Hindawi
Scientific Programming
Volume 2020, Article ID 8859237, 13 pages
https://doi.org/10.1155/2020/8859237

Research Article

Detecting Citrus in Orchard Environment by Using Improved YOLOv4

Wenkang Chen,[1] Shenglian Lu,[1,2] Binghao Liu,[3] Guo Li,[1] and Tingting Qian[4]

[1]*College of Computer Science & Information Engineering, Guangxi Normal University, Guilin 541004, China*
[2]*Guangxi Key Lab of Multisource Information Mining & Security, Guilin 541004, China*
[3]*Guangxi Academy of Specialty Crops/Guangxi Citrus Breeding and Cultivation Engineering Technology Center, Guilin 541004, China*
[4]*Agricultural Information Institutes of Science and Technology, Shanghai Academy of Agriculture Sciences, Shanghai 201403, China*

Correspondence should be addressed to Shenglian Lu; lsl@gxnu.edu.cn and Tingting Qian; qiantingting@saas.sh.cn

Received 22 July 2020; Revised 24 October 2020; Accepted 8 November 2020; Published 25 November 2020

Academic Editor: Jiwei Huang

Real-time detection of fruits in orchard environments is one of crucial techniques for many precision agriculture applications, including yield estimation and automatic harvesting. Due to the complex conditions, such as different growth periods and occlusion among leaves and fruits, detecting fruits in natural environments is a considerable challenge. A rapid citrus recognition method by improving the state-of-the-art You Only Look Once version 4 (YOLOv4) detector is proposed in this paper. Kinect V2 camera was used to collect RGB images of citrus trees. The Canopy algorithm and the K-Means++ algorithm were then used to automatically select the number and size of the prior frames from these RGB images. An improved YOLOv4 network structure was proposed to better detect smaller citrus under complex backgrounds. Finally, the trained network model was used for sparse training, pruning unimportant channels or network layers in the network, and fine-tuning the parameters of the pruned model to restore some of the recognition accuracy. The experimental results show that the improved YOLOv4 detector works well for detecting different growth periods of citrus in a natural environment, with an average increase in accuracy of 3.15% (from 92.89% to 96.04%). This result is superior to the original YOLOv4, YOLOv3, and Faster R-CNN. The average detection time of this model is 0.06 s per frame at 1920 × 1080 resolution. The proposed method is suitable for the rapid detection of the type and location of citrus in natural environments and can be applied to the application of citrus picking and yield evaluation in actual orchards.

1. Introduction

Citrus is one of the most important fruits in the world. Citrus is also the fruit with the largest cultivated area, the highest yield and the largest consumption in China [1]. However, China's citrus production has developed slowly, with dense human labour, low productivity, and low efficiency. In recent years, with the continuous improvement in information technology, many innovative techniques for machine vision and artificial intelligence have been tested and used in agricultural production and management with an aim of improving the automation. However, there are still many challenging bottlenecks in practical applications scenarios, in which computer vision recognition could be one of the crucial technologies that affects the actual effects of an agricultural automation system. Therefore, designing a low-cost machine vision system with strong operability for the real-time identification of fruits in different growth stages under natural orchard environments is of great significance for mechanization in the fruit industry.

In recent years, because of advances in technology such as computers, cameras, and image processing, computer vision technology has been specifically applied in agriculture and other corresponding fields [2–5]. Since then, people have developed a series of methods based on fruit image calculation to judge fruit yield [6]. Image pixels are more sensitive to changes in illumination under unstructured lighting conditions, so using the above method reduces the accuracy of fruit detection [7]. For fruits of different shapes, the researchers used a method based on partial shape matching to detect [8]. The use of artificial intelligence methods, especially artificial neural networks (ANNs), helps with the yield estimation of fruit testing. For example, artificial neural networks have been used to successfully identify citrus fruits [2] and tomatoes [9] as well as estimate their numbers. These technologies can assist farmer's management and help them work better.

At present, researchers from around the world are working vigorously to develop automatic picking robots [10]. Compared with traditional artificial intelligence methods, deep learning technology is directly driven by data and its self-learning characteristics used to express that relationships can solve a series of problems that cannot be solved by traditional methods [11]. There are many methods for real-time fruit recognition using deep learning, and they have been successfully used in the agricultural field [12, 13]. Among them, the convolutional neural network framework represented by Faster-RCNN and YOLO is undoubtedly the most widely used in recent years. Koirala et al. [14] considered the impact of light conditions in an orchard on fruit recognition. Furthermore, they used multifunction vehicles equipped with RGB digital cameras and lighting equipment to collect mango fruit images at sufficient light and at night and redesigned the YOLO model. A new framework, "MangoYOLO," can be used to identify mangos well during the day. For the night environment, Xiong et al. [15] proposed the " Des-YOLOv3 network," which is suitable for the recognition of mature citrus under the more complex night environment; it has stronger robustness and higher detection accuracy, with an average accuracy of 90.75% and detection speed of 53f/s. Under the interference of branches and leaves, fruit recognition is challenging; in response to this problem, Hanwen Kang et al. [16] used visual sensors to detect and segment apples and branches in an orchard environment in real time. They developed a lightweight backbone network model based on the remaining network architecture, and the detection accuracy values for apples and branches were 86.5% and 75.7%, respectively. The researchers also studied the ability of the equipment to collect fruit images. For 3D perception and reconstruction, Mingyou Chen et al. [17] used adaptive multivision technology to enhance the three-dimensional perception of banana central rootstock in orchard.

