loss. CIoU can better represent the gap between the prediction and annotation frames, making the network model more robust during training. The CIoU loss function is defined as follows:

$$\mathrm{IoU} = \frac{area(ar \cap tr)}{area(ar \cup tr)} \tag{7}$$

$$\mathrm{CIoU} = 1 - \mathrm{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{8}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{9}$$

$$\mathrm{v} = \frac{4}{\pi^2}\left(\arctan\frac{\mathrm{w_{gt}}}{\mathrm{h_{gt}}} - \arctan\frac{\mathrm{w}}{\mathrm{h}}\right)^2 \tag{10}$$

Among them, ar and tr represent the anchor box and the bounding box $\rho^2(b, b^{gt})$ and the Euclidean distance between the center points of the anchor box and the bounding box, respectively. α is an equilibrium parameter and does not participate in gradient calculation; *v* is a parameter used to measure the consistency of aspect ratio. $^{\mathrm{W_{gt}}}$ and $^{\mathrm{h_{gt}}}$ are the width and the height of the bounding box, while w and h are the widths and the height of the anchor box.

TABLE 3 Statistical average error and average accuracy.

Methods	Mean error (%)	Mean accuracy (%)
YOLOv5s	33.34%	66.66%
YOLOv5m	33.29%	67.29%
YOLOv5l	30.89%	69.11%
YOLOv5x	27.52%	72.48%
Improved YOLOv5s	28.39%	71.61%
Faster R-CNN	54.07%	45.93%

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - q_i)^2} \tag{11}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|p_i - q_i| \tag{12}$$

where *N* is the number of images, p_i is the angle of the oriented detection box in the *i* image, and q_i is the angle of the corresponding oriented bounding box.

Quantitative analysis of experimental results

YOLOv5s, YOLOv5m, YOLOv5I, YOLOv5x, improved YOLOv5s, and the Faster R-CNN were used to evaluate the performance metrics of wheat spike data collected in the field. It can be seen from Table 2 that the evaluation metrics of Faster R-CNN were the worst. The evaluation metrics of improved YOLOv5s were better than those of standard YOLOv5s, YOLOv5m, and YOLOv5I and were similar to those of YOLOv5x.

The evaluation metrics of the average error rate and AP rate of the above different models on the test images are shown in Table 3. YOLOv5x has the highest AP, and Faster R-CNN has the lowest AP. Compared with the standard YOLOv5s, the accuracy of the improved YOLOv5s is improved by 4.95%, and compared with YOLOv5m and YOLOv5l, the AP is improved by 4.32 and 2.50%, respectively, and the AP is basically close to that of YOLOv5x.

Table 4 shows the comparison of different models in parameter quantity, giga floating-point operations per second (GFLOPs), inference, inference speed, and graphic processing unit (GPU) resource occupancy. Although the standard YOLOv5s parameter quantity, GFLOPs, inference, inference speed, and GPU resource occupancy are the least, the detection accuracy is low. While Faster R-CNN has the most GFLOPs, inference, inference speed, and GPU resource occupancy, the effect is the worst. The parameter quantity, GFLOPs, inference, inference speed, and GPU resource occupancy of the improved YOLOv5s are all larger than those of the standard YOLOv5s and less than those of the standard YOLOv5I and YOLOv5x.

Table 5 compares the AP and training time between EloU and CloU. By comparing the effects of EloU and CloU in the YOLOv5s model, the AP after using EloU is slightly higher than that of CloU, but the training time is significantly increased. Therefore, this paper selects CloU as the loss function to calculate the localization loss.

Qualitative analysis of experimental results

Figure 8 compares the recognition results of the standard YOLOv5s and YOLOv5m network models with the improved YOLOv5s network model in this paper for the recognition of wheat spikes in the field environment. It can be seen from Figure 8 that the standard YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x network models have seriously missed detections in areas with dense wheat spikes. With a high recognition rate and good generalization performance, the purple box area shows the superiority of the improved YOLOv5s detection results.

The images of wheat spikes are dense and sparse. Figure 9 shows the experimental results of the improved YOLOv5s model under different densities and backgrounds. Figures 9A,F show the counting results when the spikes of wheat are sparse; Figures 9B–D show the counting results in the case of dense wheat spikes. Among them, the color of wheat leaves in Figures 9B,D is similar to that of wheat spikes, and the color of wheat leaves in Figures 9C,E is yellow, and the color of wheat spikes is green.

Discussion

Spike number is an important indicator for determining wheat yield phenotypic traits, and spike detection is a hot spot in wheat phenotype research (Fernandez-Gallego et al., 2019). The wheat spike image data comes from the heading stage of this study. At this time, due to the large difference in the shape and the high density of the wheat spike, there are too many occluded parts, and the characteristics of the wheat spike are not obvious. In the process of spike recognition, there is a problem of omission in the detection of wheat spike occlusion, which leads to an error in the wheat spike count. In the wheat spike detection, the overlapping wheat spike in some images is not identified and marked, the adjacent wheat spike is not identified and marked, and the two wheat spikes are closely connected and identified as one wheat spike. This study proposes a target detection based on improved YOLOv5s, which corrects these problems in the process of wheat spike recognition. It effectively solves the problem of missed detection caused by occlusion and overlap in wheat spike detection. Therefore, the target detection method based on the improved YOLOv5s significantly improves the accuracy and recognition ability of the wheat spike in the image.

Deep learning is currently the main technical means of wheat spike recognition, detection, and counting. Using digital images of winter wheat to obtain the color, texture, and

TABLE 4 Comparison of parameter quantity, GFLOPs, inference, inference speed, and GPU resource occupancy of different models.

Methods	Parameter quantity (M)	GFLOPs	Inference (Min)	Inference speed (ms)	GPU resource occupancy (G)
YOLOv5s	13.38	15.8	370.5	7.5	1.70
YOLOv5m	39.77	47.9	396.2	11.6	1.80
YOLOv5l	87.90	107.6	415.6	17.3	2.10
YOLOv5x	164.36	204.0	479.9	29.0	2.40
Improved YOLOv5s	28.81	31.6	372.5	14.7	2.42
Faster R-CNN	41.30	278.2	755.3	227.7	7.87

TABLE 5 Comparison of average accuracy and training time between CloU and EloU of YOLOv5 models.

Methods	Mean accuracy (%)	Inference (Min)
Improved YOLOv5s with CIoU	71.61%	372.5
Improved YOLOv5s with EIoU	72.82%	405.6

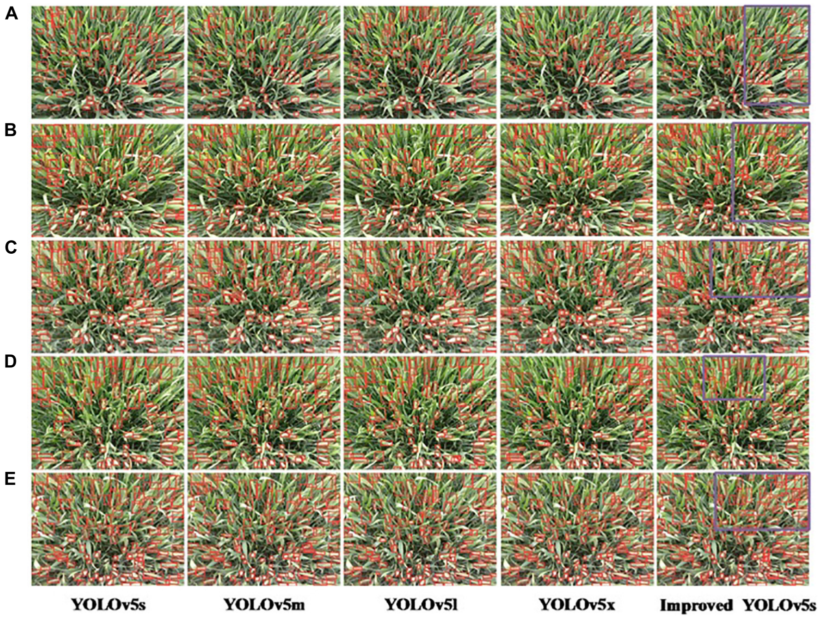


FIGURE 8
Qualitative analysis of experimental results of YOLOv5 algorithm. **(A–E)** Represent the number of images.

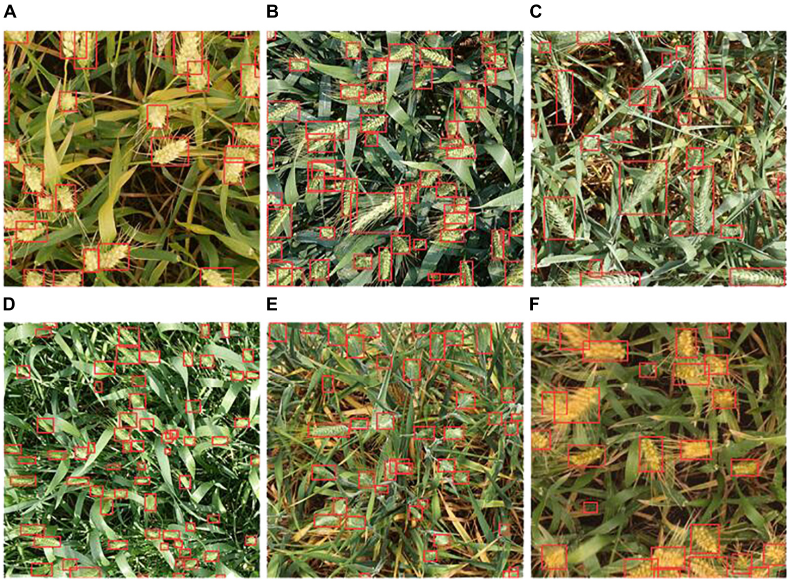


FIGURE 9
Experimental effects of improved YOLOv5s under different densities and backgrounds. **(A–F)** Represent six different images randomly selected from the global wheat challenge 2021 International Conference on computer vision 2021 dataset.

shape features of a wheat spike and establishing a wheat spike recognition classifier through deep learning methods, we identified wheat spike recognition and detection and counting. Zhou et al. (2018b) proposed an SVM segmentation method for segmenting wheat spikes in visible light images. Sadeghi-Tehran et al. (2019) developed the wheat spike number counting system DeepCount, which is used to automatically identify and count the number of wheat spikes in the images of wheat spikes. Alkhudaydi et al. (2019b) and Misra et al. (2020) constructed the SpikeletFCN spikelet counting model based on a fully convolutional network, which used the density estimation method to calculate the number of wheat spikelets.

These research results show that the deep convolutional neural network has good robustness for wheat spike counting. In this study, when the resolution of the input image is higher, the detection accuracy is also higher, which is consistent with other research results tested on general datasets (Singh et al., 2018). This study introduces ECA in the C3 module of the backbone structure of the YOLOv5s network model. The GAM module is inserted between the neck structure and the head structure. The accuracy and efficiency of the improved YOLOv5s target detection method are significantly improved, which solves the problem of wheat spikes caused by cross occlusion to a certain extent. The problem of unclear and omitted spike identification has better practical application value.

Conclusion

We developed an improved YOLOv5s-based attention mechanism for wheat spike number image detection. The method includes three key steps: data preprocessing of the wheat spike image, adding an attention mechanism module for network improvement, and YOLOv5s network model fused with an attention mechanism. In the wheat spike counting task, the accuracy of the improved YOLOv5s model reached 71.61%, which was 4.95% higher than that of the standard YOLOv5s model and had higher counting accuracy. The improved YOLOv5s and YOLOv5m have similar parameters, while RMSE and MEA are reduced by 7.62 and 6.47, respectively, and the performance is better than YOLOv5l. The experimental results show that the improved YOLOv5s algorithm improves the applicability in complex field environments, which can accurately detect the number of small-scale wheat spikes and better solve the occlusion and overlapping problems of a wheat spike.

In the case of extremely dense samples, the coincidence probability of wheat spike heads is high, and the regression idea of the YOLO algorithm is based on dividing the image into grids; that is, each grid can only predict one target at most, so it does not perform well when there are multiple target objects in the same grid, and it is impossible to identify all the targets. Due to its portability and lightweight network, YOLOv5s is used as the main model for training, which improves its flexibility and speed compared with YOLOv4, and reduces many of its parameters to make it applicable to portable devices. The improved model needs to take into account the training accuracy and training speed and increase the number of parameters.

The improved YOLOv5s method proposed in this study can realize the counting of wheat spikes and can meet the needs of high-throughput operations in the wheat field environment. In future research work, we will gradually optimize the built-in YOLOv5s network structure and analyze the wheat spike detection network structure for the wheat spike images acquired by smartphones to obtain better wheat detection performance. In addition, we will envisage applying this method to other crop counts to demonstrate its robustness in solving occlusion and overlap problems. Subsequently, the improved YOLOv5s method can save time and effort.

Data availability statement

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

Author contributions

HZ and LR: conceptualization, software, formal analysis, and visualization. HZ: methodology, writing – original draft preparation, and supervision. HZ, MZ, DC, and QZ: validation. HZ and YW: investigation. HZ, GZ, and GL: resources. HZ, YW, and LR: data curation. YW and GL: writing – review and editing. HZ and GZ: project administration and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding

This study has been funded by key science and technology projects in Henan Province (Contract Number: 212102110253), the Henan Academy of Agricultural Sciences Independent Innovation Project (Contract Number: 2022ZC51), and the science and technology innovation leading talent cultivation program of the Institute of Agricultural Economics and Information, Henan Academy of Agricultural Sciences (Contract Number: 2022KJCX02).

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

Alkhudaydi, T., Reynolds, D., Griffiths, S., Zhou, J., and Iglesia, B. D. L. (2019a). An exploration of deep-learning based phenotypic analysis to detect spike regions in field conditions for UK bread wheat. *Plant Phenomics* 2019, 7368761. doi: 10.34133/2019/7368761

Alkhudaydi, T., Zhou, J., and La lglesia, B. D. (2019b). "SpikeletFCN: Counting spikelets from infield wheat crop images using fully convolutional networks," in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICASC)*, (Cham: Springer Nature Switzerland AG), 3–13. doi: 10.1007/978-3-030-20912-4_1

Bai, D. X., Sun, Y., Tao, B., Tong, X. L., Xu, M. M., Jiang, G. Z., et al. (2022). Improved single shot multibox detector target detection method based on deep feature fusion. *Concurr. Comput.* 34, e6614. doi: 10.1002/cpe.6614

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

Chen, C., Frank, K., Wang, T., and Wu, F. (2021). Global wheat trade and codex alimentarius guidelines for deoxynivalenol: A mycotoxin common in wheat. *Glob. Food Secur.* 29:100538. doi: 10.1016/j.gfs.2021.100538

Chen, S. P., and Chen, B. C. (2022). "Research on object detection algorithm based on improved Yolov5," in *Artificial Intelligence in China*, eds Q. Liang, W. Wang, J. Mu, X. Liu, and Z. Na (Singapore: Springer), 290–297. doi: 10.1007/978-981-16-9423-3_37

David, E., Madec, S., Sadeghi-tehran, P., Aasen, H., Zheng, B. Y., Liu, S. Y., et al. (2020). Global wheat head detection (GWHD) dataset: a large and diverse dataset of high-resolution RGB-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics* 2020:3521852. doi: 10.34133/2020/3521852

David, E., Serouart, M., Smith, D., Madec, S., Velumani, K., Liu, S. Y., et al. (2021). Global wheat head detection 2021: An improved dataset for benchmarking wheat head detection methods. *Plant Phenomics* 2021:9. doi: 10.34133/2021/9846158

FAO (2021). *Impact of disasters and crises on agriculture and food security.* Available online at: https://www.fao.org/director-general/speeches/detail/zh/c/1382466/ (accessed March 18).

