Citrus Yellow Shoot Disease Detection based on YOLOV5

Shenglong Wang [1, †], Yixin Zhang [2, †, *], and Runqi Li [3, †]

[1] School of Information science and technology, Qingdao University of Science and Technology, Qingdao, China

[2] Department of Engineering Architecture and Information Technology, The University of Queensland, Brisbane, Australia

[3] School of Logistics and E-Business, Henan University of Animal Husbandry and Economy, Henan, China

* Corresponding author email: yixin.zhang3@uqconnect.edu.au

[†]These authors contribute equally to this work

Abstract. Citrus yellow shoot disease (also called Huanglongbing in Chinese) is a devastating disease of citrus. A large number of experiments have shown that citrus yellow branch disease cannot be cured, but can only be prevented and controlled. Traditional detection methods are limited by long time and low coverage, and cannot predict diseases in time. Thanks to the powerful feature representation capabilities of convolutional neural networks, the detection of citrus yellow shoot disease based on deep learning has become the mainstream of current research. In this paper, we transform the recognition task of citrus greening disease into the detection of lesion area, focus on the feasibility of using deep learning association algorithm to identify the symptoms of citrus yellow shoot disease, and evaluate the recognition accuracy. Specifically, we preprocessed the collected citrus plant images to construct the training data set, and built a citrus yellow shoot disease recognition model based on the YOLOv5s algorithm. The results show that the accuracy reaches 91.3% and the recall rate reaches 88.9% after detection by YOLOv5s model. Finally, it is found that this model can accurately find citrus products with citrus yellow shoot disease, so as to timely prevention and control, improve the identification efficiency of citrus yellow shoot disease and reduce the cost of detection.

Keywords: Disease Detection; Citrus; YOLOV5; Object Detection.

1. Introduction

Agriculture provides the food and other vital raw materials necessary for the development of modern societies. However, traditional agriculture mainly relies on extensive cultivation and maintenance work, with low productivity and high labor costs. How to improve agricultural production efficiency and crop yield has always been a topic of concern to many scientists. Thanks to the rapid development of hardware equipment and pattern recognition technology, agricultural automation has gradually become possible. Among them, as an important part of smart agriculture, the automatic identification of crops, especially fruit pests and diseases, can reduce labor costs to a certain extent and improve the cultivation problems encountered in the process of crop growth [1].

As a fruit that has been widely loved by the public, citrus has extremely high economic value; at the same time, it is also extremely susceptible to various diseases during its growth. Citrus yellow shoot is a devastating disease of citrus, which is a bacterial disease spread by psyllids or fruit flies, or the typical yellowing caused by nutrient deficiencies. A large number of experiments have shown that citrus yellow shoot cannot be cured, but can only be prevented and controlled. In this context, how to timely and accurately identify citrus yellow shoot has grown urgent and needs to be resolved.

Object recognition has advanced significantly in the field of computer vision with the growing development of deep learning, particularly convolutional neural network. One of the most significant areas of research in computer vision is target detection, which may be loosely divided into two categories: one-stage target detection and two-stage target detection. The YOLO series is a representation of one of the stage target detection models. We suggest using the yolov5 artificial

intelligence identification approach to address the existing challenge of determining if citrous is infected with citrus yellow shoot illness. This will increase the research effectiveness of the traditional yolo method. Yolov5 is more practical and quicker than the conventional approach. Higher, better suited for recognizing huge objects. Additionally, the algorithm has a stronger feature extraction capability and a greater detection accuracy while detecting large targets as a result of avoiding the drawbacks of the original study approach. The data set used in this study is taken from the preliminary round of the iFLYTEK 2021 AI Algorithm Competition and includes information on citrous illnesses and insect pests. Three labels altogether—citrus yellow shoot, healthy, and various diseases—are used. The training of the entire model is more effective. Through the analysis of citrous leaves, it aims to assist fruit growers in determining whether a citrous tree is affected by citrus yellow shoot.