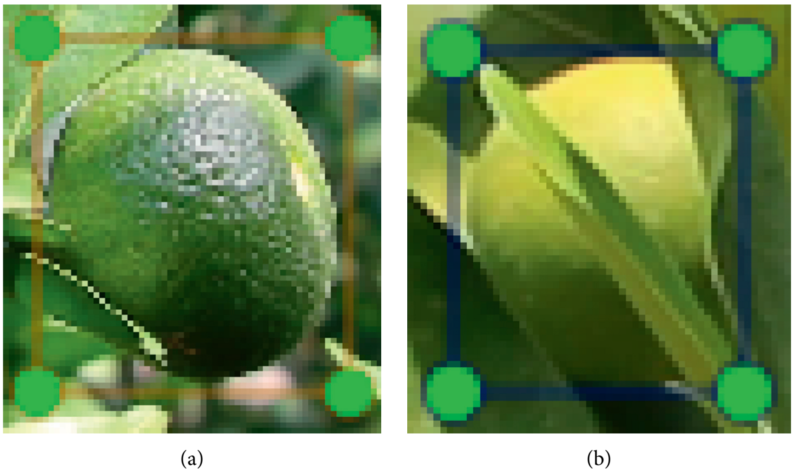
The traditional method is not suitable for identifying citrus in different growth periods in a complex and changeable environment, and the deep learning method has a balance between accuracy and real-time. To solve these problems, the experiments in this paper use an improved YOLOv4 detector, with comparison to Faster-RCNN, YOLOv3, and YOLOv4. The four deep learning models for the real-time identification of four different species of citrus at different growth stages are compared under different experimental conditions, laying the foundation for the further positioning of citrus three-dimensional space.

2. Dataset

2.1. Citrus Image Collection. In this study, image acquisition was conducted using the Kinect V2 depth camera by Microsoft which was used for image acquisition and has a shooting distance between 2 metres and 3 metres. The camera can obtain RGB and depth image data at the same time. The Kinect V2 has a 1920 × 1080 resolution RGB camera, a depth sensor (including an infrared CMOS camera and an infrared emitter), and a microphone array. The camera generates images with a 512 × 424 resolution depth at a rate of 30 frames per second, reconstructs the surrounding environment in real time, and has a shooting distance between 1 metre and 1.5 metres. Two varieties of kumquats and Nanfeng tangerines were selected at the Citrus Experiment Base of the Guangxi Special Crop Research Institute. Two varieties of fertile orange and tangerine were selected in the citrus planting base of Lingui City, Guilin, China. In this study, we collected images of citrus fruit from approximately 1.5 cm in diameter (spring) to ripeness (autumn) in 2019. Images were acquired twice a week at 9 : 00 am, 11 : 00 am, 2 : 00 pm, and 4 : 00 pm, respectively. During the image acquisition process, the shooting direction of the Kinect camera and sunlight illumination directions were parallel to simulate headlights and other backlit situations. Images were also gathered in cloudy conditions to simulate scattered lighting. Because the camera's angle of view would affect the recognition performance, images were collected from multiple angles during the image acquisition process. During the growth and maturity stages of the four sample groups, there were 500 pictures collected for kumquats, 450 pictures collected for Nanfeng tangerines, 400 pictures collected for fertile orange, and 400 pictures collected for tangerine. Among the citrus fruits collected in this paper, the shapes of the fruits in the growing period are mostly spherical, small, and with smooth and green peel surface; the fruits in the mature period are spherical and flat, with smooth or rough peel surface, and the colors of the fruits are mostly yellow and some are still green.

2.2. Image Annotation. The image annotation process included annotating the collected images, selecting the citrus in the frame, and providing training data sets for subsequent recognition model training. The image annotation tool uses LabelImg. When the citrus fruits were labelled in LabelImg, not only the location of the citrus but also the classification of the sample had to be marked. The degree to which each fruit was blocked by leaves or branches was also determined and the degree of overlap between fruits. Two different labels were used: fruits that were more than 50% blocked and those not blocked more than 50%; fruits with blocking rate over 80% are not labelled and the effect is shown in Figure 1.

2.3. Image Data Augmentation. Data augmentation can increase the richness of the experimental data set, process the collected images in terms of colour, brightness, rotation, and image definition, and expand the data set [18] to be more complete. In this study, the Augmentor and imgaug


(a)

(b)

Figure 1: Image annotation: (a) occlusion does not exceed 50%; (b) occlusion exceeds 50%.

image data enhancement libraries were used to amplify the dataset. At the same time, the “keypoint” and “bounding box” parameters were transformed accordingly. Based on the interference factors in the natural environment, the augmentation techniques used in this experiment included rotating the original images (Figures 2(a) and 2(b)), adjusting the image colour (Figure 2(c)), adding noise (Figure 2(d)), and so on.

2.4. Dataset Production. To compare the performance of different algorithms, the image format in the training needs to be adapted to the format requirements of different algorithms. The directory structure of the dataset used in the experiment was generated similar to the directory structure of the PASCAL VOC dataset, and the recognition algorithm parsed the PASCAL VOC annotation as the required format type.

To facilitate training, the input image size required adjustment to a specific resolution. The Faster-RCNN algorithm adjusts the height or width of the input image to 600 pixels while keeping the image aspect ratio unchanged. YOLOv3 adjusts the size of the input image to 416 × 416 pixels, and YOLOv4 adjusts the size of the input image to 608 × 608 pixels. Of course, the network resolution can be increased to accept larger inputs images, but the consequence may be a computer memory leak or increased computing requirements.

We divided the processed data set into a training set, a test set, a verification set, and a training verification set, which accounted for 80%, 10%, 10%, and 90% of the data (sum of the training set and verification set), respectively. Description of citrus dataset is shown in Table 1.

3. Methods

3.1. Faster-RCNN Algorithm. Faster-RCNN is one of the most commonly used deep learning algorithms in recent years. Structurally, Faster-RCNN can be regarded as an RPN area generation network and Fast-RCNN detection of the combination of networks. When Fast-RCNN is integrated into the RPN area generation network, it can merge the target candidate area acquisition, deep feature selection, target recognition, and detection processes in the deep learning network [19]. Previous studies have shown that, compared with R-CNN and Fast-RCNN, Faster-RCNN has a shorter detection running time, and, compared to the YOLO and SSD algorithms, it has better performance in terms of robustness [20].

3.2. YOLOv3 and YOLOv4 Algorithm. The YOLOv3 algorithm is evolved from the YOLO and YOLOv2 algorithms. Compared with the Faster-RCNN network, the YOLO network transforms the detection problem into a regression problem. YOLOv3 does not need to suggest regions, and it directly generates the bounding box coordinates and probability of each class through regression, which greatly improves the detection speed.