Fernandez-Gallego, J. A., Kefauver, S. C., Gutiérrez, N. A., Nieto-Taladriz, M. T., and Araus, J. L. (2018). Wheat spike counting in-field conditions: high throughput and low-cost approach using RGB images. *Plant Methods* 14:22. doi: 10.1186/s13007-018-0289-4

Fernandez-Gallego, J., Buchaillot, M. L., Aparicio Gutiérrez, N., Nieto-Taladriz, M. T., Araus, J. L., and Kefauver, S. C. (2019). Automatic wheat spike counting using thermal imagery. *Remote Sens.* 11, 751–764. doi: 10.3390/rs11070751

Ganeva, D., Roumenina, E., Dimitrov, P., Gikov, A., Jelev, G., Dragov, R., et al. (2022). Phenotypic traits estimation and preliminary yield assessment in different phenophases of wheat breeding experiment based on UAV multispectral images. *Remote Sens.* 14:1019. doi: 10.3390/rs14041019

Gao, Q., Liu, J. G., Member, S., Ju, Z. J., and Zhang, X. (2019). Dual-hand detection for human-robot interaction by a parallel network based on hand detection and body pose estimation. *IEEE Trans. Ind. Electron.* 66, 9663–9672. doi: 10.1109/TIE.2019.2898624

Ge, D. Z., Long, H. L., Zhang, Y. N., Ma, L., and Li, T. T. (2018). Farmland transition and its influences on grain production in China. *Land Use Policy* 70, 94–105. doi: 10.1016/j.landusepol.2017.10.010

Gong, B., Ergu, D., Cai, Y., and Ma, B. (2021). Real-time detection for wheat head applying deep neural network. *Sensors* 21:191. doi: 10.3390/s21010191

Gou, F., Van Ittersum, M. K., Wang, G., Van der Putten, P. E., and Van der Werf, W. (2016). Yield and yield components of wheat and maize in wheat–maize intercropping in the Netherlands. *Eur. J. Agron* 76, 17–27. doi: 10.1016/j.eja.2016.01.005

Grillo, O., Blangiforti, S., and Venora, G. (2017). Wheat landraces identification through glumes image analysis. *Comput. Electron. Agric.* 141, 223–231. doi: 10.1016/j.compag.2017.07.024

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, 2–13. doi: 10.1186/s13007-018-0366-8

He, M. X., Hao, P., and Xin, Y. Z. (2020). A robust method for wheat spike detection using UAV in natural scenes. *IEEE Access* 8, 189043–189053. doi: 10.1109/ACCESS.2020.3031896

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Salt Lake City, UT, 7132–7141. doi: 10.1109/CVPR.2018.00745

Huang, L., Chen, C., Yun, J. T., Sun, Y., Tian, J. R., Hao, Z. Q., et al. (2022). Multi-scale feature fusion convolutional neural network for indoor small target detection. *Front. Neurorobitics* 16:881021. doi: 10.3389/fnbot.2022.881021

Huang, L., Fu, Q. B., He, M. L., Jiang, D., and Hao, Z. Q. (2021). Detection algorithm of safety helmet wearing based on deep learning. *Concurr. Comput.* 33:e6234. doi: 10.1002/cpe.6234

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

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

Laborde, D., Martin, W., Swinnen, J., and Vos, R. (2020). COVID-19 risks to global food security-Economic fallout and food supply chain disruptions require attention from policy-makers. *Science* 369, 500–502. doi: 10.1126/science.abc4765

Li, J. B., Li, C. C., Fei, S. P., Ma, C. Y., Chen, W. N., Ding, F., et al. (2021). Wheat spike recognition based on RetinaNet and transfer learning. *Sensors* 21:4845. doi: 10.3390/s21144845

Li, L., Hassan, M. A., Yang, S. R., Jing, F. R., Yang, M. J., Rasheed, A., et al. (2022). Development of image-based wheat spike counter through a Faster R-CNN algorithm and application for genetic studies. *Crop J.* doi: 10.1016/j.cj.2022.07.007

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). "SSD: Single shot multibox detector," in *Proceedings of the European conference on computer vision (ECCV 2016)*, Amsterdam, 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, Y., Shao, Z., and Hoffmann, N. (2021). Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* [Preprint]

Lu, H., Liu, L., Li, Y. N., Zhao, X. M., Wang, X. Q., and Cao, Z. G. (2021). TasselNetV3: Explainable plant counting with guided upsampling and background suppression. *IEEE Trans. Geosci. Remote Sens.* 60:4700515. doi: 10.1109/TGRS.2021.3058962

Lu, J., Hu, J., Zhao, G. N., Mei, F., and Zhang, C. S. (2017). An in-field automatic wheat disease diagnosis system. *Comput. Electron. Agric.* 142, 369–379. doi: 10.1016/j.compag.2017.09.012

Madec, S., Jin, X., Lu, H., De Solan, B., Liu, S., Duyme, F., et al. (2019). Spike density estimation from high resolution RGB imagery using deep lspikening technique. *Agric. For. Meteorol.* 264, 225–234. doi: 10.1016/j.agrformet.2018.10.013

Ministry of Emergency Management of the People's Republic of China (2022). *Basic situation of national natural disasters in 2021*. Beijing: Ministry of Emergency Management of the People's Republic of China.

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 wheat plant from visual imaging. *Plant Methods* 16:40. doi: 10.1186/s13007-020-00582-9

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 *Proceedings of the IEEE International Conference on Computer Vision (ICCVW)*, Venice, 2055–2063. doi: 10.1109/ICCVW.2017.241

Redmon, J., and Farhadi, A. (2017). "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, HI, 7263–7271. doi: 10.1109/CVPR.2017.690

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 (CVPR 2016)*, Las Vegas, NV, 779–788. doi: 10.1109/CVPR.2016.91

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.* 10:1176. doi: 10.3389/fpls.2019.01176

Singh, B., Najibi, M., and Davis, L. S. (2018). SNIPER: Efficient multi-scale training. *arXiv* [Preprint]. Available online: https://arxiv.org/abs/1805.09300 (accessed on 23 May 2018).

Sreenivasulu, N., and Schnurbusch, T. (2012). A genetic playground for enhancing grain number in cereals. *Trends Plant Sci.* 17, 91–101. doi: 10.1016/j.tplants.2011.11.003

Sun, Y., Zhao, Z. C., Jiang, D., Tong, X. L., Tao, B., Jiang, G. Z., et al. (2022). Low-illumination image enhancement algorithm based on improved multi-scale Retinex and ABC algorithm optimization. *Front. Bioeng. Biotechnol.* 10:865820. doi: 10.3389/fbioe.2022.865820

Ultralytics, (2021). *YOLOv5*. Available online at: https://github.com/ultralytics/yolov5 (accessed April 4, 2022).

Wang, D., Zhang, D., Yang, G., Xu, B., Luo, Y., and Yang, X. (2021). SSRNet: In-field counting wheat spikes using multi-stage convolutional neural network. *IEEE Trans. Geosci. Remote Sen* 60, 1–11. doi: 10.1109/TGRS.2021.3093041

Wang, Q. L., Wu, B. G., Zhu, P. F., Li, P. H., Zuo, W. M., and Hu, Q. H. (2020). "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, WA, 11531–11539. doi: 10.1109/CVPR42600.2020.01155

Wang, X., Xuan, H., Evers, B., Shrestha, S., Robert, P., and Jesse, P. (2019). High-throughput phenotyping with deep learning gives insight into the genetic architecture of flowering time in wheat. *Gigascience* 8:giz120. doi: 10.1101/527911

Wang, Y. D., Qin, Y. X., and Cui, J. L. (2021). Occlusion robust wheat ear counting algorithm based on deep learning. *Front. Plant Sci.* 12:645899. doi: 10.3389/fpls.2021.645899

Wen, C. J., Wu, J. S., Chen, H. R., Su, H. Q., Chen, X., Li, Z. S., et al. (2022). Wheat spike detection and counting in the field based on SpikeRetinaNet. *Front. Plant Sci.* 13:821717. doi: 10.3389/fpls.2022.821717

Weng, Y. Q., Sun, Y., Jiang, D., Tao, B., Liu, Y., Yun, J. T., et al. (2021). Enhancement of real-time grasp detection by cascaded deep convolutional neural networks. *Concurr. Comput.* 33:e5976. doi: 10.1002/cpe.5976

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). "CBAM: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, (Cham: Springer), 3–19. doi: 10.1007/978-3-030-01234-2_1

Xiong, H., Cao, Z., Lu, H., Madec, S., Liu, L., and Shen, C. H. (2019). TasselNetv2: in-field counting of wheat spikes with context-augmented local regression networks. *Plant Methods* 15:150. doi: 10.1186/s13007-019-0537-2

Yang, B. H., Gao, Z. W., Gao, Y., and Zhu, Y. (2021). Rapid detection and counting of wheat Ears in the field using YOLOv4 with attention module. *Agronomy* 11:1202. doi: 10.3390/agronomy11061202

Zhang, H., Turner, N. C., Poole, M. L., and Asseng, S. (2007). High spike number is key to achieving high wheat yields in the high-rainfall zone of south-western Australia. *Aust. J. Agric. Res.* 58, 21–27. doi: 10.1071/AR05170

Zhao, J. Q., Yan, J. W., Xue, T. J., Wang, S. W., Qiu, X. L., Yao, X., et al. (2022). A deep learning method for oriented and small wheat spike detection (OSWSDet) in UAV images. *Comput. Electron. Agric.* 198:107087. doi: 10.1016/j.compag.2022.107087

Zhao, J. Q., Zhang, X. H., Yan, J. W., Qiu, X. L., Yao, X., Tian, Y. C., et al. (2021). A wheat spike detection method in UAV images based on improved YOLOv5. *Remote Sens.* 13:3095. doi: 10.3390/rs13163095

Zhou, C. Q., Liang, D., Yang, X., Xu, B., and Yang, G. (2018a). Recognition of wheat spike from field based phenotype platform using multi-sensor fusion and improved maximum entropy segmentation algorithms. *Remote Sens.* 10:246. doi: 10.3390/rs10020246

Zhou, C. Q., Liang, D., Yang, X. D., Yang, H., Yue, J. B., and Yang, G. J. (2018b). Wheat ears counting in field conditions based on multi-feature optimization and TWSVM. *Front. Plant Sci.* 9:1024. doi: 10.3389/fpls.2018.01024

Zhou, H., Riche, A. B., Hawkesford, M. J., Whalley, W. R., Atkinson, B. S., Sturrock, C. J., et al. (2021). Determination of wheat spike and spikelet architecture and grain traits using X-ray computed tomography imaging. *Plant Methods* 17:26. doi: 10.1186/s13007-021-00726-5

TYPE Original Research
PUBLISHED 01 November 2022
DOI 10.3389/fpls.2022.1002312

Dilated convolution capsule network for apple leaf disease identification

Cong Xu, Xuqi Wang and Shanwen Zhang*

School of Electronic Information, Xijing University, Xi'an, China

Check for updates

OPEN ACCESS

EDITED BY
Gregorio Egea,
University of Seville, Spain

REVIEWED BY
Aparajita Ojha,
PDPM Indian Institute of Information Technology, Design and Manufacturing, India
Bin Liu,
Northwest A&F University, China

*CORRESPONDENCE
Shanwen Zhang
wjdw716@163.com

SPECIALTY SECTION
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

RECEIVED 30 August 2022
ACCEPTED 05 October 2022
PUBLISHED 01 November 2022

CITATION
Xu C, Wang X and Zhang S (2022) Dilated convolution capsule network for apple leaf disease identification. *Front. Plant Sci.* 13:1002312. doi: 10.3389/fpls.2022.1002312

Accurate and rapid identification of apple leaf diseases is the basis for preventing and treating apple diseases. However, it is challenging to identify apple leaf diseases due to their various symptoms, different colors, irregular shapes, uneven sizes, and complex backgrounds. To reduce computational cost and improve training results, a dilated convolution capsule network (DCCapsNet) is constructed for apple leaf disease identification based on a capsule network (CapsNet) and two dilated Inception modules with different dilation rates. The network can obtain multi-scale deep-level features to improve the classification capability of the model. The dynamic routing algorithm is used between the front and back layers of CapsNet to make the model converge quickly. In DCCapsNet, dilated Inception instead of traditional convolution is used to increase the convolution receptive fields and extract multi-scale features from disease leaf images, and CapsNet is used to capture the classification features of changeable disease leaves and overcome the overfitting problem in the training network. Extensive experiment results on the apple disease leaf image dataset demonstrate that the proposed method can effectively identify apple diseases. The method can realize the rapid and accurate identification of apple leaf disease.

KEYWORDS

apple leaf disease identification, dilated convolution, capsule network (CapsNet), dilated convolution CapsNet (DCCapsNet), inception

Introduction

Apple is one of the most popular fruits. However, it is often affected by various diseases, which reduce its yield and quality (Pandiyan et al., 2020). Rapid and accurate detection and identification of these diseases is a prerequisite for disease control and accurate use of pesticides. Traditional methods of manual detection and identification of apple diseases mainly rely on visual recognition, which is not only subjective but also time-consuming, laborious, and inefficient and requires sufficient field experience and subjective assumptions. This method cannot be used for the quantitative identification of diseases; nor can it be widely used in large apple plantations. Apple leaves are susceptible to diseases. Because of the complex symptoms of apple leaf disease, detection and identification by apple disease leaf image is challenging research (Mishra et al., 2017; Puspha Annabel et al., 2019). Zhang et al. (2017) proposed an apple leaf disease recognition method based on image processing techniques and pattern recognition, including image lesion segmentation, feature extraction, dimension reduction, and disease identification. In the method, 38 classifying features of color, texture, and shape were from each segmented spot image, and the few most valuable features were selected by combining genetic algorithm (GA) and correlation feature selection algorithm. Finally, the diseases were recognized by a support vector machine (SVM) classifier. In fact, the similarity between the different-class disease spot images is small, while the similarity between the within-class disease spot images is largely due to the complex background environment, so the traditional apple leaf disease recognition using complex image pretreatment and feature extraction cannot guarantee a high disease recognition rate.

With the development of deep learning and big data processing technologies, convolutional neural networks (CNNs) realize end-to-end detection by learning multi-level features of different receptive fields, scenes, and scales (Lei et al., 2018; Li et al., 2019; Sun et al., 2021) and have become a topic of research in the crop automatic disease recognition fields (Sun et al., 2017). Sun et al. (2021) proposed a lightweight CNN model to detect apple leaf diseases in real time. They constructed a dataset of apple leaf disease image dataset, namely, AppleDisease 5, proposed a MEAN block, and built an apple leaf disease detection model by using the MEAN block and Apple-Inception module. Agarwal et al. (2019) developed a CNN model to identify apple disease. It consists of three convolution layers and three max-pooling layers followed by two densely connected layers. They tested the model with varying numbers of convolution layers from two to six and found that three layers have the best. Jiang et al. (2019) proposed an apple leaf disease real-time detection based on improved CNN. In the method, the apple leaf disease dataset was constructed *via* data augmentation and image annotation technologies, and an apple leaf disease detection method based on deep CNN (DCNN) was proposed by introducing the GoogLeNet Inception structure and Rainbow concatenation. The proposed model was trained using a dataset of 26,377 images of diseased apple leaves to detect these five common apple leaf diseases. Yan et al. (2020) proposed an improved VGG16 model, namely, VGG-ICNN, for apple leaf disease recognition. It consists of approximately 6 million parameters that are substantially fewer than most of the available high-performing deep learning models. Zhong et al. (Zhong and Zhao, 2020) proposed DenseNet-121 to identify apple leaf diseases and used an apple leaf image dataset including 2,462 images of six apple leaf diseases to train and evaluate the model. Some deep learning approaches have recently been introduced for leaf disease identification, such as VGG and residual network (ResNet). Son et al. (Yu and Son, 2020) proposed a deep learning architecture for apple disease recognition by considering the leaf spot attention mechanism. To realize this, they designed a feature segmentation subnetwork to provide more discriminative features and a spot-aware classification subnetwork for the feature segmentation subnet and then trained through early fusion and late fusion to generate semantic point feature information. The results proved that the proposed method outperforms conventional state-of-the-art deep learning models. Luo et al. (2021) proposed an apple disease classification model based on a multi-scale conventional ResNet. To solve the problem of serious loss of information in the ResNet downsample, the channel projection and spatial projection of downsample were separated, the 3 × 3 convention in ResBlocks was replaced by pyramid convolution, and the dilated convolution with different dilation rates was introduced into pyramid convolution to enhance the output scale of feature maps and improve the robustness of the model. The results on the dataset of this paper demonstrated that the optimal model has a high accuracy, which can provide a reference for the prevention and control of apple leaf diseases. Khana et al. (2022) proposed a real-time apple leaf disease detection system based on deep learning. The qualitative results validated that the proposed system can efficiently and accurately identify leaf disease symptoms and can be used as a practical tool by farmers and apple growers to aid them in the diagnosis, quantification, and follow-up of infections. Di et al. (Di and Li, 2022) proposed an apple disease detection approach based on improved CNN, namely, DF-Tiny-YOLO. Feature reuse is combined with DenseNet dense connection network to reduce the disappearance of depth gradient, so as to strengthen feature propagation and improve detection accuracy. The calculation parameters of DF-Tiny-YOLO are reduced by convolution kernel compression, and the operation detection speed is improved. Feature fusion is realized by feature superposition. The results showed that this method can improve detection performance significantly.