Focusing on the above motivations, in this paper, we provide citrus yellow shoot detection method based on YOLOV5s, which can determine whether citrus is yellow shoot-positive and suffers from other diseases as well as which citrus varieties are healthy. The accuracy after detection by the YOLOv5s model was 91.3%, and the recall rate was 88.9%. It examines the best model for this target detection as well as whether it can improve accuracy by changing some parameters. Finally, it was discovered that this model accurately identified the citrous goods affected by citrus yellow shoot, allowing for their prompt resolution and a final boost in yield.

The structure of the essay is as follows. In Section 2, we go into great detail about our strategy. In section 3, we examine the detection outcomes from numerous experiments using our approach. Finally, we discuss the remaining research issues in the field of crop disease identification and look forward to its future development directions.

2. Method

Yolov5 contains yolov5s, yolov5m, yolov5l, yolov5x, and yolov53-spp. The Yolov5s network is the Yolov5 series network with the least depth and feature map width [2]. We select the smallest yolov5s model to employ in this situation because the size of our data is pretty little. The network structure of yolov5s is depicted in the Figure 1 below [3].

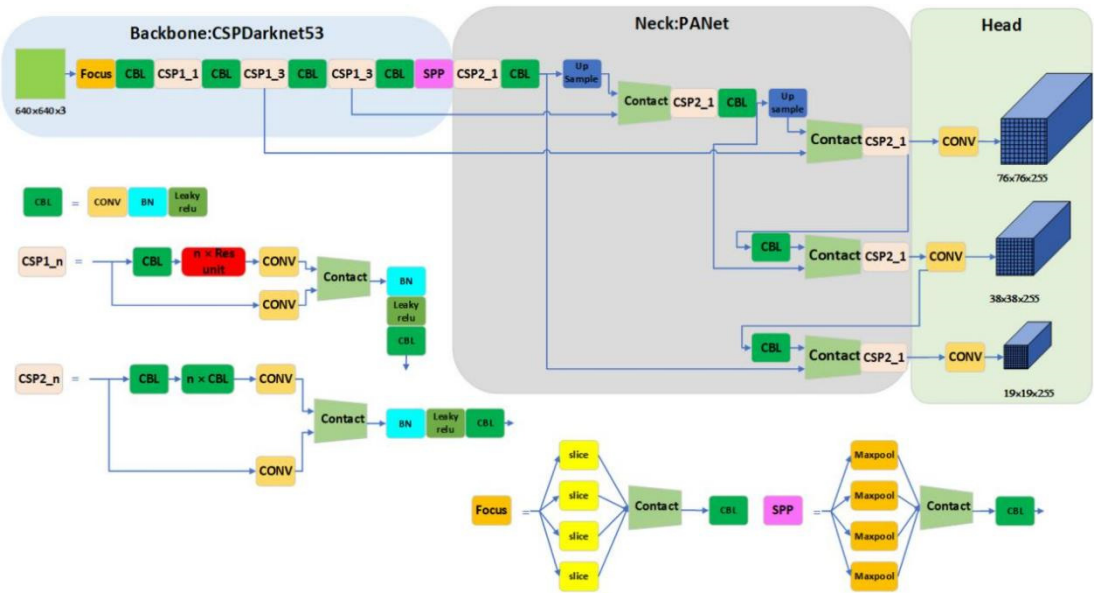


Fig 1. The structure of yolov5s network

As shown in Figure 1, it is clear that the YOLOv5 network is divided into three sections: the output for object detection, the neck for feature fusion, and the backbone for feature extraction. The number of categories, anchor setting, backbone network structure, and head network structure are some of the factors that go into the model's construction. Three categories—healthy citrous, citrous infected with

citrus yellow shoot, and citrous infected with viruses other than citrus yellow shoot—are covered in this article. There are a total of three lines for the anchor setting; the first line appears on the smallest feature map, the second line appears on the second-biggest feature map, and the third line appears on the largest feature map. The three distinct resolutions may recognize targets of various sizes by using tiny values to detect huge objects. Then comes the backbone, which has a total of four phases in this article. Focus structure, Bottleneck layer, BottleneckCSP, and SPP structure make up the backbone module (Spatial Pyramid Pool).