The YOLO detection model is shown in Figure 3. The model divides each image in the dataset into an $S \times S$ grid. If the centre of the recognition target is in the grid, the grid is responsible for detecting the target. Each grid predicts the boundary box and its confidence score, as well as the category C conditional probability. The definition of confidence is as follows:

$$C_{IJ} = \Pr \times (\text{Object}) \times \text{IOU}_{\text{predtruth}}. \quad (1)$$

Among them, C_{IJ} represents the confidence of the j^{th} bounding box of the i^{th} grid cell. $\text{IOU}_{\text{predtruth}}$ is used to indicate the coincidence between the reference and the predicted bounding box. The confidence reflects whether the grid contains the detected objects and the accuracy when predicting whether the bounding box contains objects. When multiple bounding boxes detect the same target, YOLO uses the nonmaximum suppression (NMS) method to select the best bounding box [21].

The YOLOv4 algorithm was proposed by Bochkovskiy et al. [22], who combined weighted residual connections (WRC), cross-stage partial connections (CSP), and cross-small batch connections (Cross mini-Batch Normalization

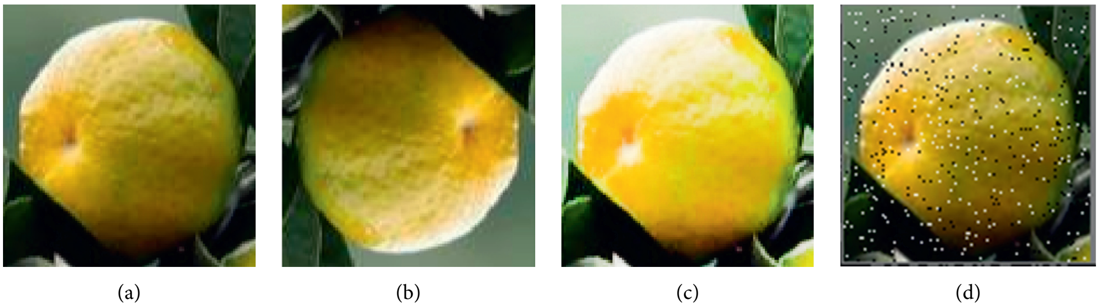


Figure 2: Image amplification effect diagram: (a) original image; (b) spin; (c) brighten; (d) adding noise.

Table 1: Description of citrus dataset.

Dataset	Species	Image size (pixels)	Number of original images	Number of images after augmentation
Citrus	Kumquat	1920 × 1080	500	3500
	Nanfeng tangerine	1920 × 1080	450	3300
	Fertile orange	1920 × 1080	400	3000
	Tangerine	1920 × 1080	400	3000

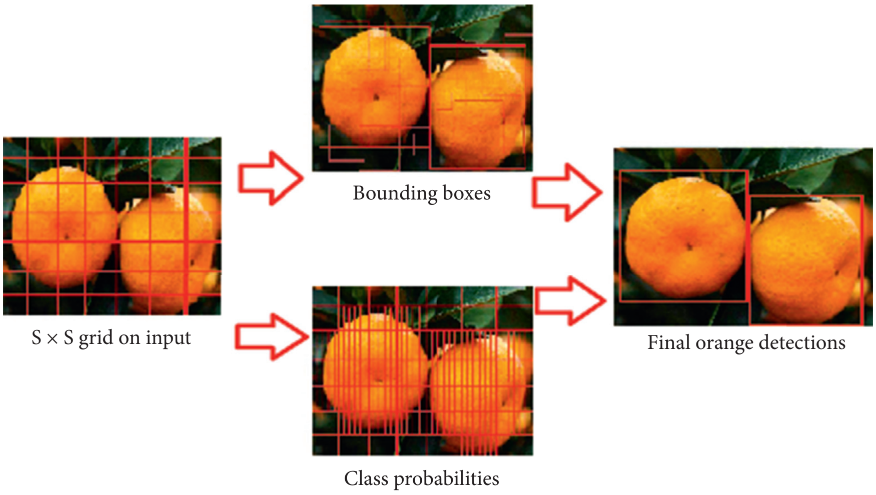


Figure 3: YOLO detection model.

(CmBN)), self-adversarial training (SAT), and Mish activation. In addition, the YOLOv4 algorithm has achieved amazing detection accuracy and speed on many datasets.

3.3. Improved YOLOv4 Model. The YOLOv4 network uses higher resolution than YOLOv3 to detect small targets and combines with the CSP module 5 times to obtain a 19 ∗ 19 size feature map, which enhances the learning ability of the convolutional neural network. In the YOLOv4 network structure, the target detection output layer contains 6 CBL units and a 1 ∗ 1 convolution. Based on the deconvolutional single shot detector (DSSD) network [23], to avoid the disappearing gradient and enhancement of feature multiplexing when the network structure is deep, 6 CBL units are changed into 2 CBL units and 2 Res units, as shown in Figure 4.

In the residual network diagram (Figure 4(c)), x is the network input, $H(x)$ is the expected output, and Res Net is only the difference $H(x) - x$ between the learning output and the input, that is, the residual $F(x)$. When the network is optimal, the module is set to 0, transferring the characteristics downward from an identical map while keeping the network in optimal condition without too many layers. Residual units can be defined as follows:

$$\begin{aligned} y_k &= f\left(x_k + F\left(x_k, w_k\right)\right), \\ f &= \max(0, x). \end{aligned} \tag{2}$$

Among them, x_k and y_k are the input and output of the k_{th} residual unit, f is the activation function, generally a modified linear unit (ReLU), and w_k is the convolution kernel [24].

In the residual unit, the Mish function [25] is used instead of the Leaky ReLU function as the activation function in the network structure. The expression of Mish activation function is as follows:

$$\text{Mish} = x \times \tan h\left(\ln\left(1 + e^{x}\right)\right). \tag{3}$$

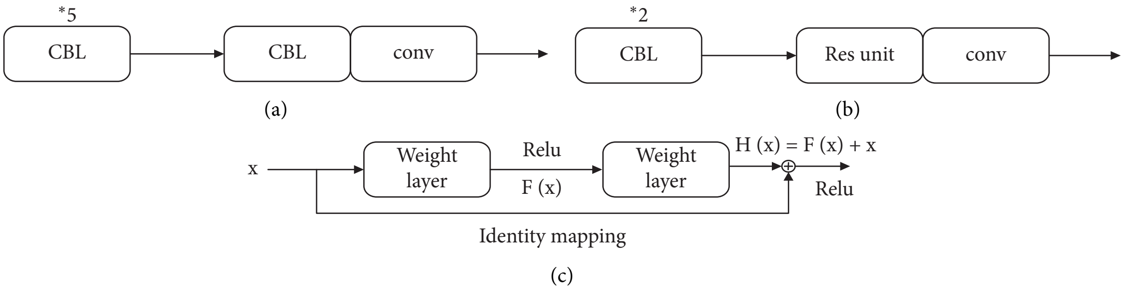


Figure 4: YOLOv4 network output structure including (a) YOLOv4 target detection output layer structure and (b) improved YOLOv4 target detection output layer structure. (c) Basic module of residual network.