According to the above methods, the deeper the convolution layer is, the more abstract the extracted features are, and the higher the recognition rate is. However, the larger convolution kernel and the deeper CNN model have more training parameters, requiring longer training time and greater computational power.

Most of the existing apple detection models based on CNN are difficult to use on hardware resource platforms with limited computing capacity and storage capacity due to too many parameters. To improve the performance and adaptability of the existing apple detection model under the condition of limited hardware resources, while maintaining detection accuracy, reducing the calculation of the model and the model computing and storage footprint, and shortening detection time, Xia et al. (2020) proposed an apple detection model based on lightweight anchor-free deep CNN, namely, lightweight MobileNetV3. MobileNetV3 outperforms CenterNet and SSD (Single Shot Multibox Detector) in comprehensive performance, detection accuracy, capacity, and convergence speed. Li et al. (2022) proposed an apple identification method based on lightweight RegNet. To evaluate the effectiveness of this method, a series of comparative experiments were conducted using 2,141 images of five field apple leaf diseases and compared with the state-of-the-art improved CNN such as ShuffleNet, EfficientNet-B0, MobileNetV3, and Vision Transformer. The results show that the performance of RegNet-Adam is better than that of other pre-training models, and transfer learning can realize fast and accurate identification of apple leaf diseases.

In CNN, pooling is usually used to increase the receptive field and reduce the amount of calculation, but some useful information may be lost. Dilated convolution can increase the receptive field of the convolution kernel without increasing the number of parameters to improve the feature resolution, and the size of the output feature map can remain unchanged (Ahmed, 2021). Dilated convolution can be used to improve the quality of the training results and decrease the required computational costs. For example, a 3 × 3 convolution kernel with an expansion rate of 2 has the same receptive field as a 5 × 5 convolution kernel, while the number of parameters is only 9, which is 36% of the number of 5 × 5 convolution parameters. Therefore, dilated convolution can be used for constructing a lightweight CNN model (Fang et al., 2019). Thakur et al. (2022) introduced a lightweight CNN, namely, VGG-ICNN, for the identification of crop diseases using plant-leaf images. It consists of approximately 6 million parameters that are substantially fewer than most of the available high-performing deep learning models. Many models with large parameters have difficulty providing an accurate and fast diagnosis of apple leaf pests and diseases on mobile terminals. Zhu et al. (2022) proposed a lightweight model for early apple leaf pests and disease classification, where a LAD-Inception is built to enhance the ability to extract multi-scale features of different sizes of disease spots. Li et al. (2022) proposed a lightweight convolutional neural network RegNet to realize the rapid and accurate identification of apple leaf disease and conducted a series of comparative experiments based on 2,141 images of five apple leaf diseases (rust, scab, ring rot, panonychus ulmi, and healthy leaves) in the field environment.

CNN has a strong feature extraction ability, but it cannot acquire the relationship between feature attributes, such as relative position and size. Its high recognition rate on the complex image dataset depends on a large number of training samples, but the actual amount of data obtained is often limited, leading to the overfitting of CNN. Capsule Network (CapsNet) can make up for the deficiency of CNN. Capsule is a set of neurons that capture various parameters of a particular feature, each representing various properties of a particular entity that appears in an image. These attributes include many different types of instantiation parameters such as posture (position, size, and direction), deformation, speed, hue, and texture. One special property in the capsule is the presence of an instance of a category in the image. CapsNet transforms the scalar output of neurons into vector output, which is the probability of the entity's existence. It not only can represent whether the image has a certain feature but also can represent the physical features such as rotation and position of the feature (Wang et al., 2019). Xiang et al. (2018) designed a multi-scale CapsNet (MS-CapsNet), in which the multi-scale features are extracted by multi-scale convolutional kernels and then used to construct the multi-dimensional primary capsules. Deng et al. (2018) used the improved double-layer CapsNet to classify the PaviaU (PU) dataset of hyperspectral images and obtained a recognition rate of 93.45%. Yang et al. (2018) compared the classical CNN with CapsNet in terms of network structure, parameter update, and training results. Experimental results showed that CapsNet is better on gray images than the classical CNNs. CNN-based architectures have performed amazingly well for disease detection in plants but at the same time lack rotational or spatial invariance. CapsNet addresses these limitations of CNN architectures. Janakiramaiah et al. (2021) proposed a variant of CapsNet called Multilevel CapsNet to characterize the mango leaves tainted by anthracnose and powdery mildew diseases. It is validated on a dataset of mango leaves collected in the natural environment.

Inspired by dilated convolution, MS-CapsNet, and their improvement, a dilated convolution capsule network (DCCapsNet) is constructed for apple leaf disease identification. The main contributions are given as follows:

- Two dilated Inception modules are introduced into CapsNet to extract the multi-scale classifying features of disease leaf images, improve the classification capability of the model, and overcome the overfitting problem.
- DCCapsNet is constructed to recognize apple leaf diseases, where the dynamic routing algorithm is used between the front and back layers of CapsNet to make the model converge quickly.
- The effectiveness of this method is verified by many experiments.

The rest of this paper is organized as follows. Section 2 briefly introduces dilated convolution and CapsNet. DCCapsNet is introduced in detail in Section 3. The experiments and analysis are presented in Section 4. The summary and prospect of the paper are given in Section 5.

Related works

In this section, dilated convolution and CapsNet are briefly introduced.

Dilated convolution

Dilated convolution can enlarge the receptive field of the convolution layer by filling 0 in the middle of the convolution kernel, without increasing network parameters and then avoiding feature loss caused by pooling operation in CNN. Dilated convolution structures of three dilated rates are shown in Figures 1A–C, where (A) the receptive field is 3 × 3 with an expansion rate of 1 (that is, the traditional convolution kernel of 3 × 3); (B) the receptive field is enlarged to 5 × 5 with a dilated rate of 2 by filling with a 0 in the 3 × 3 standard convolution; (C) the receptive field is increased to 7 × 7 with a dilated rate of 3 by filling with two 0 in the 3 × 3 standard convolution. As can be seen from Figures 1A–C, with the increase of dilated rate, the size of the receptive field increases, but the network parameters do not increase, that is, nine parameters. Therefore, using the dilated convolutional instead of the traditional convolutional can extract more features without increasing the amount of computation.

Assume an apple rust leaf image and a 3 × 3 sharp kernel [−1 −1 −1;−1 9 −1;−1 −1 −1] and conduct several convolutions of the leaf image and dilated convolution kernels (r = 1, 2, 3, 5). The convolution maps are shown in Figures 1E–H. From the convolution maps in Figure 1, it can be seen that dilated convolution not only can expand the receptive field but also can extract more discriminant features than classical convolution and keep the relative spatial position of spot pixels unchanged without increasing computation and losing resolution. Comparing Figures 1G, H, there is not much difference between the two maps. Therefore, we utilized dilated convolution kernels (r = 1, 2, 3).

In DCNN, downsampling is usually used to increase the receptive field, but the image resolution will be reduced, resulting in the loss of spatial detail of the image. The dilated convolution expands the receptive field by setting the dilated rate, and setting different dilated rates can also capture multi-scale context information. It can be seen from Figure 1, on the basis of no additional parameters, that the receptive field of 3 × 3 convolution is expanded to 5 × 5 and 7 × 7, which can capture multi-scale features of the image. Therefore, multi-scale receptive fields can be obtained through the dilated convolution of different expansion rates. Dilated convolution can be considered a multi-scale convolution network. Dilated convolutional kernel and receptive field are calculated as follows

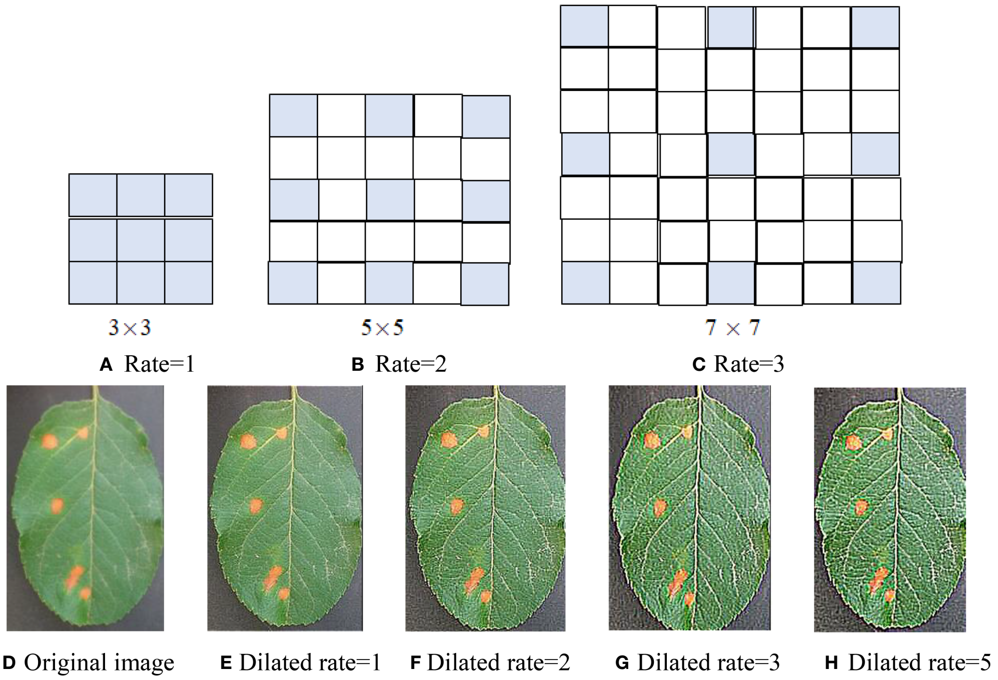


FIGURE 1
Dilated convolution with three dilation rates. **(A)** Rate = 1. **(B)** Rate = 2. **(C)** Rate = 3. **(D)** Original image. **(E)** Dilated rate = 1. **(F)** Dilated rate = 2. **(G)** Dilated rate = 3. **(H)** Dilated rate = 5.

$$n = k + (k-1)(r-1)$$
$$l_m = l_{m-1} + [(f_m - 1)\prod_{i=1}^{m-1} S_i] \qquad (1)$$

where k and n are the size of the original convolution kernel and dilated convolution kernel, respectively; l_{m-1} is the receptive field size of the $(m-1)$ layer; l_m is the receptive field size at the mth layer after the convolution of the void; f_m is the size of the convolution kernel at the mth layer; S_i is the step size of layer l.

Capsule network

CapsNet consists of one convolution layer and a primary capsule layer and a digital capsule layer. In its internal structure, the capsule layer is taken as the data processing unit, and the dynamic routing algorithm is adopted to transmit data between capsule layers, which has better feature expression ability than CNN. Its basic architecture is shown in Figure 2, where the convolution layer extracts the classifying features from the original images, the primary capsule layer mainly transforms the upper scalar representation to a vector representation and outputs a vector, and the digital capsule uses a dynamic routing algorithm to update the network parameters and avoids the loss caused by pooling. The final output is the eigenvector whose length is the probability that the test sample belongs to a certain class.

In Figure 2, W represents the weight. In a fully connected neural network, every neuron is a scalar (that is, there is only one numeric value), so every weight is just a scalar and a numeric value. However, in CapsNet, each capsule neuron is a vector (that is, it contains multiple values, such as $[x_1, x_2, x_3, \ldots, x_n]$; the specific number n is designed according to the network), so the weight of each capsule neuron W should also be a vector. It is still updated according to backpropagation.

The input s of CapsNet is obtained as follows:

$$s_j = \sum_i c_{ij}\hat{u}_{j|i},\ .\ \hat{u}_{j|i} = W_{ij}u_i \qquad (2)$$

where u is the output of CapsNet of the upper layer and W_{ij} is the learnable weight matrix between the ith capsule and jth capsule; to be multiplied by each output, the coupling coefficient c added to the linear sum stage, is calculated by

$$c_{ij} = \mathrm{Soft} \quad \max(b_{ij}) = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \qquad (3)$$

In the process of calculating s by forward propagation, W is set as a random value, b is initialized to 0, u is the output of the previous layer, and s of the next layer can be obtained. Sigmoid is often used as an activation function in FCN, while Squashing is an activation function. Its output v is as follows:

$$v_j = \frac{||s_j^2||}{1 + ||s_j^2||}\frac{s_j}{||s_j||} \qquad (4)$$

In Sq. (4), the former part $||s_j||^2/(1+||s_j||^2)$ of the activation function is the scale of the input vector s, and the latter part $s_j/||s_j||$ is the unit vector of s. This activation function not only preserves the direction of the input vector but also compresses the modulus of the input vector to between [0, 1]. It is regarded as the probability of an entity's appearance.

Dynamic routing is employed to update b and then update c, as follows:

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j \qquad (5)$$

Other convolution parameters of the entire network and W need to be updated according to the loss function, as follows:

$$L_c = \sum_{k \in CNum} T_k \max(0, m^+ - ||V_k||^2) + \lambda(1$$
$$- T_k) max(0, ||V_k|| - m^-)^2 \qquad (6)$$

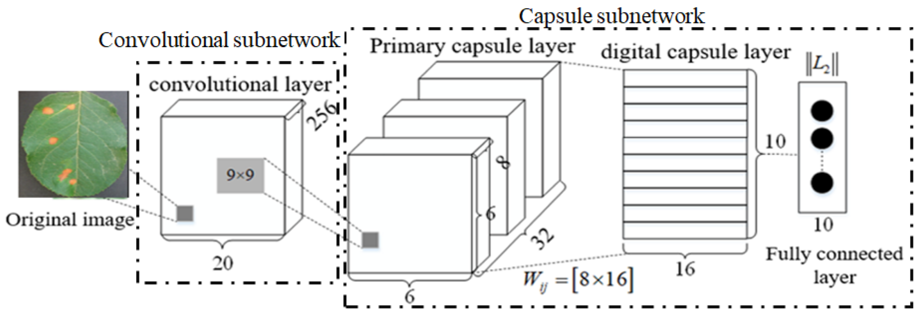


FIGURE 2
Architecture of capsule network (CapsNet).

where m^+ and m^- are the category prediction values, Λ is the balance coefficient, T_k is the label of category, $T_k = 1$ is the correct label, *CNum* is the number of disease categories, k is the category number, and $||V_k||$ is the length of the vector representing the probability of discriminating as the class kth disease; the total loss is the sum of all digital capsule loss functions. The default values are set as $m^+ = 0.9$, $m^- = 0.1$, and $\Lambda = 0.5$.

Dilated convolution capsule network

In complex image classification methods based on CNN and its variants, a large number of labeled training samples are usually required to train their parameters and improve their performance. However, it is very time-consuming to label a large number of samples. Although increasing network depth can improve the recognition rate, it means increasing network training time to optimize a large number of parameters. Traditional CapsNet only uses one convolution layer to extract the classification features, which cannot extract the deep multi-scale features from the complex images of disease leaves, resulting in low disease identification accuracy. To overcome the above problem, a DCCapsNet is constructed for apple disease recognition. Its architecture is shown in Figure 3, consisting of a convolution subnetwork and capsule subnetwork.

In DCCapsNet, Conv 1 of the convolution subnetwork is the same as the convolutional layer in CapsNet, and the capsule subnetwork is the same as the capsule layer in CapsNet, while Conv 2 and Conv 3 are two additional dilated Inception modules, which are introduced to enhance deep multi-scale feature extraction capability, thus improving the feature learning ability on complex disease leaf image dataset.