Focus is used in the initial phase. Prior to entering the backbone, the Focus module in version 5 splits the image. The specific technique, which is comparable to neighboring downsampling, involves obtaining a value for each additional pixel in a picture. There is no information lost despite the fact that the four photographs I received are complementary and roughly the same length. Thus, the input channel is stretched by 4 times, the W and H information is focused in the channel space, and the spliced images are in relation to the 12-channel RGB mode has replaced the previous three-channel RGB mode. A double downsampling feature map is obtained without information loss after the newly obtained image has undergone convolution processing.

The bottleneck layer in CSP is made up of stages 2, 3, and 4. A bottleneck is made up of two 1x1 convolutional layers sandwiching a 3×3 convolutional layer. The 1×1 convolutional layers lower dimensionality before restoring it, resulting in a 3×3 convolution with fewer parameters and a faster computation time. For instance, the 256-dimensional channel is initially reduced to 64 dimensions using 1×1 convolution, and then it is restored through 1×1 convolution at the conclusion. The total number of parameters used is 69632, which is equal to $1 \times 1 \times 256 \times 64 + 3 \times 3 \times 64 \times 64 + 1 \times 1 \times 64 \times 256$. Since there is a shortcut for C3 in the backbone but not in the neck, the CSP structure used in the backbone is C3, and the C3 layer in the backbone is represented by CSP1 x while the C3 in the neck is represented by CSP2 x.

The objective multi-scale problem can be partially resolved by the SPP structure by passing the input through a number of parallel MaxPools of various sizes. The backbone's parameters stand for several interpretations. For instance, the input from the first n levels is represented by -n in from, where -1 is the input from the layer before that. Number furthermore denotes the quantity of network modules, e.g., [-1, 3, C3, [128]] denotes the presence of 3 C3 modules. Additionally, the name of numerous network modules is represented by the model parameter. Last but not least, the parameter args lists the arguments given to several modules, including [ch out, kernel, stride, padding, groups]. The head network component is the next object to be employed in the model. The three Detect detectors, or the method of doing target detection on feature maps of various scales using grid-based anchors, make up the bulk of the head. CSPDarkNet53, which has four feature maps of various sizes, serves as the backbone network [1]. The neck network's job is to combine feature maps in order to gain more contextual information and minimize loss. The fusion process uses the feature pyramid structure of the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) [4]. Strong semantic features are transferred from top feature maps to lower feature maps by the FPN structure. In the meantime, the PAN structure moves potent localization features from lower to higher feature maps. The output network section performs item detection and classification using these updated feature maps.

The focus module in this design slices and concatenates pictures to improve feature extraction during downsampling. Convolution, normalization, and a Leaky Rectified Linear Unit (ReLu) activation function make up the CBL module. In YOLOv5s, there are two different Cross-Stage Part (CSP) networks. One is for the neck, the other is for the backbone. Cross-layer connections between the front and rear layers of the network allow CSP networks to boost inference speed while preserving accuracy and a smaller model size [5]. The two different CSP network types have a comparable structure. However, the backbone's CSP network still contains one to three residual units, but a CBL module has taken the position of the CSP network's neck residual unit. Additionally, by concatenating features, the Spatial Pyramid Pooling (SPP) module performs max pooling with various kernel sizes

[6]. Tensor concatenation operations are represented by the Concat module. The flowchart for the detection of citrus leaves with yellow shoot is shown in Figure 2.