The Mish function has the characteristics of no boundary (i.e., the positive value can reach any height). When x is negative, it is not completely truncated but allows a relatively small negative gradient to flow in to ensure the flow of information so that the entire loss function remains smooth.

To enable the network to obtain more feature information of small targets and improve the detection rate of small targets, we use a $152 * 152$ feature map obtained by the second residual module in the original network to detect the target, because it contains smaller target location information. Upsampling is performed twice on the 8 times downsampling feature map output by YOLOv4 to obtain a $152 * 152$ feature map, and the 2 times upsampling feature map is connected to the feature map obtained from the second residual module in the CSPDarknet53 network structure. A feature fusion target detection layer with an output of 4 times downsampling is established to detect small targets. In addition, the 2 residual units of the second residual module in the CSPDarknet53 network structure are increased to 4 residual units, and the 4 residual units of the fifth residual module in the original CSPDarknet53 network structure are reduced to 2. The residual unit maintains the original CSPDarknet53 backbone network layer number. The improved YOLOv4 network structure is shown in Figure 5.

The mosaic data enhancement method used by YOLOv4 stitches 4 pictures together, which greatly enriches the detection data set and especially adds many small targets by the random zoom. Taking advantage of this while drawing on the Stitcher method proposed by Chen Y et al. [26], the image is adjusted to a smaller component and is stitched to the same size as the conventional image. YOLOv4 uses 4 pictures for stitching. In this study, the code of the mosaic data enhancement method is modified so that an image can be stitched according to the preset number of pictures. Four types of citrus pictures in the citrus dataset are selected as training materials, and the method of controlling variables is used. In the original YOLOv4 method, only the modified mosaic data enhancement method is used, and the models with different numbers of pictures are used for training. According to the calculation formula of accuracy [27], the accuracy of the model for stitching different numbers of pictures is calculated, and the result is shown in Figure 6. This article selects 5 pictures for stitching.

3.4. Canopy Algorithm and K-Means++ Algorithm. Choosing the anchor box value suitable for the user's own dataset in the convolutional neural network as a training parameter can improve the accuracy of the final model recognition. In this study, the Canopy algorithm is used to perform coarse clustering on the dataset to provide the K value and the initial cluster centre point for the K-means++ algorithm. That is, the number of cluster centres to be divided into the dataset is selected in advance, and then the predivided K value is substituted into the K-means++ algorithm for fine clustering to obtain the anchor box value for the convolutional neural network. The algorithm flowchart is shown in Figure 7. This approach allowed us to more accurately determine the K value input and improve the network model recognition accuracy.

In the experiment presented in this paper, the label information of the dataset and the number of network feature output layers are input into the algorithm, and the algorithm automatically calculates and outputs the value and number of the prior frame. For this paper, several experiments are performed on the citrus dataset, and the clustering accuracies corresponding to the number of different a priori boxes are compared; see Table 2. Finally, the number of a priori boxes is selected as 12.

3.5. Channel-and-Layer Pruning. To ensure the improved recognition accuracy and speed, we use the layer pruning and channel pruning methods proposed by Liu et al. [28]. Without the loss of a large amount of recognition accuracy, the improved YOLOv4 model trained by the algorithm performs sparse training and pruning operations, so the model obtained after pruning has faster recognition speed and takes up less storage space, which is convenient for the subsequent use of the recognition model. The flow chart of pruning is shown in Figure 8.

In this study, using a constant scale parameter sparse strategy, the model trained by the improved YOLOv4 method is first sparsely trained; then, the CBL layer before each shortcut layer in the CSPDarknet53 backbone network is evaluated, and the gamma mean of each layer is sorted. The smallest layer is taken for pruning. Finally, by fine-tuning the model obtained after pruning, the final citrus recognition model can be obtained; the model size after pruning is shown in Table 3.