For the perception of the convolution kernel, the larger the convolution is, the stronger the ability of extracting disease information is. In fact, the lesions are smaller than the whole image, and other information on the image can be regarded as "noise", which needs to be filtered. As a consequence, the dilated Inception module is designed as shown in Figure 4A (Janakiramaiah et al., 2021). The traditional Inception module is also shown in Figure 4B for comparison.

By comparing Figures 4A, B, it can be seen that DCCapsNet has more different receptive fields, such as 1 × 1, 3 × 3, 5 × 5, and 7 × 7. Since the 5 × 5 convolutions in Figure 4B are replaced by a 3 × 3 dilated convolution, the number of its convolution kernel parameters is smaller. The superiority of DCCapsNet is described as follows.

1. Adding two convolutional layers. The disease leaf images are often complex with irregular and multi-scale spots and contain an amount of healthy region and noise. To reduce the interference of useless information, the relationship between various features in the image can be fully connected, and the healthy region and noises can be filtered before entering the primary capsule layer. After Conv 1, Conv 2 and Conv 3 are added to reduce the interference caused by redundant information in complex backgrounds.
2. Dimension extension of capsules. After three convolutional modules, a large number of deep-level multi-scale features of the input images are extracted, and the extracted features are processed by the primary capsule layer and digital capsule layer and then compressed into capsules. The typical structure of the network is the capsule structure, which is the unit of storing information. When the dimension of the capsule structure is larger, there are enough storage units to store effective information in the network. Therefore, the network extends its dimension to 10D.
3. Intermediate capsule. In the capsule layer, the feature capsule at the bottom predicts the feature of the upper layer by attitude relation and then activates the upper layer by dynamic routing algorithm and selection decision mechanism.

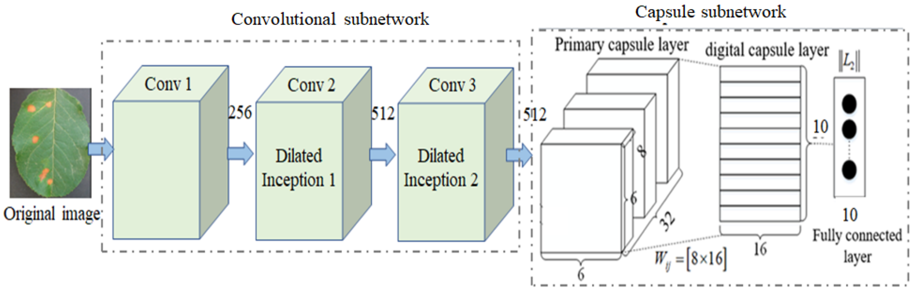


FIGURE 3
dilated convolution capsule network (DCCapsNet) architecture.

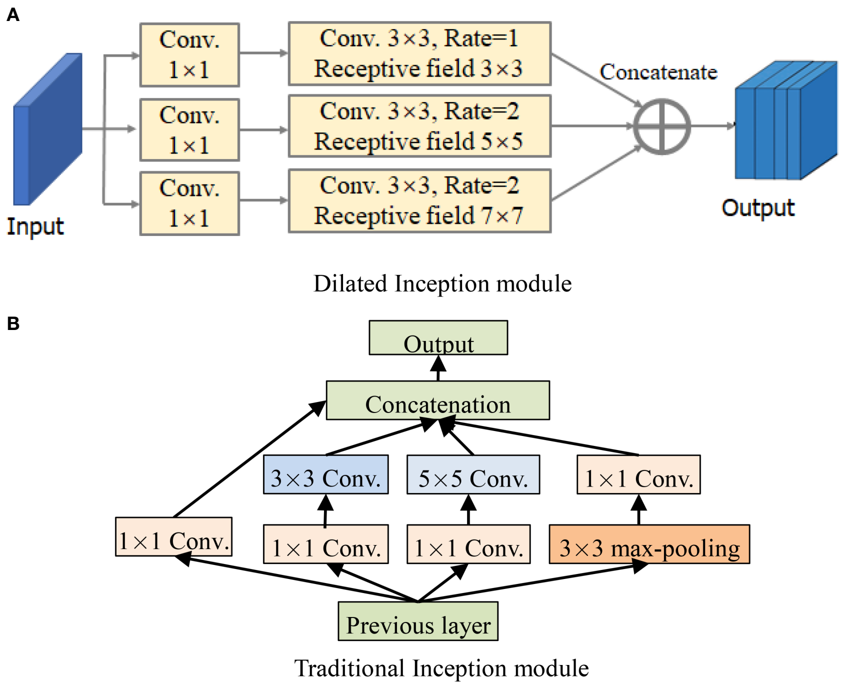


FIGURE 4
Structures of Inception module and dilated Inception module. **(A)** Dilated Inception module. **(B)** Traditional Inception module.

The operation of DCCapsNet is as follows. In Conv 1, the input color image is first convolved with 256 convolution kernels of 3 × 3 size, and the convolution step is 1. The ReLu activation function is also used during the coiling operation. In Conv 2, dilated Inception module is used to carry out multi-scale convolution operation for the preliminary features obtained by Conv 1 convolution, and the convolution step is 1, so as to obtain the output results of the Conv 2 layer. In Conv 3, further carry out a dilated Inception module on the features obtained by Conv 2 convolution layers. In the primary capsule layer, vectorize the output results of Conv 3 layer. Ten groups of different convolutional kernels are adopted, and each group of coil-product kernels contained different convolutional kernels. The step of convolution is set as 1, and the activation function of this convolution operation is ReLu. After this step, the low-level feature is obtained, which is a vector of 1 × 10.

Dilated Inception module is composed of 1 × 1, 3 × 3, 5 × 5, and 7 × 7 convolutional kernels and a 3 × 3 maximum pooling in parallel. Its advantage is that four receptive fields with four sizes are used to extract the multi-scale features without increasing the parameters of the kernels individually at each stage of the network. Multi-scale kernels have better feature expression effects on the input complex images, so dilated Inception module has a better feature expression ability by the parallel configuration of the kernels. To test DCCapsNet on disease leaf images, the k-dimension feature vectors extracted by the capsule subnetwork are input into the Softmax classifier, which is described as follows:

$$P(Y = i|x) = Soft\max(Y_i) = \frac{exp(\varpi_i Y_i)}{\sum_{i=1}^{K} exp(\varpi_k Y_k)} \tag{7}$$

where P is the probability that the feature vector x belongs to the ith category, K is the total number of categories, ϖ is the weight items, and y_i is the corresponding label of the ith training sample.

The average recognition rate of apple disease experiments is often adopted to test the network performance. The test images in each class are used to measure the classification accuracy, which is calculated as follows:

$$Accuracy = \frac{\text{Number of disease leaf images correctly identified}}{\text{Total number of test disease leaf images}} \tag{8}$$

The number of floating point operations (FLOPs), including multiplication and addition, depends on the model and can be used to evaluate model complexity. It is used as a criterion to assess the complexity of the model. To compute the number of FLOPs, suppose the convolution is implemented as a sliding window and the nonlinearity function is computed for free. For convolution layers, the FLOPs are computed as

$$FLOPs= (2C_{in}K^2 - 1)HWC_{out} \tag{9}$$

where H, W, and C_{in} are the height, width, and the number of channels of the input feature map, respectively; K is the kernel size (assumed to be symmetric); C_{out} is the number of output channels.

For fully connected layers, the FLOPs are computed as follows:

$$FLOPs= (2S_{in} - 1)S_{out} \tag{10}$$

where S_{in} is the input dimensionality or the number of input neurons and S_{out} is the output dimensionality or the number of output neurons.

The FLOPs of the model are the sum of the FLOPs of the convolution layers and fully connected layers.

Experiments and analysis

In this section, many experiments of apple disease recognition are conducted to validate the proposed method DCCapsNet and compared with improved convolutional neural network (ICNN) (Yan et al., 2020), VGG-ICNN (Thakur et al., 2022), LAD-Net (Zhu et al., 2022), and RegNet (Li et al., 2022). The comparative experiments and results are analyzed and discussed. The experimental configuration is shown in Table 1.

Dataset

The dataset of apple disease leaf images built by Northwest A&F University was used in the experiment. The dataset contains 26,377 images of five common apple disease leaves taken by BM-500GE color camera in an outdoor environment and laboratory environment. The data distribution are shown in Table 2. The dataset is randomly divided into a training set and a test set, in which the training set is used for training parameters, and the test set is used to verify the model. Five simple disease leaf images and five complex disease leaf images are shown in Figure 5.

TABLE 1 Experiment configuration.

Experimental configuration	Parameter value
Processor	Intel Xeon E5-2643v3@3.40GHz
Graphics card	GTX2080Ti11 GB 64 GB
Memory	32 GB
Disk	100 GB
Deep learning framework	PaddlePaddle 1.8.4
Operating system	Ubuntu 16.04.1 LTS (64 bit)
Other tools	Python 3.7.1 CUDA Toolkit10.0 Pytorch

As can be seen from Figure 5, the color and texture of rust and brown spots are similar with little difference. Due to different shooting conditions and complex backgrounds, the same subclasses may be affected by a single leaf or a cluster of leaves, leading to a large gap within classes. Therefore, a CNN-based method has a high probability of misjudgment in the process of disease identification. Image annotation is a crucial step in building the dataset. It is used to mark out the location and category of diseased spots in infected leaves. In this section, a tool has been developed to annotate images through rectangular bounding boxes. With the use of the annotation tool and the knowledge of experienced agriculture experts, areas of diseased spots in the image can be accurately labeled. When the annotation is complete, an XML file is generated for each image, which includes the types of diseased spots and their locations. The annotated image is shown in Figure 6A, and the infected areas are surrounded by boxes. Figure 6B is a fragment of the generated XML file, in which the disease name of rust is described and the location of diseased spots is determined by the upper left and lower right coordinates of the box.

Experimental results

Experimental parameters are set as follows. Batch size is 16, the number of iterations is 3,000, the initial learning rate is 0.0005, and the momentum is 0.9. As the number of iterations increases, the learning rate is decreased by 0.05 times. If the loss of the network does not decrease after 10 iterations during training, stop the training. Each image is uniformly normalized to 512 × 512. The network parameters are initialized to generate weight parameters with a mean value of 0 and variance of 1, conforming to normal distribution. The average recognition accuracy is used to measure the performance of the network.

DCCapsNet and four comparative deep learning models—ICNN, VGG-ICNN, LAD-Net, and RegNet—are trained on the image training set of apple disease leaves, from the beginning of the model training to convergence, so as to ensure that the training conditions of these models are the same. Each model is trained from the beginning until the model converged, and the training conditions of each model are guaranteed to be the same for a fair comparison. Their training losses versus the number of training iterations on the training set are shown in Figure 7, which can more intuitively display the performance changes of these models in the training process.

It can be seen from Figure 7 that DCCapsNet has better convergence performance and recognition performance than other networks, and its convergence is relatively fast; the change in trend after 1,000 training iterations is relatively stable. Within the 3000th training iteration, all models converge basically, and before the 1000th training iteration,

TABLE 2 Apple disease leaf image distribution.

Apple leaf disease	Dataset	Training set	Test set
Mosaic	4,875	3,412	1,463
Brown spot	5,655	3,958	1,697
Rust	5,694	3,985	1,709
Gray spot	4,810	3,367	1,443
Spotted leaf litter	5,343	3,740	1,603
Total	26,377	18,462	7,915

the loss of each network model decreases greatly, and the loss of each network model shows a downward trend as a whole. After 2,000 training iterations, the convergence performances of all models are improved and tend to be stable.

The apple disease recognition experiments are carried out with a fivefold cross-validation scheme. To be fair, four trained models are chosen after 3,000 training iterations to identify the leaf disease images in the test set. Their recognition results are shown in Table 3.

From Table 3, it can be seen that DCCapsNet achieves the highest identification accuracy of 93.16%. Compared with ICNN, VGG-ICNN, LAD-Net, and RegNet, the recognition accuracy is improved by 4.04%, 2.05%, 0.99%, and 3.52%, respectively. DCCapsNet has fewer FLOPs and has higher PA than other models except for RegNet. RegNet is a lightweight convolutional network with 5.2M training parameters and has the least FLOPs because it aims to design spaces and find some network design principles, rather than just search for a set of parameters.

To verify the effectiveness of dilated Inception modules, several kinds of experiments are set up by introducing several Inceptions and dilated Inceptions into the convolution subnetwork of CapsNet. The modified networks are similar to DCCapsNet. The structures of Inception and dilated Inception are shown in Figures 4A, B. The experimental conditions are the same as above. The results of CapsNet and modified CapsNet are shown in Table 4.

From Table 4, the conclusions obtained are summarized as follows. In general, adding convolutional modules can improve the recognition rate, while adding dilated Inceptions can further increase accuracy and reduce model training time. The main reason is that, compared with Inception, dilated Inception has four different-scale convolutional kernels without increasing additional training parameters, which can extract multi-scale features by applying different convolutional kernels in parallel and cascading their output feature maps. Its advantage is that there is no need to set the parameters of the convolutional kernels separately in each stage of the network. Multi-scale convolution has a better feature expression effect on the irregular disease leaf image, so Inception can have better feature expression ability through the parallel configuration of the convolution kernel. Dilated Inception is superior to Inception because it has different convolutional kernels with different respective fields without increasing training parameters.

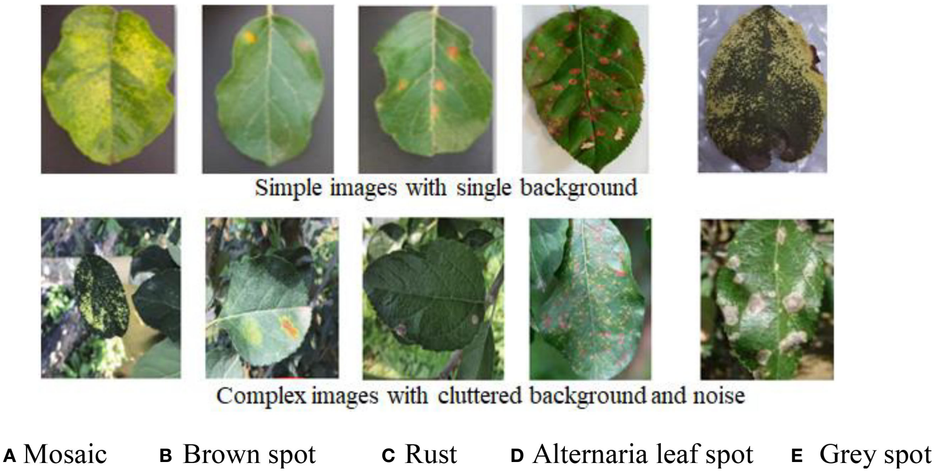
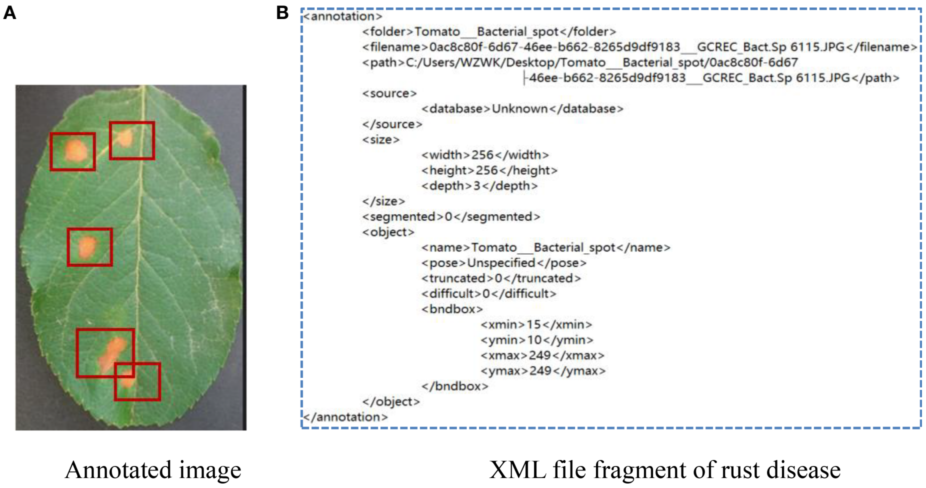

Simple images with single background
Complex images with cluttered background and noise
A Mosaic
B Brown spot
C Rust
D Alternaria leaf spot
E Grey spot

FIGURE 5
Ten image examples of five kinds of diseases. **(A)** Mosaic. **(B)** Brown spot. **(C)** Rust. **(D)** *Alternaria* leaf spot. **(E)** Gray spot.



```
<annotation>
        <folder>Tomato__Bacterial_spot</folder>
        <filename>0ac8c80f-6d67-46ee-b662-8265d9df9183__GCREC_Bact.Sp 6115.JPG</filename>
        <path>C:/Users/WZWK/Desktop/Tomato__Bacterial_spot/0ac8c80f-6d67
                        -46ee-b662-8265d9df9183__GCREC_Bact.Sp 6115.JPG</path>
        <source>
                <database>Unknown</database>
        </source>
        <size>
                <width>256</width>
                <height>256</height>
                <depth>3</depth>
        </size>
        <segmented>0</segmented>
        <object>
                <name>Tomato__Bacterial_spot</name>
                <pose>Unspecified</pose>
                <truncated>0</truncated>
                <difficult>0</difficult>
                <bndbox>
                        <xmin>15</xmin>
                        <ymin>10</ymin>
                        <xmax>249</xmax>
                        <ymax>249</ymax>
                </bndbox>
        </object>
</annotation>
```

FIGURE 6
Annotation of apple rust disease leaf image. **(A)** Annotated image. **(B)** XML file fragment of rust disease.