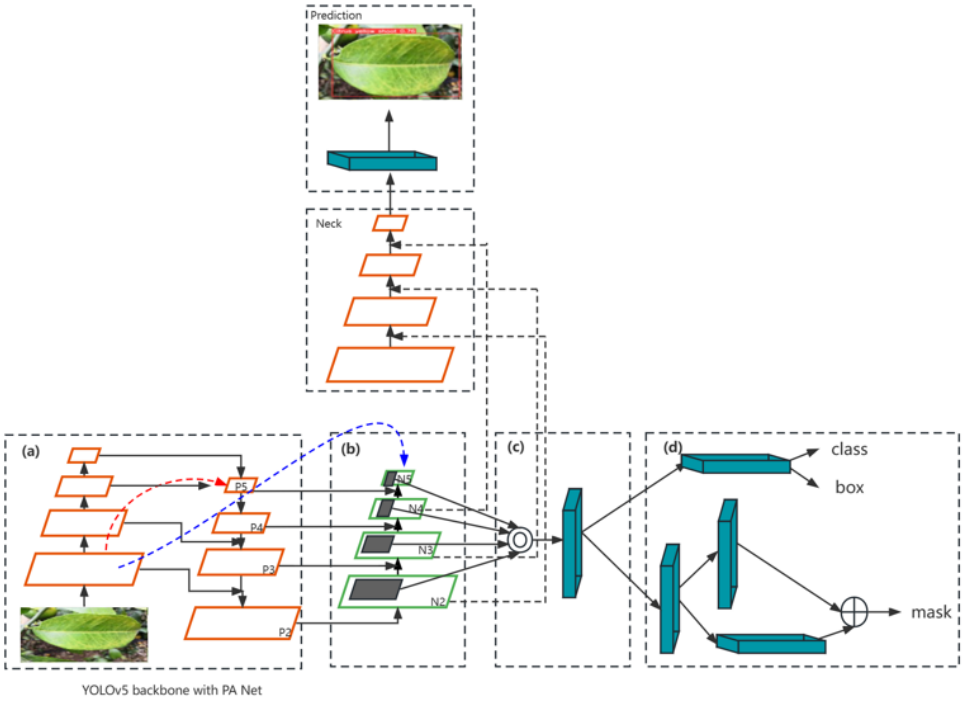


Fig 2. The whole pipeline of our citrus yellow shoot detection network

3. Experiments and Performance Analysis

3.1 Original Data

This object detection model uses 522 labeled citrus disease images from the training set provided by the preliminary training set of the Orange Disease and Insect Pest Identification Challenge, the Question 7 of iFLYTEK 2021 AI Developer Algorithm Contest. The num_classes of the dataset are limited to 3, including “Citrus yellow shoot”, “healthy” and “other diseases”. “Citrus yellow shoot” represents that the citrus leaves in the picture are suffering from the disease named yellow shoot and “healthy” represents that the citrus leaves in the picture are healthy, and “other disease” represents that they are suffering from diseases other than citrus yellow shoot, such as yellowing due to nutrient deficiency. The specific image label is shown in Figure 3.

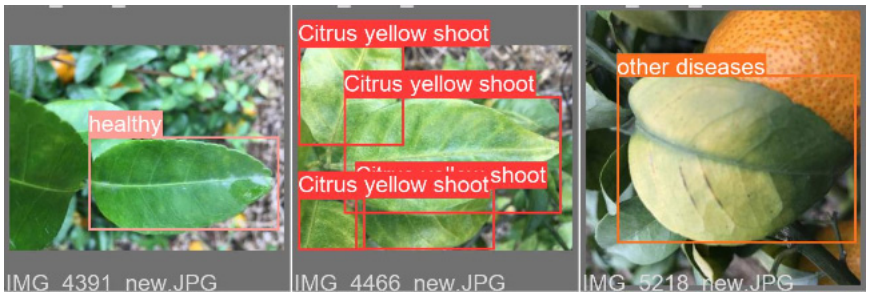


Fig 3. Samples of training images and corresponding labels in the training set

The data set is in Yolo format, and the name of the data set label file is consistent with that of the image file. The division of the data set is shown in Table 1.

Table 1. The details of different datasets

Dataset	Images number
Total	522
Train	448
Test	74

3.2 Experiment Environment

The experiment environment chooses the Ubuntu 18.04 operating system and the Python framework. The graphics card GeForce RTX 3080 is used for image processing. The specific experimental configuration is shown in Table 2.

Table 2. Configurations of the experiment environment

Parameter	Configuration
CPU	12 cores Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz
GPU	GeForce RTX 3080
Operating system	Ubuntu 18.04
Framework	Python 3.8
Acceleration environment	Cuda 11.1

3.3 Experiment Settings

In the training phase of the network model, the batch_size of the model is set to 8, the total number of iterations is 224, the attenuation coefficient is 0.0005, the initial learning rate is set to 0.01, and the model converges after about 200 iterations.