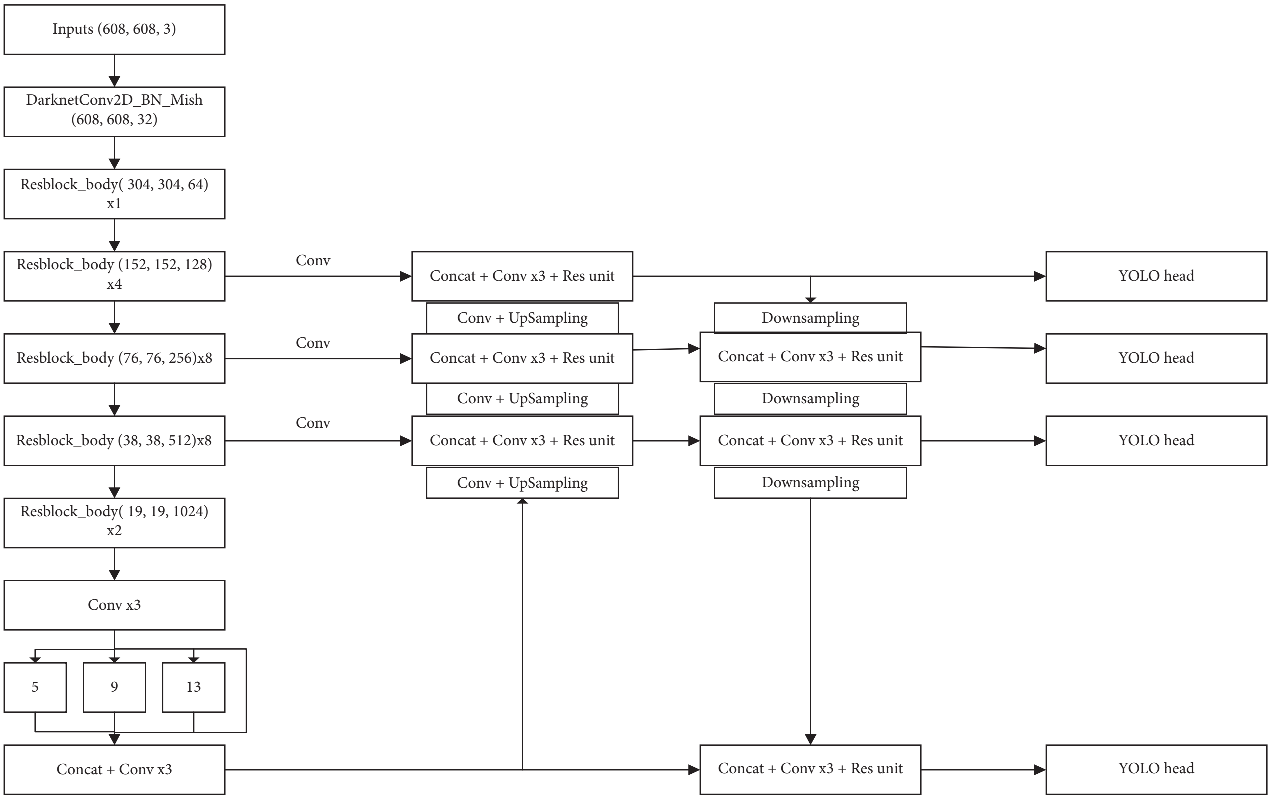

Inputs (608, 608, 3)
DarknetConv2D_BN_Mish (608, 608, 32)
Resblock_body(304, 304, 64) x1
Resblock_body (152, 152, 128) x4
Resblock_body (76, 76, 256)x8
Resblock_body (38, 38, 512)x8
Resblock_body(19, 19, 1024) x2
Conv x3
5
9
13
Concat + Conv x3
Conv
Conv
Conv
Concat + Conv x3 + Res unit
Conv + UpSampling
Concat + Conv x3 + Res unit
Conv + UpSampling
Concat + Conv x3 + Res unit
Conv + UpSampling
Downsampling
Concat + Conv x3 + Res unit
Downsampling
Concat + Conv x3 + Res unit
Downsampling
Concat + Conv x3 + Res unit
YOLO head
YOLO head
YOLO head
YOLO head

Figure 5: Improved YOLOv4 network structure.

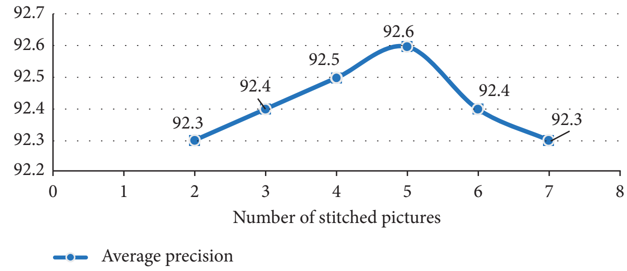


Figure 6: Picture stitching accuracy comparison with different stitching number.

4. Experimental Results and Discussion

4.1. Experimental Environment and Parameter Index. The above algorithm runs under the following computer configurations: Windows 10 operating system; Intel Core i7-8550U CPU; the graphics cards are Quadro P4000, NVIDIA 430.26 driver, CUDA 10.1, and CUDNN 7.6.5. The experiment used the DarkNet53.conv.74, yolov4.conv.137 and VGG-16 as deep learning models.

In the YOLOv3 algorithm, the batch size and subdivisions were set to 64, and the maximum number of training was 6000. The momentum, initial learning rate, weight decay regularization, and other parameters were the original parameters in the YOLOv3 model. In the YOLOv4 algorithm and improved YOLOv4 algorithm, the training steps were similar to those of YOLOv3 model training, but the training parameters are modified. The random of each YOLO layer was set to 1, enabling multiscale training. In the Faster-RCNN algorithm, the VGG_16 model was used for training.

The samples in the confusion matrix can be divided into the following four types: TP, positive sample predicted by the model; TN, negative sample predicted by the model; FP, negative sample predicted by the model, and FN, positive sample predicted by the model [29]. Precision (P), recall (R), F_1 score, mean average precision (mAP), and recognition time per image are used to evaluate the pros and cons of the model:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$
$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad (4)$$
$$F_1 = \frac{2 \times P \times R}{P + R}.$$

(i) mAP is mean average accuracy, used to measure the quality of the learned model in all categories [30].

(ii) Recognition time of each image is used to measure the speed of model recognition image after learning.

In this study, the experiment is qualitatively and quantitatively analysed as follows: (1) compared the recognition performance of the four models at different growth period and maturity when the occlusion degree does not exceed 50%, (2) compared the performance of the four models at different occlusion degrees, and (3)