From Table 4, it is also seen that the accuracy rates show an upward trend versus adding Inception or dilated Inception modules, and dilated Inception is better than Inception. However, adding three dilated Inception modules can greatly improve the identification accuracy while increasing the long training time. However, the addition of three dilated Inception modules can slightly improve the accuracy of recognition while greatly increasing the training time. Dilated Inceptions with four dilated rates have five different convolution kernels, such as 1 × 1, 3 × 3, 5 × 5, 7 × 7, and 9 × 9. When two dilated Inceptions with four dilated rates are added, the accuracy decreases instead of improving, indicating the dilated Inception module with convolution kernel 9 × 9 is not suitable for the image classification of disease leaves. Finally, the dilated Inception with dilated rate r = 1, 2, and 3 is selected.

To verify the effect of the dilated Inception module on multi-scale features, Figure 8 shows the visualization of convolutional feature maps of DCCapsNet. From Figure 8, it can be seen that DCCapsNet can obtain the multi-scale and multi-level feature by dilated Inception with three dilated rates.

Result analysis

The results of Figure 7 and Tables 3, 4 show that DCCapsNet has the highest recognition rate and the least FLOPs except for RegNet. The reason is that it makes use of the advantages of dilated Inception module and CapsNet. RegNet has the fewest FLOPs, but its recognition rate is lower but slightly higher than that of ICNN. LAD-Net is the next best because it uses LAD-

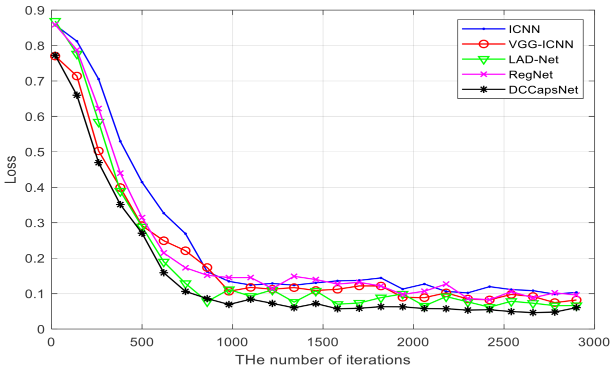


FIGURE 7
Losses of five networks versus training iterations.

TABLE 3 The recognition results of ICNN, VGG-ICNN, LAD-Net, RegNet, and DCCapsNet.

Method	ICNN	VGG-ICNN	LAD-Net	RegNet	DCCapsNet
Pixel Seg. accuracy (PA)	89.12	91.11	92.17	89.64	93.16
FLOPs (G)	44.5	45.7	42.5	27.4	41.8
Training time (h)	7.51	6.41	7.17	6.50	3.44
Testing time (s)	3.18	2.82	3.19	3.73	2.51

TABLE 4 The results of CapsNet and modified CapsNet with different Inception modules.

Insert module into CapsNet	Accuracy	Training time
0 Inception, i.e., CapsNet	82.63	8.12 h
1 Inception	86.52	6.74 h
2 Inceptions	89.73	5.25 h
3 Inceptions	90.14	5.97 h
1 dilated Inception	90.15	4.76 h
2 dilated Inceptions, i.e., DCCapsNet	93.16	3.44 h
3 dilated Inceptions	93.18	4.61 h
1 Inception and 1 dilated Inception	92.06	5.11 h
2 dilated inceptions with 4 dilated rates	93.11	3.83 h

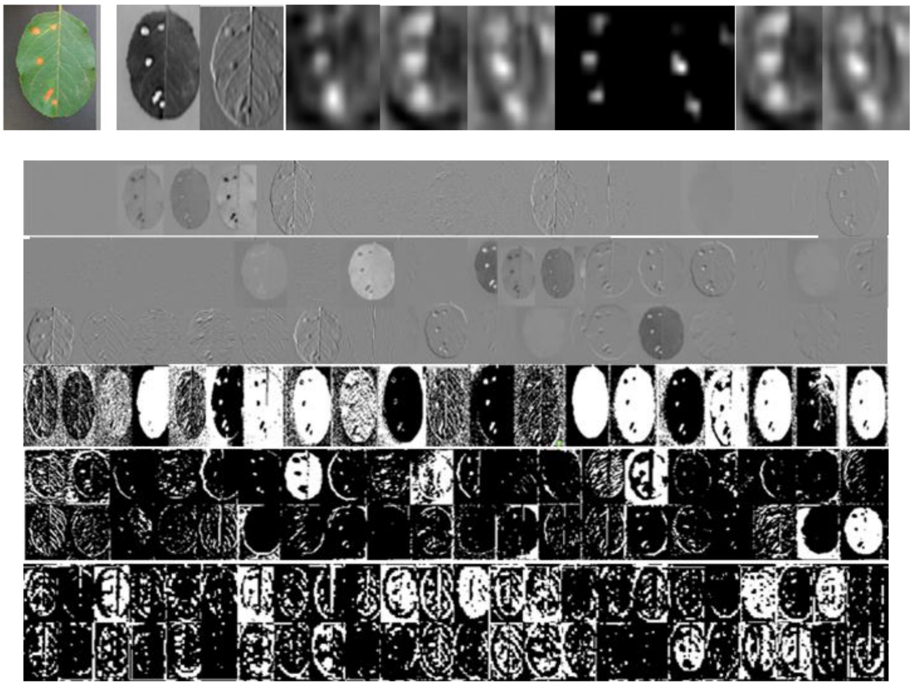


FIGURE 8
An original image and its feature map examples in different convolutional layers.

Inception and attention mechanism to enhance the ability to extract multi-scale features of different sizes of disease spots and replaces a full connection with global average pooling to further reduce parameters. Although it is a lightweight model, it has little higher FLOPs than DCCapsNet due to the attention mechanism. VGG-ICNN is better than ICNN because it has few training parameters and has three Inception v7 blocks to extract the multi-scale features.

The result validates that when the depth of the network reaches a certain level, increasing convolutional layers of the network again is not as significant as expected, but as the depth of the network model increases, the model becomes more complex and the training time becomes longer. Therefore, ICNN is not easy to converge. Compared with ICNN and RegNet, DCCapsNet has better convergence performances due to the multi-branch parallel structure of dilated Inception, indicating that a multi-branch network is superior to a single-branch network in the disease identification task. It can extract multi-scale image features. Compared to VGG-ICNN and LAD-Net, DCCapsNet adds two dilated Inception modules that can extract rich features and overcome well the adverse effects of complex background environments and disease spots.

Conclusion

CNN focuses on detecting important features of the input image and obtains invariance by pooling but loses some local information. Its output is only one scalar value, while the output of CapsNet is a vector, which not only can represent the characteristics of the input image but also can include the direction and state of the target. It is suitable for irregular disease leaf image classification, but its recognition accuracy is not high because there is only one convolutional layer. To improve accuracy, a DCCapsNet is constructed for apple leaf disease identification. Multi-scale classification features are extracted by adding two dilated Inception modules into CapsNet. The results on the apple disease leaf image dataset show that DCCapsNet is superior to other networks in recognition rate and training performance. This method has stronger practical application capabilities to promote the development of intelligent management systems for crop diseases in the field. In the future, we will embed this work into a smartphone-based disease diagnostic system for farmers in remote places.

Data availability statement

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

Author contributions

CX designed and performed the experiment, analyzed the data, trained the algorithms, and wrote the manuscript. CX and XW collected data. SZ selected the algorithm and monitored the data analysis. XW and SZ conceived the study and participated in its design. All authors contributed to this article and approved the submitted version.

Funding

This work is supported by the National Natural Science Foundation of China (Nos. 62172338 and 62072378).

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., Kaliyar, R. K., Singal, G., and Gupta, S. K. (2019). FCNN-LDA: A faster convolution neural network model for leaf disease identification on apple's leaf dataset. *12th International Conference on Information and Communication Technology and System (ICTS*. IEEE 246–251. doi: 10.1109/ICTS.2019.8850964

Ahmed, K. R. (2021). Smart pothole detection using deep learning based on dilated convolution. *Sensors* 21 (24), 8406. doi: 10.3390/s21248406

Deng, F., Pu, S., Chen, X., Shi, Y., Yuan, T., and Pu, S. (2018). Hyperspectral image classification with capsule net-work using limited training samples. *Sensors* 18 (9), s18093153. doi: 10.3390/s18093153

Di, J., and Li, Q. (2022). A method of detecting apple leaf diseases based on improved convolutional neural network. *PloS One* 17 (2), 1–15. doi: 10.1371/journal.pone.0262629

Fang, Y., Li, Y., Tu, X., Tan, T., and Wang, X. (2019). Face completion with hybrid dilated convolution. *Signal Process. Image Commun.* 80, 115664. doi: 10.1016/j.image.2019.115664

Janakiramaiah, B., Kalyani, G., Prasad, L. V. N., Karuna, A., and Krishna, M. (2021). Intelligent system for leaf disease detection using capsule networks for horticulture. *J. Intel. Fuzzy Syst.: Appl. Eng. Technol.* 41 (6), 6697–6713. doi: 10.3233/JIFS-210593

Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7, 59080. doi: 10.1109/ACCESS.2019.2914929

Khana, A., Quadrib, M. K., Banday, S., and Shah, J. L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *Comput. Electron. Agric.* 198, 107093. doi: 10.1016/j.compag.2022.107093

Lei, J., Gao, X., Song, J., Wang, X. L., and Song, L. M. (2018). Survey of deep neural network model compression. *J. Softw.* 29, 251–266. doi: 10.13328/j.cnki.jos.005428

Li, L., Zhang, S., and Wang, B. (2022). Apple leaf disease identification with a small and imbalanced dataset based on lightweight convolutional networks. *Sensors* 22 (1), 173. doi: 10.3390/s22010173

Li, J. Y., Zhao, Y. K., Xue, Z. E., Cai, Z., and Li, Q. A. (2019). A Survey of model compression for deep neural networks. *Chin. J. Eng.* 41, 1229–12399. doi: 10.13374/j.issn2095-9389.2019.03.27.002

Luo, Y., Sun, J., Shen, J., Wu, X., Wang, L., and Zhu, W. (2021). Apple leaf disease recognition and Sub-class categorization based on improved multi-scale feature fusion network. *IEEE Access* PP (99), 1–15. doi: 10.1109/ACCESS.2021.3094802

Mishra, B., Nema, S., Lambert, M., and Nema, S. (2017). "Recent technologies of leaf disease detection using image processing approach-a review," in *4th International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 17573977. doi: 10.1109/ICIIECS.2017.8275901

Pandiyan, S., Ashwin, M., Manikandan, R., Karthick Raghunath, K. M., and Anantha Raman, G. R. (2020). Heterogeneous internet of things organization predictive analysis platform for apple leaf diseases recognition. *Comput. Commun.* 154, 99–110. doi: 10.1016/j.comcom.2020.02.054

Puspha Annabel, L. S., Annapoorani, T., and Deepalakshmi, P. (2019). "Machine learning for plant leaf disease detection and classification – a review," in *International Conference on Communication and Signal Processing*. India: IEEE, 0538–0542. doi: 10.1109/ICCSP.2019.8698004

Sun, J., Tan, W. J., Mao, H. P., Wu, X. H., Chen, Y., and Wang, L. (2017). Recognition of multiple plant leaf diseases based on improved convolutional neural network. *Trans. Chin. Soc Agric. Eng.* 33, 209–215. doi: 10.11975/j.issn.1002-6819.2017.19.027

Sun, H., Xu, H., Liu, B., He, D., He, J., Zhang, H. T., et al. (2021). MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. *Comput. Electron. Agric.* 189, 106379. doi: 10.1016/j.compag.2021.106379

Thakur, P. S., Sheorey, T., and Ojha, A. (2022). VGG-ICNN: A lightweight CNN model for crop disease identification. *Multim. Tools Appl.*, 1–24. doi: 10.51470/PLANTARCHIVES.2022.v22.no1.035

Wang, D., Xu, Q., Xiao, Y., Tang, J., and Luo, B. (2019). "Multi-scale convolutional capsule network for hyperspectral image classification," in *Pattern Recognition and Computer Vision - Second Chinese Conference (PRCV)* (Xi'an, China,), 11858. , 749–760.

Xiang, C., Zhang, L., Tang, Y., Zou, W., and Xu, C. (2018). MS-CapsNet: a novel multi-scale capsule network. *IEEE Signal Process. Lett.* 25 (12), 1850–1854. doi: 10.1109/LSP.2018.2873892

Xia, X., Sun, Q., Shi, X., and Chai, X. (2020). Apple detection model based on lightweight anchor-free deep convolutional neural network. *Smart Agric.* 2 (1), 99–110. doi: 10.12133/j.smartag.2020.2.1.202001-SA00 4

Yang, F., Li, W., Tang, W., and Wu, X. (2018). "The analysis between traditional convolution neural network and capsuleNet," in *International Conference on Control and Automation*. (Hangzhou, China: ICCAIS), 210–215.

Yan, Q., Yang, B., Wang, W., Wang, B., Chen, P., Zhang, J., et al. (2020). Apple leaf diseases recognition based on an improved convolutional neural network. *Sensors* 20 (12), 3535. doi: 10.3390/s20123535

Yu, H. J., and Son, C. H. (2020). "Leaf spot attention network for apple leaf disease identification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 229–237 (Seattle, WA, USA: IEEE). doi: 10.1109/CVPRW50498.2020.00034

Zhang, C., Zhang, S., Yang, J., Shi, Y., and Chen, J. (2017). Apple leaf disease identification using genetic algorithm and correlation based feature selection method citation. *Int. J. Agric. Biol. Eng.* 10 (2), 74–83. doi: 10.3965/j.ijabe.20171002.2166

Zhong, Y., and Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* 168, 105146. doi: 10.1016/j.compag.2019.105146

Zhu, X., Li, J., Jia, R., Liu, B., Yao, Z., Yuan, A., et al. (2022). LAD-net: A novel light weight model for early apple leaf pests and diseases classification. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 35849665, 1–14. doi: 10.1109/TCBB.2022.3191854

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 14 December 2022
DOI 10.3389/fpls.2022.992700

OPEN ACCESS

EDITED BY
Gregorio Egea,
University of Seville, Spain

REVIEWED BY
Orly Enrique Apolo-Apolo,
Faculty of Bioscience Engineering,
Ghent University, Belgium
Borja Espejo-Garca,
Agricultural University of
Athens, Greece

*CORRESPONDENCE
Zahraa Al Sahili
zma35@mail.aub.edu

SPECIALTY SECTION
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

RECEIVED 12 July 2022
ACCEPTED 27 September 2022
PUBLISHED 14 December 2022

CITATION
Al Sahili Z and Awad M (2022) The power of transfer learning in agricultural applications: AgriNet. *Front. Plant Sci.* 13:992700.
doi: 10.3389/fpls.2022.992700

The power of transfer learning in agricultural applications: AgriNet

Zahraa Al Sahili* and Mariette Awad

Department of Electrical and Computer Engineering, Maroun Semaan Faculty of Engineering, American University of Beirut, Beirut, Lebanon

Advances in deep learning and transfer learning have paved the way for various automation classification tasks in agriculture, including plant diseases, pests, weeds, and plant species detection. However, agriculture automation still faces various challenges, such as the limited size of datasets and the absence of plant-domain-specific pretrained models. Domain specific pretrained models have shown state of art performance in various computer vision tasks including face recognition and medical imaging diagnosis. In this paper, we propose AgriNet dataset, a collection of 160k agricultural images from more than 19 geographical locations, several images captioning devices, and more than 423 classes of plant species and diseases. We also introduce AgriNet models, a set of pretrained models on five ImageNet architectures: VGG16, VGG19, Inception-v3, InceptionResNet-v2, and Xception. AgriNet-VGG19 achieved the highest classification accuracy of 94% and the highest F1-score of 92%. Additionally, all proposed models were found to accurately classify the 423 classes of plant species, diseases, pests, and weeds with a minimum accuracy of 87% for the Inception-v3 model. Finally, experiments to evaluate of superiority of AgriNet models compared to ImageNet models were conducted on two external datasets: pest and plant diseases dataset from Bangladesh and a plant diseases dataset from Kashmir.