3.4 Experimental Evaluation Index

Mean Average Precision (mAP) is the average precision of all classes divided by the sum of the average precision of all classes in the dataset, which is an evaluation index to intuitively reflect whether the object detection model has good performance. When the model is iterated 30 times, the mean average precision is close to 1. The precision and recall of the model can be calculated according to the overlap rate IoU (the default initial value is 0.5), TP represent the number of samples which are correctly detected, FP means the number of positive samples which are misclassified, and FN shows the number of negative samples which are misclassified. According to the preset IoU and the calculated recall and precision, the corresponding evaluation accuracy can be intuitively reflected by the P-R curve. Finally, the final evaluation index mAP was obtained according to the average evaluation accuracy of each category. F1-score is the value of comprehensive evaluation of precision and recall, the larger the value of F1-score, the better the comprehensive performance of the algorithm. All of these indexs can be calculated through the following formulation.

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$F_{score} = \frac{2*Precision*Recall}{Precision+Recall} \tag{3}$$

3.5 Performance Analysis

3.5.1 Analysis of Training Loss

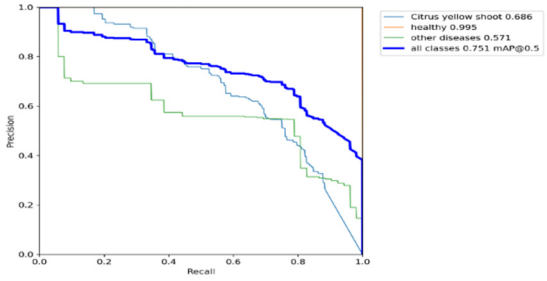
Aiming to verifying that the training of the model has converged, we first visualize the changes in the loss of each part of the model, and the results are shown in Figure 4. When Yolov5 is used to train the model, there are three loss functions, namely cls_loss, box_loss and obj_loss. The loss function changes with the convergence of the model along with the training curve of the last 100 epochs. The classification loss and localization loss decreased significantly during the training process, and began to oscillate slightly after reaching the lowest value at 90 epoch.



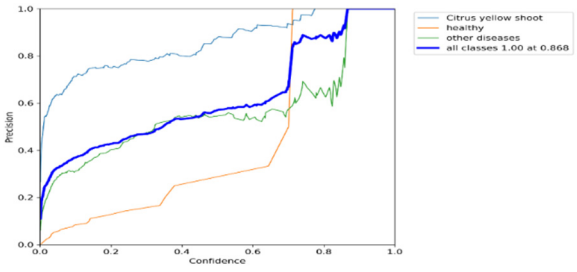
Fig 4. Model loss under different training stages

3.5.2 Detection Results Analysis

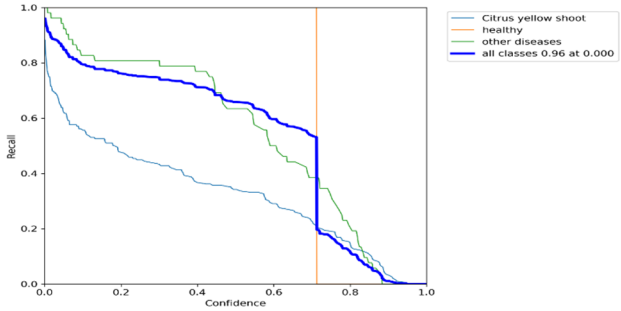
To further test the detection robustness of the model, we do a series of experiments to calculate the Precision, Recall and F1-score of the model on the test set, which can be visually reflected in Figure 5. A higher Precision represents a great true positive samples proportion in the test set that are determined by the classifier. The higher the precision, the higher the prediction accuracy of the model,. Accuracy When the model is iterated to 180 times, the value can be stabilized above 0.8. Recall is the probability that the correct classes in the sample are predicted correctly, and the higher the recall, the higher the accuracy of the algorithm identification. In the process of iterative training, the recall rate decreases with the increase of precision, and the change of the recall rate with the confidence of the prediction result is shown in Figure 5(c).



(a) Precision-Recall curve



(b) Precision change with different confidence settings



(c) Recall change with different confidence settings

Fig 5. Model performance with different settings

3.5.3 Performance Comparison

Compared with the constrained multi-objective optimization algorithm used in the domestic mainstream detection of citrus yellow shoot, the YOLOv5 single-stage object detection algorithm has its own advantages. Due to space reasons, YOLOv5 is only listed here for comparison with some algorithms such as PSO-MOFSA, FSMEA and CMOA-CHDIA [7-9]. It is trained based on the same data set. When the model converges, three performance indicators such as accuracy, recall rate and F1-score are selected for comparison, and the results are shown in Table 3.