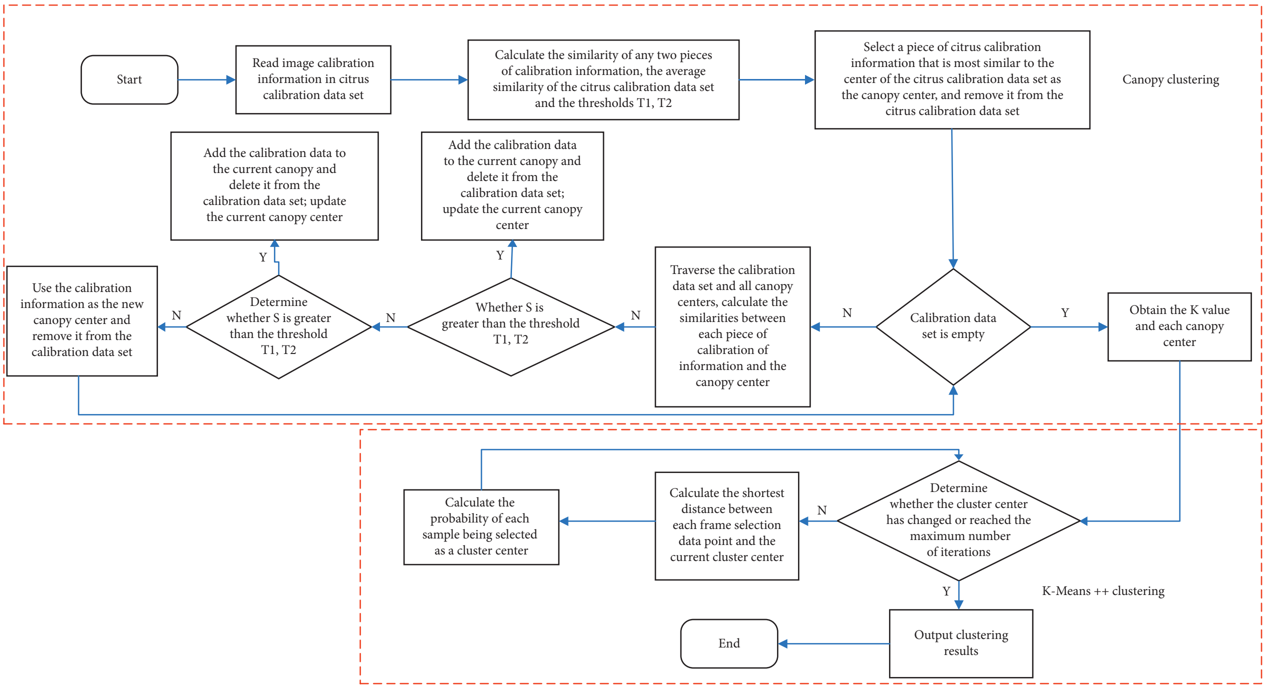

Start
Read image calibration information in citrus calibration data set
Calculate the similarity of any two pieces of calibration information, the average similarity of the citrus calibration data set and the thresholds T1, T2
Select a piece of citrus calibration information that is most similar to the center of the citrus calibration data set as the canopy center, and remove it from the citrus calibration data set
Canopy clustering
Add the calibration data to the current canopy and delete it from the calibration data set; update the current canopy center
Add the calibration data to the current canopy and delete it from the calibration data set; update the current canopy center
Y
Y
Use the calibration information as the new canopy center and remove it from the calibration data set
N
Determine whether S is greater than the threshold T1, T2
N
Whether S is greater than the threshold T1, T2
N
Traverse the calibration data set and all canopy centers, calculate the similarities between each piece of calibration of information and the canopy center
N
Calibration data set is empty
Y
Obtain the K value and each canopy center
Calculate the probability of each sample being selected as a cluster center
Calculate the shortest distance between each frame selection data point and the current cluster center
N
Determine whether the cluster center has changed or reached the maximum number of iterations
Y
K-Means ++ clustering
End
Output clustering results

Figure 7: Flow chart of combining the Canopy algorithm and K-means++ algorithm.

Table 2: Comparison of the K-Means and Canopy + K-means++ under different numbers of prior frames.

Number of prior frames	K-means (%)	Canopy + K-means++
6	65.54	86.14%
9	69.91	88.76%
12	71.21	90.67%

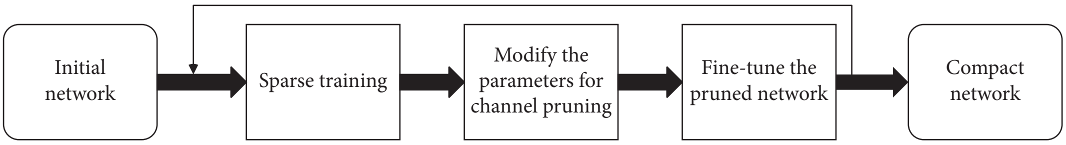

Initial network
Sparse training
Modify the parameters for channel pruning
Fine-tune the pruned network
Compact network

Figure 8: Pruning flow chart.

Table 3: Comparison of different model sizes.

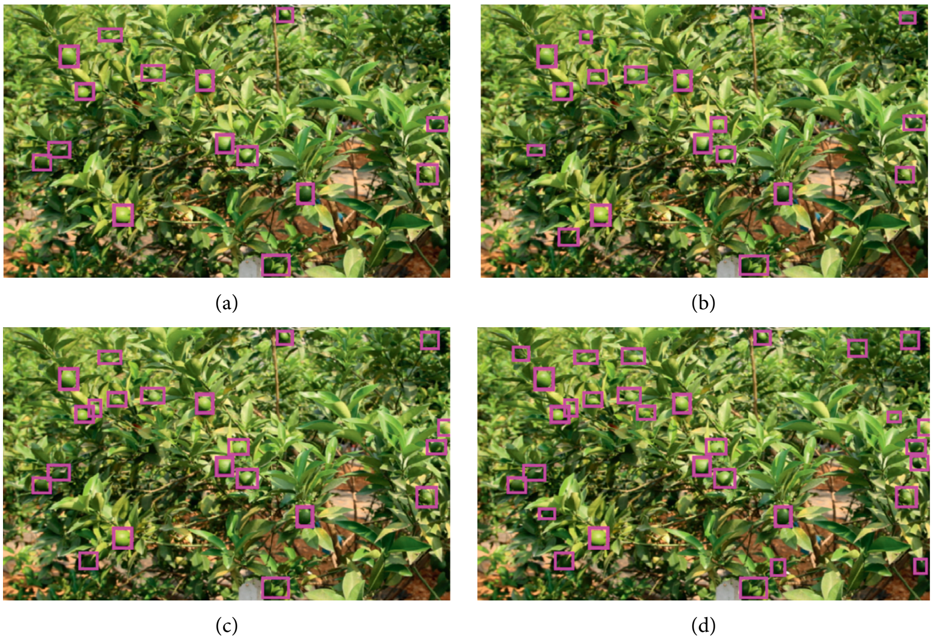
Model	Faster-RCNN	YOLOv3	YOLOv4	Improved YOLOv4
Size (M)	540	229	250	187

compared the performance of the four models in identifying kumquat, Nanfeng tangerine, fertile orange, and tangerines.

4.2. Citrus Detection at Different Growth Stages. The purpose of this experiment is to use kumquat data at different growth stages with occlusion levels not exceeding 50% to train and determine which model has better recognition performance for kumquats at different stages. The effect of model recognition is shown in Figures 9 and 10:

It can be seen from the detected pictures and Table 4 that, for growing kumquats, the improved YOLOv4 method detects a large number of citrus fruits, while the YOLOv3 and Faster-RCNN methods detect relatively few fruits and have a small number of recognition errors. For mature kumquats, the improved YOLOv4 method is superior to the other three methods. Kumquats has small fruit size during the growth period, the green fruits and leaves, and nonobvious colour characteristics. The improved detection ability of the improved YOLOv4 method is very prominent, and it will not affect the recognition accuracy because of colour and individual characteristics. Mature fruits have more obvious colour characteristics, larger single volume, and less overlap. Through multiple comparative experiments, it is proven that the improved YOLOv4 method is superior and faster than other three methods.