KEYWORDS

transfer learning, convolutional neural network, agriculture, pretrained models, plant disease, pest, weed, plant species

Introduction

The world population is expected to reach over 9 billion by 2050, which will require an increase in food production by 70% (Silva and M. S. U. E, 2021). Considering scarcity of resources and climate change, intervention of artificial intelligence (AI) in agriculture is needed to overcome this challenge (Talaviya et al., 2020). AI advantages can span from plant diseases detection, robotic weeds and pests control, to herbal discovery. Plant diseases are not only a risk for food security only, but they also have disastrous effects on smallholder farmers where pests and weeds can lead to the destruction of around 50% of the farm's plants (Rachman et al., 2017). Automated recognition of weeds, pests, and plant diseases can support smallholder farmers through free diagnosis services using mobile applications. Additionally, weed control robotics and sensor monitoring are another form of automation applied in regions with a limited number of agricultural expertise. Another important detection task is automated plant species recognition which is used in medical herbal research and in preventing extinction of non-discovered plant species (Tan et al.,).

Historically, the recognition task was relying on algorithms that needs handcrafted features, which were processed using relatively simple discriminative models such as linear classifiers or support vector machines (SVM) (Halevy et al., 2009; Rumpf et al., 2012; Wäldchen and Mäder, 2018; Madsen et al., 2020). After being the leading algorithm in all computer vision tasks, deep learning (DL) has been widely used in agriculture research for plant classification tasks (Madsen et al., 2020). To achieve good accuracy, DL models need very large datasets for their requirements as data-hungry neural networks (Halevy et al., 2009; Madsen et al., 2020). In the agricultural domain, datasets size is limited, so transfer learning would allow models reach higher accuracy without the need for more field data (Gandorfer et al., 2022). However, pretrained models used for transfer learning are not agriculture domain specific and were trained general computer vision datasets such as ImageNet. This creates a big challenge since convolutional models moves from low level features to higher level features and can lead to negative transfer (Gandorfer et al., 2022). For example, Yan et al. proposed transfer learning framework based on synthetic images to improve *in-vitro* soybean segmentation (Yang et al., 2022). The proposed framework resulted in a precision improvement of 8% considering the data abundancy in soybean applications (Yang et al., 2022).

Another challenge is models' robustness which is affected by the type of agricultural data use. Mohanty et al. compared usage of AlexNet and Google LeNet pretrained models for 26 diseases and 14 crops species through PlantVillage dataset which constitutes of 54,306 lab images. The Google LeNet achieved the highest accuracy of 99.3% (Mohanty et al., 2016). However, upon testing on trusted online sources, the accuracy dropped drastically to 31.4% (Mohanty et al., 2016). After that Singh et al. introduced PlantDoc, a 2,598 field images dataset of 13 crops and 27 classes (Singh et al., 2020a). To classify the dataset's images, multiple experiments were done on both uncropped and cropped images. For the non-cropped images, using ImageNet architectures with PlantVillage weights (Mohanty et al., 2016) resulted in twice accuracy compared to using the same architectures but with ImageNet weights (Singh et al., 2020a). In the cropped dataset experiment, transfer learning on VGG16 architecture with ImageNet weights resulted in an accuracy of 44.52% compared to 60.42% accuracy when VGG16 was used with plant village weights (Singh et al., 2020a).

Class imbalance degrades the performance of deep learning models on small classes including agricultural applications. For example, transfer learning was applied through ResNet50 architecture by Thapa et al. to detect two common apple diseases: apple scab and apple rust (Thapa et al., 2020). The accuracy obtained was 97% with an accuracy of only 51% for mixed diseases, which was caused by the small number of apples that have both diseases (Thapa et al., 2020). Additionally, Teimouri et al. used deep learning in the estimation of the weed growth stage (Teimouri et al., 2018). The dataset of 9649 images for various weed species was classified from 1 to 9 growth stages (Teimouri et al., 2018). Inception-v3 model was selected due to its good performance and low computational cost, and transfer learning was applied resulting in a 70% accuracy with a minimum accuracy of 46% for black-grass species that had the smallest set of images in the dataset (Teimouri et al., 2018).

Motivated to provide the agritech field with domain specific pretrained models that are robust and generalizable in various agricultural applications, the contributions of this work can be summarized as follows:

1. AgriNet dataset: a collection of 160k agricultural images from more than 19 geographical locations, several images captioning devices, and more than 423 classes of plant species and diseases.
2. AgriNet models: a set of pretrained models on five ImageNet architectures: VGG16, VGG19, Inception-v3, InceptionResNet-v2, and Xception and using AgriNet dataset. The proposed models are introduced to robustly classify the 423 classes of plant species, diseases, pests, and weeds with a minimum accuracy 94%, 92%, 89%,90%, and 88% for each architecture respectively.
3. Pretraining using AgriNet models: transfer learning using AgriNet models compared to ImageNet models was evaluated using experiments on two agricultural datasets: pest and plant diseases dataset from Bangladesh and a plant diseases dataset from Kashmir.

Materials and methods

Dataset

The AgriNet dataset is a collection of 160142 images belonging to 423 plant classes. The dataset was collected from 19 public datasets (The TensorFlow Team, Flowers (2019); Kumar et al., 2012a; Nilsback and Zisserman; Cassava disease classification (Kaggle); Olsen et al., 2019; Söderkvist, 2016; U. C. I. M. Learning, 2016; Giselsson et al., 2017; Peccia, 2018; Chouhan et al., 2019; J and Gopal, 2019; Krohling et al., 2019; Rauf et al., 2019; D3v, 2020; Huang and Chuang, 2020; Huang and Chang, 2020; Makerere AI Lab, 2020; Marsh, 2020; Singh et al., 2020b) geographically distributed between United States, Denmark, Australia, United Kingdom, Uganda, India, Brazil, Pakistan, and Taiwan. It includes field and lab images from different cameras and mobile devices, and it can perform multiple agricultural classification tasks, such as species, weed, pest, and plant diseases detection. Sample dataset images is displayed in Figure 1.

The dataset classes were constructed by merging the same classes from multiple datasets in one class. This provides better classification performance through training the neural network to classify images regardless of the image location, quality, and device, which was a common challenge reported in the literature. For example, for the tomato plant diseases, images were combined from datasets (Rauf et al., 2019; D3v, 2020; Makerere AI Lab, 2020) that included lab and field images from the United States, India, and Taiwan.

The collected dataset is highly imbalanced. As listed in Table 1, the average number of images per class is 378 images. Moreover, the number of classes with images less than 100 is 102 classes and the number of classes with images greater than 1000 is 44. In addition to class imbalance, a categorical imbalance between the three major tasks also exists. While training the models, the class weight mechanism was introduced to mitigate the class imbalance.

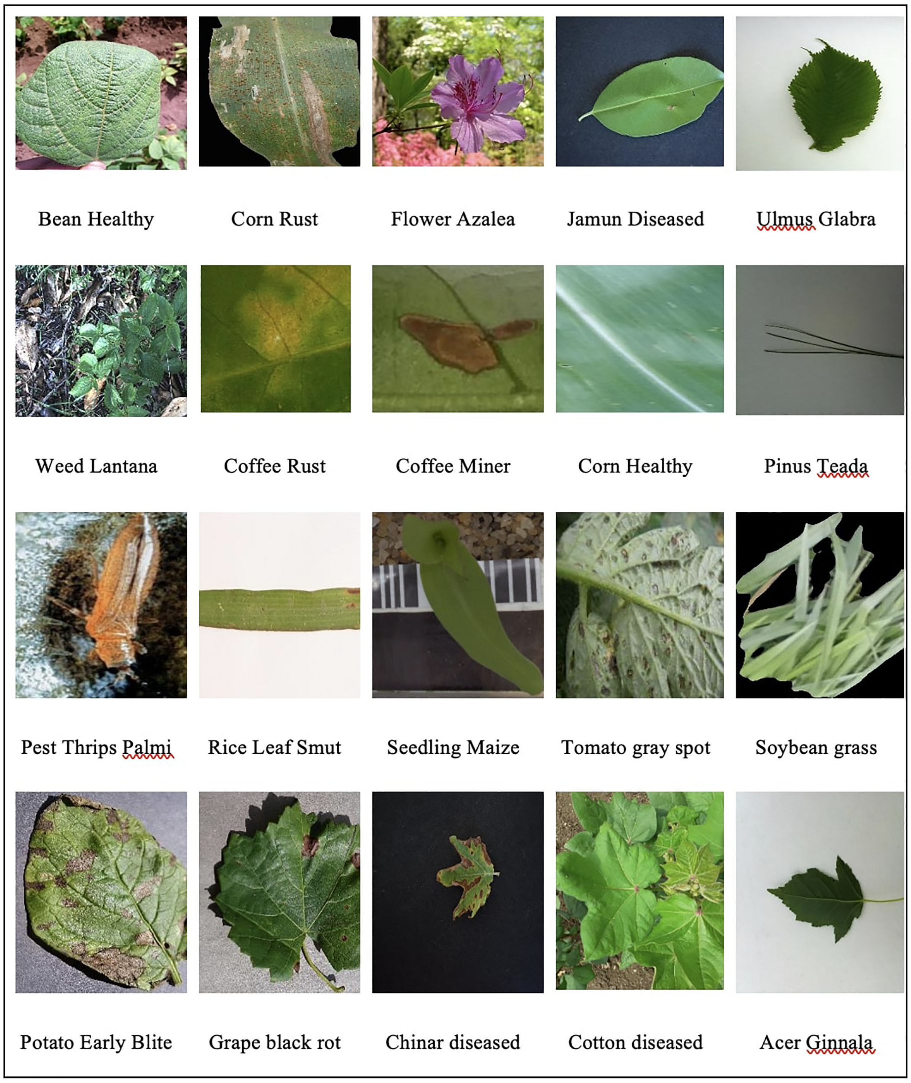


FIGURE 1
Sample images from agrinet dataset.

TABLE 1 Summary of AgriNet dataset per category.

Category	#Images	#Classes	Average	Median	Description	Reference
Species	52150	309	169	144	12 mushroom, 103 flowers, 194 leaves	(Kumar et al., 2012a; Nilsback and Zisserman, ; Olsen et al., 2019; Söderkvist, 2016; U. C. I. M. Learning, 2016; Huang and Chuang, 2020)
Pests & Weeds	38305	33	1161	672	8 weeds,5 soybean weeds,8 pests,12 seedling	(Giselsson et al., 2017; Peccia, 2018; J and Gopal, 2019; Singh et al., 2020b)
Diseases	69687	81	1700	491	30 species	(J and Gopal, 2019; Rauf et al., 2019; Krohling et al., 2019; Chouhan et al., 2019; Singh et al., 2020b; D3v, 2020; Huang and Chang, 2020; Makerere AI Lab, 2020)

Methods

Data preprocessing

The dataset is a collection of images from multiple sources. All images were converted to JPEG format and resized to 224x224 pixels which is the size recommended for the deep learning architectures used. The dataset was then split into 70% train,10% validation, and 20% test. To increase the dataset size and ensure that the model is more robust in classifying images when visual effects are modified, image augmentation was applied to the training set. The augmented images were generated through varying brightness, rotation, width shift, height shift, vertical flip, zoom, and shear.

Convolution neural network

A ConvNet is a sequence of layers where in every layer of a ConvNet one volume of activations is transformed to another volume through a differentiable function (Stanford, a). Three main types of layers are stacked to build a ConvNet architecture: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (Stanford, a). First, a Convolution layer computes the output of neurons that are connected to local regions in the input. Then, an activation function is applied, such as ReLU, which is max (0, x) thresholding at zero (Stanford, a). After that, a pooling layer performs down sampling operation along the spatial dimensions (width, height). Finally, the Fully-Connected layer is a classical neural network layer that computes the class scores (Stanford, a).

Deep learning architectures

Deep learning architectures that were frequently used in agricultural research were selected to train the AgriNet dataset.

VGG16 and VGG19

VGG is named for the Visual Geometry Group at Oxford and was introduced by Karen Simonyan and Andrew Zisserman in 2014 (Simonyan and Zisserman, 2015). The main contribution of this model was the usage of small-sized 3x3 convolutional filters. Pooling was done using Max-pooling over a 2 x 2-pixel window, with a stride of 2. VGG16 is the winning architecture of the ICLRLSVRC-2014 competition, having a top accuracy of 71.3% and a top-5 accuracy of 90.1% (Simonyan and Zisserman, 2015). The model has a depth of 16 and 143 million parameters. The main difference between VGG16 and VGG19, which was ranked second in the competition, is the model depth which is 19 in VGG19 (Simonyan and Zisserman, 2015). VGG19 achieved top accuracy of 71.3% and top 5 accuracies of 90% while having 138 million parameters (Simonyan and Zisserman, 2015).

Inception-v3

The inception model was introduced in 2012 by Szegedy et al. where the main contribution was “going deeper”. The model proposed was 27 layers deep, including inception layers. The inception layer is a combination of a (1×1 Convolutional layer, 3×3 Convolutional layer, 5×5 Convolutional layer) with their output filter banks concatenated into a single output vector forming the input of the next stage (Szegedy et al., 2015). Inception-v3 was introduced in 2016 as a convolutional neural network architecture from the Inception family with several improvements including usage of factorized 7 x 7 convolutions, label smoothing, and the use of an auxiliary classifier to propagate label information lower down the network (Szegedy et al., 2015). Those improvements resulted in a top accuracy of 77.9% and a top5 accuracy of 93.7%. It is 159 layers deep and has 23 million parameters (Szegedy et al., 2015).

Xception

Xception model was proposed by Chollet et al. in 2017. It stands for “extreme inception” taking the principle of inception to an extreme. It is a convolutional neural network architecture that relies solely on depth-wise separable convolution layers (Chollet, 2017; Akhtar, 2021). The main difference between inception and Xception is that in inception, 1x1 convolutions were used to compress the original input, and from each of those input spaces different type of filters was used on each of the depth space. On the other hand, Xception reverses this step where filters are applied followed by compression. The second difference is the absence of non-linearities in Xception compared to the usage of ReLU in inception (Chollet, 2017; Akhtar, 2021). The Xception achieved a top accuracy 79% of and a top5 accuracy 94.5% of while having 22.9M parameters and a depth of 126.

InceptionResNetv2

InceptionResNetv2 was proposed by Szegedy et al. in 2016 and builds on the Inception family of architectures but incorporates residual connections by replacing the filter concatenation stage of the Inception architecture (Elhamraoui et al., 2020; Szegedy et al., 2017). Residual connections allow shortcuts in the model leading to better performance while simplifying the Inception blocks (Elhamraoui et al., 2020; Szegedy et al., 2017). The model achieved top accuracy of 80.3% and a top 5 accuracy of 95.4% while having 55.9M parameters and a depth of 572.