Table 3. Performance comparison with other methods

	YOLOv5	PSO-MOFSA	CMOA-CHDIA	FSMEA
Precision	0.913	0.869	0.906	0.841
Recall	0.889	0.833	0.858	0.822
F1-score	0.901	0.861	0.881	0.831

Using the same data set for training and recording the performance evaluation indicators [10], it can be concluded that YOLOv5 has the highest recognition accuracy, indicating that the YOLOv5 algorithm has a greater possibility to accurately determine whether the citrus in the target picture has yellow shoot disease in this experiment. The recall rate of YOLOv5 is also the highest, which indicates that the YOLOv5 algorithm can better identify each object in this experiment. The F1-score of YOLOv5 is also far superior to all other algorithms. As the harmonic mean of Precision and Recall, the F1-score reflects that the YOLOv5 algorithm has the best recognition effect in this scene. The YOLOv5 algorithm can quickly perform effective reasoning of batch images, videos and even web camera interfaces, so the YOLOv5 algorithm can greatly improve the efficiency of citrus farmers to exclude citrus greening disease.

3.5.4 Performance under Different Scenes

In order to verify whether the YOLOv5 algorithm can still maintain a high detection accuracy for the model when multiple leaves are stacked together and there is inevitable noise in the image caused by occlusion between leaves, the single leaf image and multiple stacked leaf images in the data set are classified into two different data sets for verification. The single leaf examples and multi-leaf examples are shown in Figure 6. The verification results obtained are shown in Table 4.



(a) single leaf examples



(b) multi-leaf examples

Fig 6. Leafs under different scenes

Table 4. Detection results for different scenes

Scenes	Class	Precision	Recall	mAP
single leaf	All	0.916	0.886	0.939
	Citrus yellow shoot	0.967	0.736	0.889
	healthy	0.964	0.932	0.985
	other disease	0.817	0.991	0.943
multi-leafs	All	0.910	0.890	0.936
	Citrus yellow shoot	0.887	0.906	0.947
	healthy	0.924	0.980	0.986
	other disease	0.920	0.783	0.876

It can be seen that the YOLOv5 algorithm can complete high accuracy recognition, whether it is for the detection of a single leaf or the detection of multiple leaf stacks with noise. The total precision and recall rate of the single leaf models and the multi-leaf models are not significantly different, but the detection precision of the single-leaf models is higher than that of the multi-leaf models, and the recall rate is lower than that of the multi-leaf models in the presence of citrus yellow shoot. When there are other diseases, the detection precision of the single-leaf models is a little bit lower than that of the multi-leaf models, and the recall rate is higher than that of the multi-leaf models. Considering that yellow dragon disease requires a detailed analysis of the yellowing position and foliar characteristics of the leaves, while the judgment of other diseases is to exclude the orange infected with yellow dragon disease through the leaf characteristics, the fluctuation of precision and recall rate is within a controllable range.

4. Discussion

This paper first introduces the research background and status of target detection and recognition algorithm, thus leads to the research topic of this paper, and then introduces the method and experimental process we use in detail. This project first completed the construction of PyTorch neural network framework for target detection through anaconda environment manager, and used YOLOv5 algorithm to complete subsequent target recognition training.

At present, the research on citrus yellow shoot disease is mostly based on traditional machine learning algorithms, and the research on deep learning is in the initial stage. It is of great value and necessity to study the symptoms of citrus yellow shoot disease systematically and deeply based on deep learning. This study preliminarily explored the application of deep learning in constructing symptom recognition of citrus yellow shoot disease. Taking PyTorch open-source learning framework as the operating platform, a recognition system with better recognition performance was successfully constructed based on a large number of high-quality citrus yellow shoot disease and health pictures [11]. The YOLOv5 algorithm can not only identify citrus yellow shoot disease more accurately, but also rapidly carry out effective reasoning of batch processing images, videos and even webcam interface. Therefore, the citrus yellow shoot disease detection program based on YOLOv5 can be embedded into the robot arm chip with corresponding software system. Assist citrus fruit growers to eliminate citrus greening more quickly and effortlessly. By constructing a citrus yellow shoot disease recognition model and using the leaf image based on network resources to carry out the algorithm training test, the accuracy of the algorithm training results is about 91.3% whether detecting single leaf or multi-leaf stack, indicating that the deep learning-based citrus yellow shoot disease recognition model is stable and available, and can initially realize the disease recognition and diagnosis.