4.3. Detection of Citrus with Different Degrees of Occlusion. The purpose of this experiment is to use mature citrus to train citrus pictures with different degrees of occlusion and



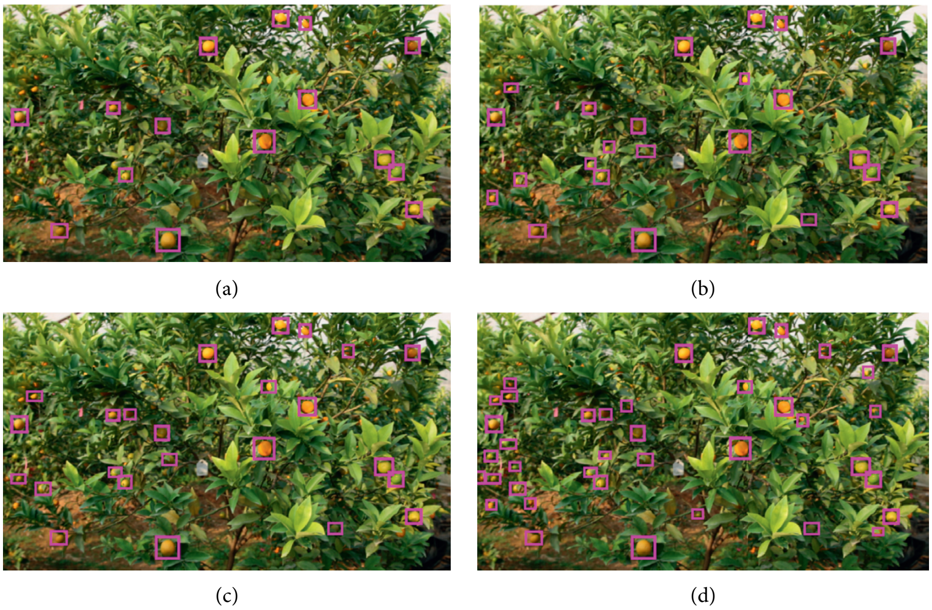
(a)

(b)

(c)

(d)

Figure 9: Detection results of four models in growth: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.



(a)

(b)

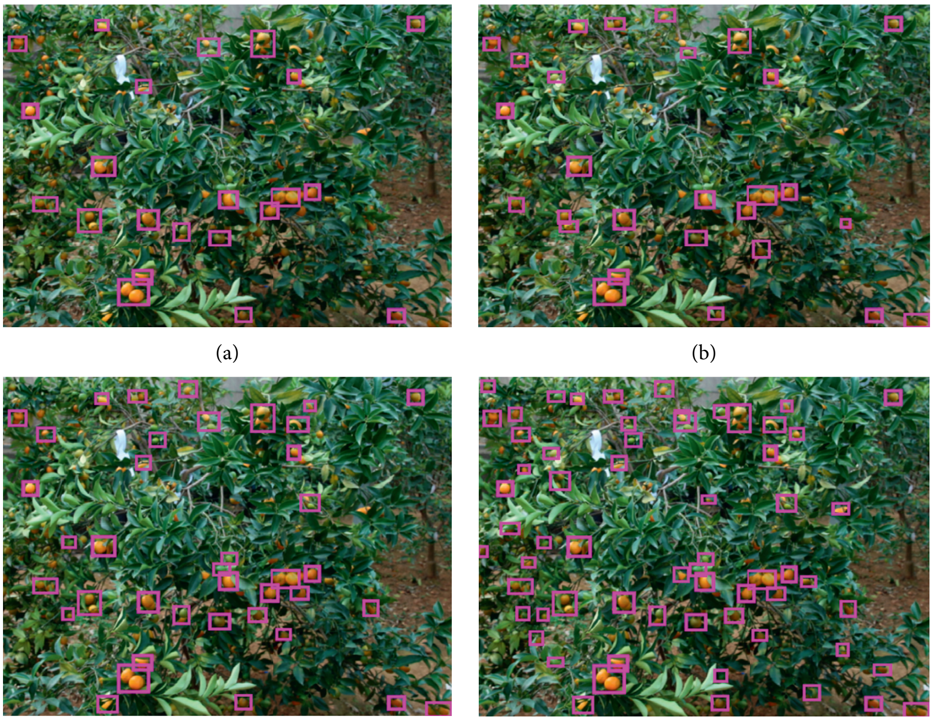
(c)

(d)

Figure 10: Detection results of four models at maturity: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

Table 4: Kumquat training parameters of four methods in different periods.

Model	Growth stage	F1 score (%)	Average time/s	Accuracy (%)
Faster-RCNN	Growing period	86.24	0.32	0.84
YOLOv3	Growing period	83.97	0.21	0.82
YOLOv4	Growing period	87.51	0.12	0.92
Improved YOLOv4	Growing period	91.95	0.10	0.95
Faster-RCNN	Maturity	86.48	0.32	0.86
YOLOv3	Maturity	84.17	0.21	0.82
YOLOv4	Maturity	87.58	0.11	0.92
Improved YOLOv4	Maturity	92.13	0.09	0.96