Thus, each of the used architecture has its benefits depending on the targeted applications. A detailed comparison of the models is presented in Table 2. Note that Xception model has the smallest size of 88 MB while InceptionResNet-v2 achieved the highest top1-accuracy and top5-acuuracy of 0.803 and 0.953 respectively. In terms of depth and parameters, Xception model has the smallest number of parameters of 2291480 though VGG16 has the shortest depth of 23 layers.

Transfer learning

Transfer learning is the state-of-the-art approach with scarce data applications. The common approach for vision-based application is to train a ConvNet on a very large dataset (for example, ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest Singh, (2021). Three Transfer Learning methods exist:

ConvNet as fixed feature extractor

This is done by removing the fully connected layer from the ConvNet pretrained on a generic dataset (ex. ImageNet), then treating the rest of the ConvNet as a fixed feature extractor for the new dataset.

Fine-tuning the ConvNet

The second approach adds to the first approach by fine-tuning the weights of the pretrained network by continuing the backpropagation (Singh, 2021). Although retraining the whole model is possible, usually, some of the earlier layers are kept and we only fine-tune some higher-level portion of the network. This is because features of a ConvNet contain more generic features like edges in the first layers, but later layers of the ConvNet are more detail-oriented toward the pretrained model's classes (Singh, 2021).

Pretrained models

Final ConvNet checkpoints are frequently released to assist in fine-tuning tasks since modern ConvNets are time-consuming. For example, it takes 2-3 weeks to train a ConvNet across multiple GPUs on ImageNet (Singh, 2021).

First, transfer learning was applied on ImageNet pretrained models, where ImageNet was the generic dataset and AgriNet was the target dataset (Figure 2). After training the AgriNet models, the ¢architectures with their weights were saved and proposed as pretrained models for any other agricultural classification task.

Improving the models' performance

To tackle the bias in the DL models, the severe class imbalance in the dataset, and to improve the models' convergence, three main methods were applied to the five AgriNet architectures:

Class weights

Class imbalance can affect the classification accuracy of small classes compared to large classes. To improve the performance of classification in small datasets, multiple solutions exist including oversampling, under-sampling, and class weight. In AgriNet, class weights were added to all the trained neural networks so that a balance is created between classes during the training process (You et al., 2019).

Decaying learning rate

Learning rate decay is a technique for training neural networks, by starting with a large learning rate and then decaying it multiple times (Srivastava et al., 2014). It aims to improve optimization and generalization (Srivastava et al., 2014). This improvement is an outcome of the fact that an initially large learning rate accelerates training or helps the network escape spurious local minima, and then decaying the learning rate helps the network converge to a local minimum and avoid oscillation (Srivastava et al., 2014).

TABLE 2 Comparison of the imagenet architectures used in agrinet.

Model	Size	Top-1 accuracy	Top-5 accuracy	Parameters	Depth
Xception	**88 MB**	0.790	0.945	**22,910,480**	126
VGG19	549 MB	0.713	0.900	143,667,240	26
InceptionResNe-v2	215 MB	**0.803**	**0.953**	55,873,736	572
Inception-v3	92 MB	0.779	0.937	23,851,784	159
VGG16	528 MB	0.713	0.901	138,357,544	**23**

Bold values, for size bold is smallest, for accuracy bold is largest value.

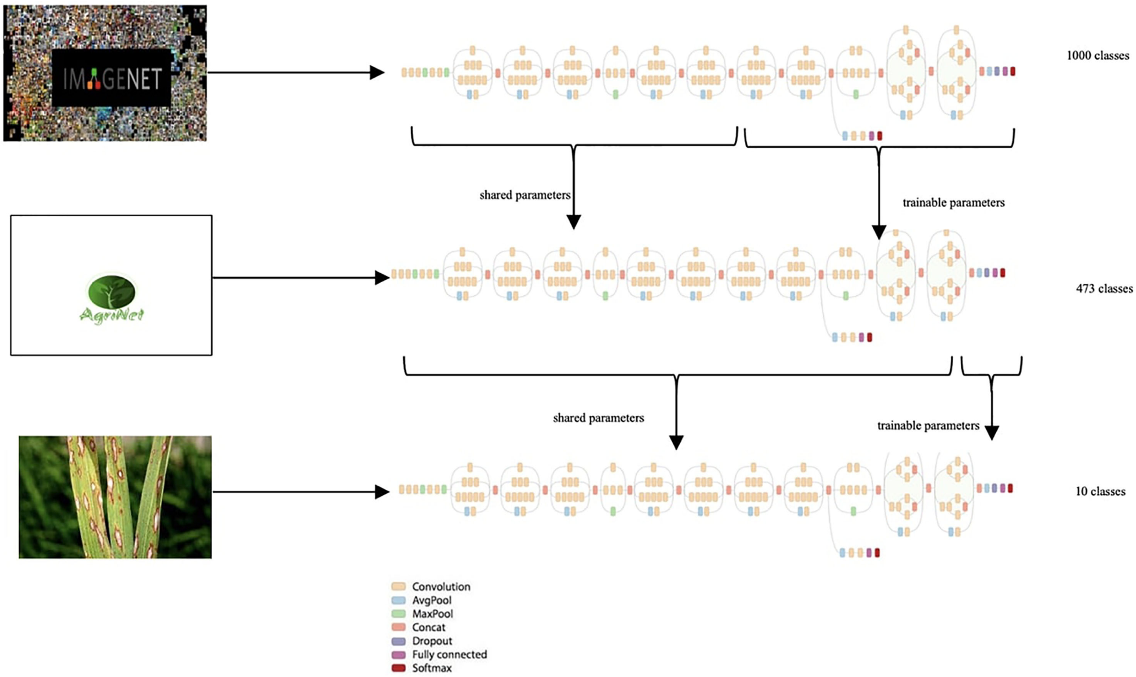


FIGURE 2
Transfer learning.

Dropout

Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel (Battini, 2018). It was proposed by Srivastava et al. to resolve the overfitting problem in large DL models. Moreover, the term refers to dropping out units, which means temporarily removing units from the network, along with all its incoming and outgoing connections (Battini, 2018). In the simplest case, each unit is retained with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks. For the input units, however, the optimal probability of retention is usually closer to 1 than to 0.5 (Battini, 2018).

Evaluation metrics

Evaluation of the proposed models was based on two metrics.

Accuracy

Accuracy represents the number of correctly classified data instances over the total number of data instances.

F1-score

The F1-score is the harmonic mean of precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations while recall is the ratio of correctly predicted positive observations to all observations in the actual class.

Accuracy is the most widely used metric to evaluate the performance of classification models. However, F1-score accompanies accuracy in classification tasks where the dataset is unbalanced.

Results and discussion

Fine tuning AgriNet models

Transfer learning was applied to all AgriNet architectures. For each architecture, multiple experiments were done to propose the most accurate model by changing the number of frozen and trainable layers. The optimizer selected in VGG16 and VGG19 was SGD while in Inception-v3, Xception, and InceptionResNet-v2 Adam optimizer was used. All models were trained on a batch size of 32 (Table 3).

Inception-v3 experiments

The Inception-v3 model constitutes 311 layers. We tested freezing the first 133, 165, 197, 228, 249, and 280 layers. The layers are respectively named mixed4, mixed5, mixed6, mixed7,

TABLE 3 Fine-tuning in agrinet models.

AgriNet architecture	Frozen layers	Trainable layers
Inception-v3	165	146
Xception	116	16
VGG19	17	5
InceptionResNet-v2	400	380
VGG16	15	4

mixed8, and mixed 9. We found that freezing the first 165 layers (mixed5) achieved the highest performance.

Xception experiments: The Xception model has 132 layers. The model was trained while fixing the weights of the first 66, 76, 86, 96, 106, 116, and 126 layers. The layers are respectively named add_5, add_6, add_7, add_8, add_9, add_10, and add_11. We found that fixing the first 116 layers (add_10), achieved the optimum performance.

InceptionResNet-v2 experiments

The InceptionResNet-v2 model constitutes of 780 layers. We tested fixing the first 400, 480, 560, 631, and 711 layers. The layers are respectively named block17_8_mixed, block17_13_mixed, block17_18_mixed, block8_1_mixed, and block8_6_mixed. We found that the first 400 layers (block17_8_mixed) were frozen to achieve the most accurate classification.

VGG16

The VGG16 model has 19 layers. The model was trained while freezing the weights of the first 7, 11, 15, and 19 layers, which are respectively named layer to block2_pool, block3_pool, block4_pool, and block5_pool. We found that fixing the first 15 layers (block4_pool) achieved the highest performance.

VGG19

The VGG19 model constitutes 21 layers. We tested freezing the first 7,12,17, and 21 layers, which are respectively named block2_pool, block3_pool, block4_pool, and block5_pool. We found that the first 17 layers (block4_pool) were frozen to achieve the most accurate classification.

Thus, each of the architectures had its optimum freezing percentage. For Inception-v3 and InceptionResNet-v2, 53% and 51.3% freezing of weights achieved the highest accuracy respectively. On the other hand, for VGG16, VGG19, and Xception, the highest accuracies were achieved when freezing percentages of 78.9,77.3, and 87.9 respectively.

Evaluation of AgriNet models as classification models

Overall networks performance

After training the five architectures, the overall test and per class accuracies and F1-score were reported. Supplementary materials include the total number of images for each class and the per-class test accuracy for each of the five models.

VGG19 surpassed all other models with a test accuracy of 94% and an F1 score of 92%. VGG16 was ranked second, followed by InceptionResNet-v2. However, the Inception-v3 model was the least performing with an average accuracy of 88% and an F1-score of 84% (Table 4).

Another comparison was done for each of the models' sizes, the number of parameters, and Floating-Point Operations (FLOPs). InceptionResNet-v2 had the smallest FLOPs of 375,982,836 operations, followed by Xception with 623,900,414 operations and then VGG19 that reported 718,281,877 operations. Similarly, InceptionResnet-v2 had the smallest number of parameters which is 12,983,584 parameters, followed by VGG16 with 41,888,999 parameters and then by VGG19 with 94,092,935 parameters. For the smallest model size, VGG19 is ranked first with a model size of 159.9MB, followed by VGG16 (180.1MB), and then by InceptionResNet-v2 (980.3MB). Results are displayed in Table 5.

InceptionResNetv2 achieved the best compromise between accuracy, F1-score, FLOPs, number of parameters, and model size. Additionally,

TABLE 4 AgriNet models evaluation.

	Train accuracy	Val accuracy	Test accuracy	F1-score (macro average)
Inception-v3	93.69	88.36	88	84
Xception	94.1	87.73	89	85
VGG19	**95.76**	**93.84**	**94**	**92**
InceptionResNet- v2	91.03	89.82	90	87
VGG16	91.11	91.55	92	90

Bold values, highest accuracy/f1-score.

TABLE 5 Comparing imagenet architectures used.

	FLOPs	Total number of parameters-agrinet	Size-agrinet
VGG16	802,046,505	41,888,999	**159.9 MB**
VGG19	718,281,877	94,092,935	180.1 MB
Xception	623,900,414	124,056,527	1.28 GB
Inception-v3	951,318,693	74,666,183	816.2 MB
InceptionResNet-v2	**375,982,836**	**12,983,584**	980.3 MB

Bold values, highest accuracy/f1-score.

the model has a size of 980.3MB, the lowest FLOPs and number of parameters, it achieved a 90% accuracy and an 87% F1 score.

Categorical evaluation of the models

The AgriNet Dataset consists of three main categories: species, weeds and pests, and diseases. Evaluation per category analysis was performed on each of the architectures proposed. VGG19 outperformed other architectures in species recognition and pests, weeds, and diseases detection tasks (Table 6).

Species classification task

VGG19 achieved the highest accuracies in flowers, leaves, and mushrooms detection. Flowers classes are a combination of the VGG flowers dataset (103 classes) combined with the TensorFlow flowers dataset (5 classes merged with classes of the VGG flowers dataset). The VGG flowers dataset achieved a 70.4% accuracy (Nilsback and Zisserman,). As shown in Table 7, all AgriNet models outperformed the baseline model in the flower classification task. Similarly, leaves are majorly composed of the Leafsnap dataset which achieved an accuracy of 70.8 in (Kumar et al., 2012b). In the mushrooms classification task, the highest accuracy of 75.77% was achieved by VGG19. The low accuracy of mushrooms classification compared to other classification tasks in AgriNet is mainly caused by the different image patterns of mushrooms compared to leaves and flowers constituting all other classes.

TABLE 6 Categorical macro-average test accuracy.

Category	Species	Pests and weeds	Plant diseases
VGG16	87.38	91.56	90.84
VGG19	**91.96**	**94.71**	**92.51**
Xception	84.94	85.55	85.6
Inception-v3	82.15	86.99	87.93
InceptionResNet-v2	84.28	90.23	89.5

Bold values, highest accuracy/f1-score.

Pests and weeds classification task

The five pretrained models were able to achieve high accuracies in classifying pests and weeds as shown in Table 8. For weed images retrieved from the deep weeds dataset, the highest macro average accuracy of 89.9% was achieved by VGG19 while (Rahman et al., 2018) achieved the highest macro average accuracy of 74.93% using ResNet50. Same for weed seedlings, VGG19 achieved the top performance while reaching an accuracy of 98.62%. Moreover, for soybean weeds, all models achieved around 99% accuracy. Finally, in pests classification, Xception had the minimum accuracy of 94.87% and VGG19 had the highest accuracy of 98.62%.

Plant diseases classification task

Plant diseases classes were merged from different datasets. VGG19 was able to classify the largest number of plant diseases most accurately with a macro-average accuracy of 92.51%. VGG16 achieved top-class accuracy in a smaller number of classes than VGG19 and was ranked second with a macro-average accuracy of 90.84%. Sample macro average accuracies are presented in Table 9.

Evaluation of AgriNet models as pretrained models

Evaluation on rice pests and diseases dataset

To evaluate the superiority of the proposed models, transfer learning was applied using ImageNet and AgriNet weights for the five ImageNet architectures on the rice pest and plant diseases dataset (Kour and Arora, 2019). The dataset was split into 70% training, 10% validation, and 20% test sets. This dataset is a collection of 1426 field images of rice pests and diseases

TABLE 7 Macro average test accuracies for species category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Flowers	103	80.63	86.92	86.88	**93.47**	80.63
Leaves	194	86.42	82.65	88.67	**91.02**	86.84
Mushrooms	12	69.83	72.44	70.71	**75.77**	73.7

Bold values, highest accuracy/f1-score.

TABLE 8 Macro average test accuracies for pests and weeds category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Weeds	8	79.92	73.2	86.85	**89.99**	85.61
Seedling	12	76.72	81.88	88.66	**92.72**	86.35
Pests	8	94.95	94.87	95.74	**98.62**	94.9
Soybean	5	99.2	98.82	99.34	**99.43**	**99.43**

Bold values, highest accuracy/f1-score.

TABLE 9 Macro average test accuracies for some species used in plant diseases category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Apple	3	83.06	86.21	91.27	**93.23**	88.98
Bean	3	90.8	88.5	90.03	**93.89**	95.05
Cassava	5	67.77	64.1	71.3	**72.75**	69.29
Cherry	2	95.2	93.53	97.88	**98.07**	97.6
Coffee	5	94.14	92.82	**97.52**	97.26	95.67
Corn	4	93.07	92.23	93	**95.51**	90.68
Cotton	4	95.62	91.38	**97.6**	96.5	95.63
Guava	2	94.13	90.66	95.42	**95.62**	94.33
Grape	4	93.69	93.83	**98.9**	98.38	97.83
Citrus	2	90.3	98.09	93.87	**99.59**	89.57
Mango	2	86.8	91.5	**98.11**	96.64	93.34
Potato	2	87.1	88.28	94.73	**95.20**	85.17
Rice	3	83.33	75	83.33	**87.5**	75
Strawberry	2	95.05	97.97	97.75	**99.54**	99.54
Tomato	12	78.5	73.3	84.95	**87.04**	81.09

Bold values, highest accuracy/f1-score.

collected from paddy fields of Bangladesh Rice Research Institute (BRRI) for 7 months (Table 10).