In the process of data training, we found that the problem of curve fluctuation in our model was caused by the size of the data set. The subsequent research and application still need to pay attention to the optimization of the recognition model and the training scale of the data set, expand the data set of citrus yellow shoot disease, and try to collect the multi-state citrus yellow shoot disease images under the complex background with different light, different humidity. The curve fluctuation was

reduced, and the feature extraction ability was strengthened in view of the disease weakening caused by the decrease of pathogen content.

Compared with traditional machine learning, deep learning combined with transfer learning has incomparable advantages. For example, transfer learning training model can be used only if necessary, programming knowledge is mastered instead of consuming a large amount of manpower and time resources to train the model from scratch, which lowers the technical threshold for related research.

The deployment of PyTorch model in this paper only realizes the deployment of embedded end and mobile end, and fails to implement real-time detection by embedding the model into the manipulator chip. In the future, more powerful models can be trained by extending fruit and vegetable disease categories and deployed into robotic arm chips for large-scale expansion.

5. Conclusion

In summary, the results of this study indicate that deep learning is a powerful research method for symptom recognition of citrus yellow shoot disease, and the combination of symptom classification, training set structure adjustment and other optimization can achieve high recognition effect. Combined with deep learning, it can be used as a new tool for citrus disease image research, providing new ideas and theoretical basis for promoting the subsequent early diagnosis, automatic recognition and symptom classification of citrus yellow shoot disease. Using deep learning methods to identify and study citrus yellow shoot disease can, on the one hand, improve the efficiency of disease identification, achieve early identification and early response in the process of citrus growth monitoring and warning, on the other hand, greatly reduce the cost of citrus yellow shoot disease detection. It is expected to be applied to more research objects and better serve the development of smart agriculture.

References

[1] Gondchawar, N., & Kawitkar, R. S. (2016). IoT based smart agriculture. International Journal of advanced research in Computer and Communication Engineering, 5(6), 838-842.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934 (2020).

[3] Han Shen, Zhiwei Dong, Yongji Yan, Rongwei Fan, Yugang Jiang, Zhaodong Chen, and Deying Chen, "Building roof extraction from ASTIL echo images applying OSA-YOLOv5s," Appl. Opt. 61, 2923-2928 (2022).

[4] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition (2018), pp. 8759–8768.

[5] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: a new backbone that can enhance learning capability of CNN,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020), pp. 390–391.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell. 37, 1904–1916 (2015). [CrossRef].

[7] Wang H, Li K S, CAI T, Wang W X, Dong C K. Citrus greening Dragon disease recognition algorithm based on constrained multi-objective optimization algorithm [J/OL]. Jiangsu Agricultural Sciences :1-9[2022-12-21]. http://kns.cnki.net/kcms/detail/32.1214.S.20221201.2006.002.html.

[8] Liu Yang, Feng Quan, Wang Shuzhi. Plant disease identification method based on lightweight CNN and mobile application[J].Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2019, 35(17): 194-204.

[9] Saleem Muhammad Hammad; Potgieter Johan; Arif Khalid Mahmood Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers [J] Plants Volume 9, Issue 10. 2020. PP 1319-1319.

[10] LU Jianqiang, LIN Jiahan, HUANG Zhongqiang, et al. Identification of citrus fruit infected with Huanglongbing based on Mixup algorithm and convolutional neural network[J]. Journal of South China Agricultural University, 2021, 42(3): 94-101.

[11] DAI Zehan, ZHENG Zheng, HUANG Lishu, et al. Recognition of Huanglongbing symptom based on d eep convolutional neural network[J]. Journal of South China Agricultural University, 2020, 41(4): 111-119.