AgriNet models achieved higher accuracies than ImageNet models. In VGG16, the model achieved a 90% accuracy using AgriNet weights, compared to 83% for ImageNet weights (Figure 3). On the VGG19 side, using AgriNet weights resulted in an 88% accuracy compared to 83% accuracy using ImageNet weights. Although VGG19 achieved the highest accuracy on the AgriNet dataset, VGG16 performed better on the rice pest and plant diseases dataset. This can be caused by specific image features that vary between a dataset and another. Thus, it is recommended to evaluate any agricultural dataset on multiple AgriNet architectures to achieve the best performance possible. It should be noted that the above accuracies on the dataset resulted after freezing the models and only training the dense layers. Further experiments can result in higher accuracies.

TABLE 10 Rice pest and diseases dataset description.

Class	Number of images
False Smut	93
Brown Plant Hopper	71
Bacterial Leaf Blight	138
Neck Blast	286
Stemborer	201
Hispa	73
Sheath Blight and/or Sheath Rot	219
Brown Spot	111

Evaluation on plant diseases of Kashmir dataset

The plant diseases dataset of Kashmir contains 2136 images for eight plant species: Apple, Apricot, Cherry, Cranberry, Grapes, Peach, Pear, and Walnut with a total of 1201 healthy images and 935 diseased images (Table 11) (52). The dataset was split into 70% training, 10% validation, and 20% test set. Similar to the case of the pest and plant diseases dataset, VGG16 achieved the highest accuracy and F1-score for both ImageNet and AgriNet models. All five AgriNet models achieved higher accuracies than ImageNet models. Moreover, the highest AgriNet accuracy reported was 83% compared to 70% in ImageNet on VGG16 as mentioned above (Figure 4).

Conclusion

In this paper, we present AgriNet dataset and AgriNet models, a collection 160k agriculture images dataset and a set

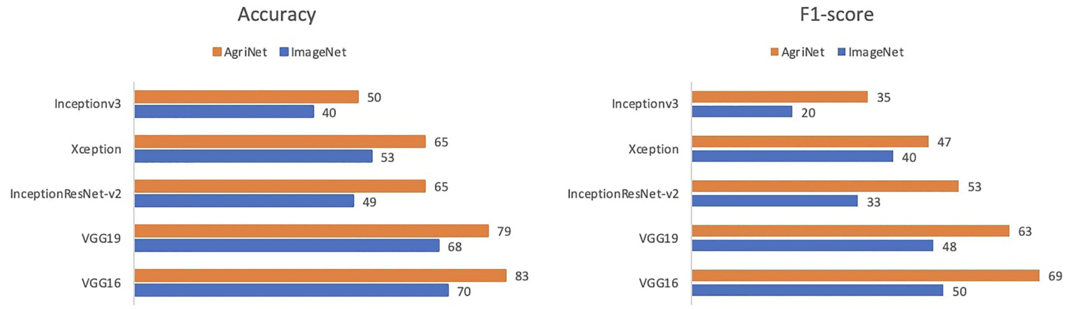


FIGURE 3
Models comparison on rice pests and diseases dataset.

TABLE 11 Plant diseases of kashmir dataset description.

Class	Number of images
Apple Healthy	93
Apple Diseased	100
Apricot Healthy	86
Apricot Diseased	100
Cherry Healthy	82
Cherry Diseased	95
Cranberry Healthy	100
Cranberry Diseased	94
Grapes Healthy	100
Grapes Diseased	9
Peach Healthy	100
Peach Diseased	18
Pear Healthy	100
Pear Diseased	58
Walnut Healthy	93
Walnut Diseased	100

of five agriculture-domain specific pretrained models respectively. VGG architectures achieved the highest accuracy with 94% accuracy for VGG19 and 92% accuracy for VGG16. InceptionResNet-v2 had the best compromise between the model's performance, and the computational cost through the number of trainable parameters, FLOPs, added to the model's size. In addition, the superiority of the proposed models was evaluated by comparing the AgriNet models with the original ImageNet models on two external pest and plant diseases datasets.VGG architectures resulted in best performance in both ImageNet and AgriNet models, where AgriNet surpassed the ImageNet models with accuracy increase of 18.6% and 8.4% using VGG16 for Kashmir dataset and rice dataset respectively.Further advancements to the AgriNet project include training the AgriNet dataset on more recent convolutional neural networks architectures, expanding pretraining to vision transformers, and increasing the dataset size through adding extra datasets or through applying advanced image augmentation techniques. However, adding additional datasets is restricted to the limited number of agricultural

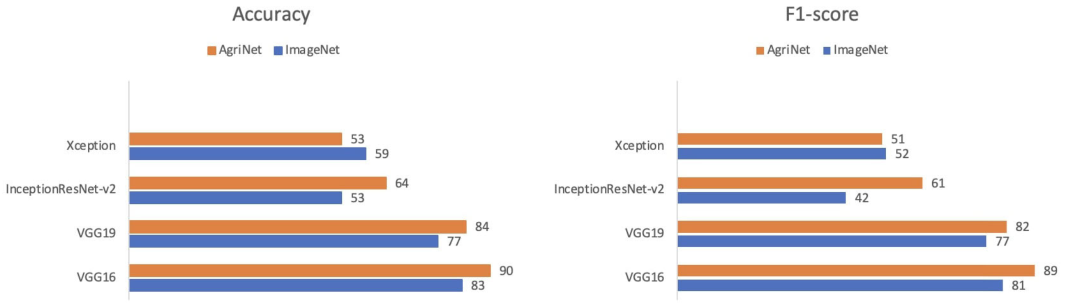


FIGURE 4
Models comparison on plant diseases of kashmir dataset.

public datasets, which urges the research community in retrieving private datasets to public status whenever possible.

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. The models are available here; Dataset will be publicly available on Kaggle. AgriNet Models link: https://drive.google.com/drive/folders/183REzCkgMSI0nlWXb4Y2APdXujSsN_em?usp=sharing.

Author contributions

ZA and MA contributed data collection, methodology, and models evaluation. Both contributed to manuscript revision, read, and approved the submitted version. MA is the advisor of ZA and was following up over all the project tasks. Both authors contributed to the article and approved the submitted version. ZA collected the dataset, proposed hte methodology, trained the models, and evaluated the models, all under MA's supervision. The project is based on MA's idea where she supervised the research including methodology updated and revised the manuscript for several times.

Funding

This research was supported by WEFRAH.

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.2022.992700/full#supplementary-material

References

Akhtar, Z. (2021) *Xception: Deep learning with depth-wise separable convolutions* (OpenGenus IQ: Computing Expertise & Legacy). Available at: https://iq.opengenus.org/xception-model/ (Accessed 27-Aug-2021).

(2016) *Improving inception and image classification in TensorFlow* (Google AI Blog). Available at: https://ai.googleblog.com/2016/08/improving-inception-and-image.html (Accessed 27-Aug-2021).

Battini, D. (2018) *Implementing drop out regularization in neural networks* (Tech). Available at: https://www.tech-quantum.com/implementing-drop-out-regularization-in-neural-networks/ (Accessed 27-Aug-2021).

Cassava disease classification (Kaggle). Available at: https://www.kaggle.com/c/cassava-disease/overview (Accessed 27-Aug-2021).

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1800–1807.

Chouhan, U. P., Singh, A. Kaul, and Jain, S., "A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 700–707. doi: 10.1109/ISCON47742.2019.9036158

D3v, (2020) *Cotton disease dataset* (Kaggle). Available at: https://www.kaggle.com/janmejaybhoi/cotton-disease-dataset (Accessed 27-Aug-2021).

Elhamraoui, Z. (2020) *Inceptionresnetv2 simple introduction* (Medium). Available at: https://medium.com/@zahraelhamraoui1997/inceptionresnetv2-simple-introduction-9a2000edcdb6 (Accessed 27-Aug-2021).

Gandorfer, M., Christa, H., Nadja El, B., Marianne, C., Thomas, A., Helga, F., et al. (2022). "Künstliche intelligenz in der agrar-und ernährungswirtschaft," in *Lecture Notes in Informatics (LNI)*, Bonn (Gesellschaft für Informatik).

Giselsson, T. M., Jørgensen, R., Jensen, P., Dyrmann, M., and Midtiby, H. (2017). A public image database for benchmark of plant seedling classification algorithms. *ArXiv*.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intell. Syst*. doi: 10.1109/MIS.2009.36

Huang, M.-L., and Chang, Y.-H. (2020). Dataset of tomato leaves. *Mendeley Data* 1. doi: 10.17632/ngdgg79rzb.1

Huang, M.-L., and Chuang, T. C. (2020). A database of eight common tomato pest images. *Mendeley Data* 1. doi: 10.17632/s62zm6djd2.1

J, A. P., and Gopal, G. (2019). Data for identification of plant leaf diseases using a 9-layer deep convolutional neural network. (PlantaeK: A leaf database of native plants of Jammu and Kashmir) *Mendeley Data* 1. 76, 323–338. https://doi.org/10.1016/j.compeleceng.2019.04.011.(https://www.sciencedirect.com/science/article/pii/S0045790619300023

Kour, V. P., and Arora, S. (2019). PlantaeK: A leaf database of native plants of jammu and Kashmir. *Mendeley Data* 2. doi: 10.17632/t6j2h22jpx.2

Krohling Renato, A., Esgario Guilherme, J. M., and Ventura José, A. (2019), "BRACOL - A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests", Mendeley Data, V1, doi: 10.17632/yy2k5y8mxg.1

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C., et al. (2012a) "Leafsnap: A computer vision system for automatic plant species identification," in *European Conference on Computer Vision*. Available at: http://leafsnap.com/dataset/.

Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D., Kress, W., Lopez, I., et al. (2012b). "Leafsnap: A computer vision system for automatic plant species identification," in *European Conference on Computer Vision - ECCV*, Vol. 7573. 502–516. doi: 10.1007/978-3-642-33709-3_36

Madsen, S. L., Mathiassen, S. K., Dyrmann, M., Laursen, M. S., Paz, L.-C., and Jørgensen, R. N. (2020). Open plant phenotype database of common weeds in Denmark. *Remote Sens.* 128, 1246. doi: 10.3390/rs12081246

Makerere AI Lab (2020) *Bean disease dataset*. Available at: https://github.com/AI-Lab-Makerere/ibean/ (Accessed 27-Aug-2021).

Marsh (2020). *Rice leaf diseases dataset* (Kaggle). Available at: https://www.kaggle.com/vbookshelf/rice-leaf-diseases (Accessed 27-Aug-2021).

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

Nilsback, M., and Zisserman, A. *102 category flower dataset* (Visual Geometry Group - University of Oxford). Available at: https://www.robots.ox.ac.uk/~vgg/data/flowers/102/ (Accessed 17-Nov-2020).

Nilsback, M.-E., and Zisserman, A.Automated flower classification over a large number of classes in *(2008), Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

Olsen, A., Konovalov, D. A., Philippa, B., et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* 9, 2058. doi: 10.1038/s41598-018-38343-3

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. *"DeepWeedsDataset" scientific reports*. Available at: https://nextcloud.qriscloud.org.au/index.php/s/a3KxPawpqkiorST/download.

Peccia, F. (2018) *Weed detection in soybean crops* (Kaggle). Available at: https://www.kaggle.com/fpeccia/weed-detection-in-soybean-crops (Accessed 27-Aug-2021).

Rachman, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., and Hughes, D. (2017). Deep learning for image-based cassava disease detection. *Front. Plant Sci.* 1852. doi: 10.3389/fpls.2017.01852

Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A., Wasif, A., Jani, M. R., et al. (2018).). identification and recognition of rice diseases and pests using deep convolutional neural networks. *ArXiv*.

Rauf, H. T., Saleem, B. A., Lali, M.I. U., Khan, 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. *Mendeley Data* 2. doi: 10.17632/3f83gxmv57.2

Rumpf, T., Römer, C., Weis, M., Sökefeld, M., Gerhards, R., and Plümer, L. (2012). Sequential support vector machine classification for small-grain weed species discrimination with special regard to cirsium arvense and galium aparine. *Comput. Electron. Agric.* 80, 89–96. doi: 10.1016/j.compag.2011.10.018

Silva, G.M. S. U. E (2021) *Feeding the world in 2050 and beyond – part 1: Productivity challenges*. Available at: https://www.canr.msu.edu/news/feeding-the-world-in-2050-and-beyond-part.

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for Large-scale image recognition. *CoRR*.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020a). "PlantDoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020b) *PlantDoc-dataset* (GitHub). Available at: https://github.com/pratikkayal/PlantDoc-Dataset (Accessed 27-Aug-2021).

Söderkvist, O. (2016) *Swedish Leaf dataset*. Available at: https://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/ (Accessed 17-Nov-2020).

Singh, K. (2021). How to dealing with imbalanced classes in machine learning. *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/ (Accessed 27-Aug-2021).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (56), 1929–1958.

Stanford *CS231n convolutional neural networks for visual recognition*. Available at: https://cs231n.github.io/convolutional-networks/ (Accessed 27-Aug-2021).

Stanford *CS231n convolutional neural networks for visual recognition*. Available at: https://cs231n.github.io/transfer-learning/ (Accessed 27-Aug-2021).

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *AAAI*. doi: 10.1609/aaai.v31i1.11231

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9.

Talaviya, T., Shah, D., Patel, N., Yagnik, H., and Shah, M. (2020). Implementation of artificial intelligence in agriculture for optimization of irrigation and application of pesticides and herbicides. *Artif. Intell. Agric.* 4, 58–73.

Tan, K. C., Liu, Y., Ambrose, B., Tulig, M., and Belongie, S. J. The herbarium challenge 2019 dataset. *ArXiv*.

Teimouri, N., Dyrmann, M., Nielsen, P., Mathiassen, S., Somerville, G., and Jørgensen, R. (2018). Weed growth stage estimator using deep convolutional neural networks. *Sensors* 185, 1580. doi: 10.3390/s18051580

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.* 89. doi: 10.1002/aps3.11390

The TensorFlow Team,Flowers. (2019) Available at: http://download.tensorflow.org/example_images/flower_photos.tgz (Accessed 17-Nov-2020).

U. C. I. M. Learning (2016) *Mushroom classification* (Kaggle). Available at: https://www.kaggle.com/uciml/mushroom-classification (Accessed 27-Aug-2021).

Wäldchen, J., and Mäder, P. (2018). Plant species identification using computer vision techniques: A systematic literature review. *Arch. Comput. Method E* 25, 507–543. doi: 10.1007/s11831-016-9206-z

Yang, S., Zheng, L., Chen, X., Zabawa, L., Zhang, M., and Wang, M. (2022). "Transfer learning from synthetic *In-vitro* soybean pods dataset for *In-situ* segmentation of on-branch soybean pods," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 1665–1674.

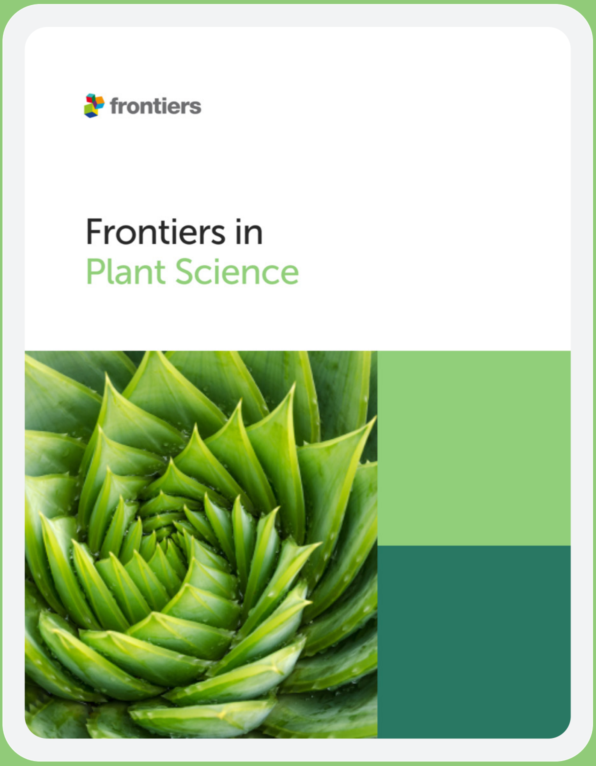
You, K., Long, M., Jordan, M. I., and Wang, J. (2019). Learning stages: Phenomenon, root cause, mechanism hypothesis, and implications. *ArXiv*.

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

frontiers | Research Topics