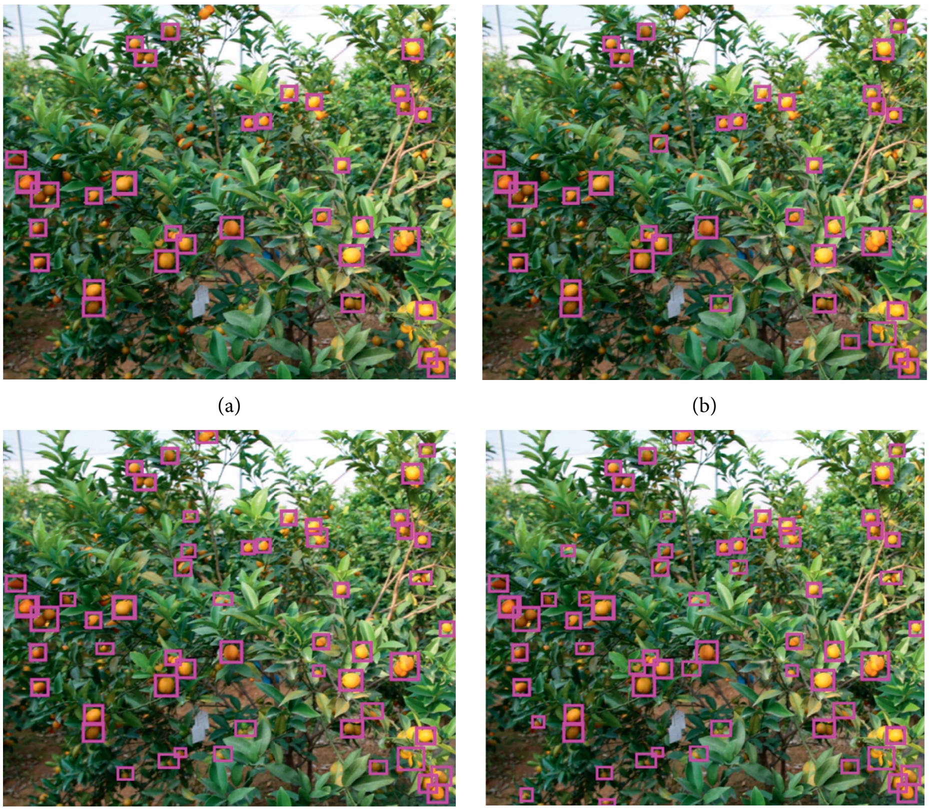
(a)

(b)

(c)

(d)

Figure 11: Effect of the four methods with occlusion over 50%: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.



(a)

(b)

(c)

(d)

Figure 12: Effect of the four methods with under 50% occlusion: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

Table 5: Identification parameters of four methods for different occlusion degrees.

Occlusion condition	Model	Citrus count	Correctly identified		Falsely identified		Missed	
			Amount	Rate (%)	Amount	Rate (%)	Amount	Rate (%)
Less than 50%	Faster-RCNN	200	162	81.24	19	9.71	31	15.44
	YOLOv3	200	156	78.22	23	11.52	38	18.96
	YOLOv4	200	173	86.38	14	7.10	22	11.21
	Improved YOLOv4	200	187	93.58	12	5.98	16	8.15
More than 50%	Faster-RCNN	200	156	78.24	26	12.81	39	19.34
	YOLOv3	200	148	74.22	35	17.52	46	22.96
	YOLOv4	200	166	83.18	20	10.05	26	13.07
	Improved YOLOv4	200	182	90.82	14	7.18	21	10.36

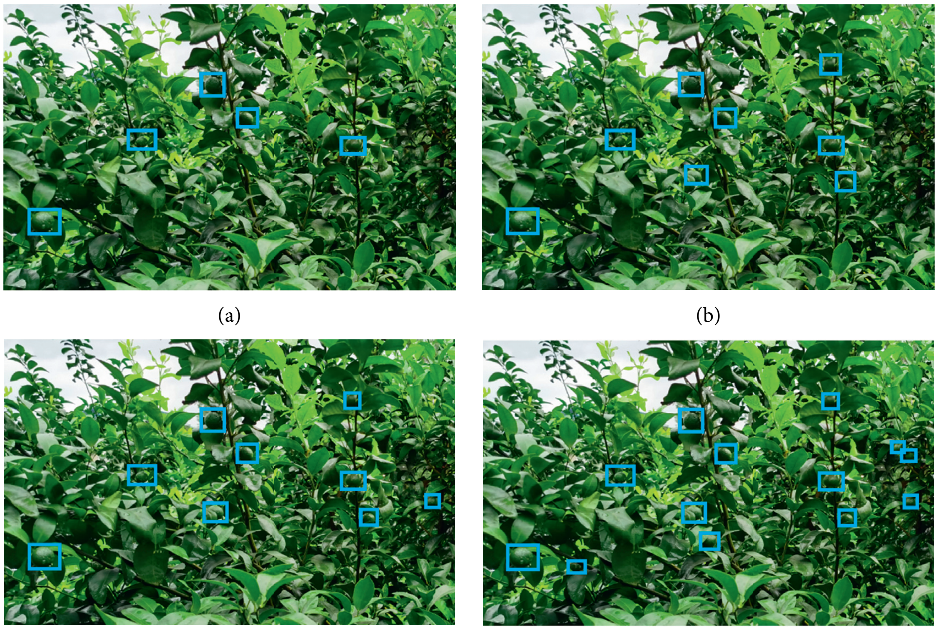

(a)
(b)
(c)
(d)

Figure 13: Detection results of tangerine: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

determine which model has better recognition performance for citrus with different degrees of occlusion. The occlusion degree of the test picture selected in this experiment is more than 50%, which means that more than half of the citrus in the picture is blocked by leaves or branches or overlapped seriously. The effects of the four models are shown in Figures 11 and 12:

As shown in the detected pictures and Table 5, when the degree of occlusion does not exceed 50%, the improved YOLOv4 method detects more citrus; meanwhile, the Faster-RCNN method detects relatively less fruit. When the degree of occlusion exceeds 50%, the Faster-RCNN model detected more fruits than the YOLOv3 model, but second only to theYOLOv4 and improved YOLOv4 methods. Combined with the research results of Osco et al. [31], the present results indicate that when dense small targets appear in groups, these dense small targets may be citrus fruits blocked by leaves or trunks; the identification and detection effect of Faster-RCNN are better than those of YOLOv3, but this deficiency is remedied in the improved YOLOv4. The improved YOLOv4 experimental results show that this method is superior to Faster-RCNN in the recognition accuracy of small targets, which provides a new feasible solution for future fruit recognition research.

4.4. *Recognition Experiment of Different Citrus Varieties.*

The purpose of this experiment is to use the images collected during the ripening period of four citrus for training to verify the generalization ability of the model. The recognition results of the four methods are shown in Figures 13–16:

Figure 17 shows the AP values of the four methods for four types of citrus. From the figure, it can be seen intuitively that the recognition accuracy of each method changes for different types of citrus.

It can be seen from the comparison between the detected picture and the AP value that the improved YOLOv4 method detects more fruit than the other three methods. This is partly due to the fact that the smaller fruit and various distances from the fruit can be seen from the image after

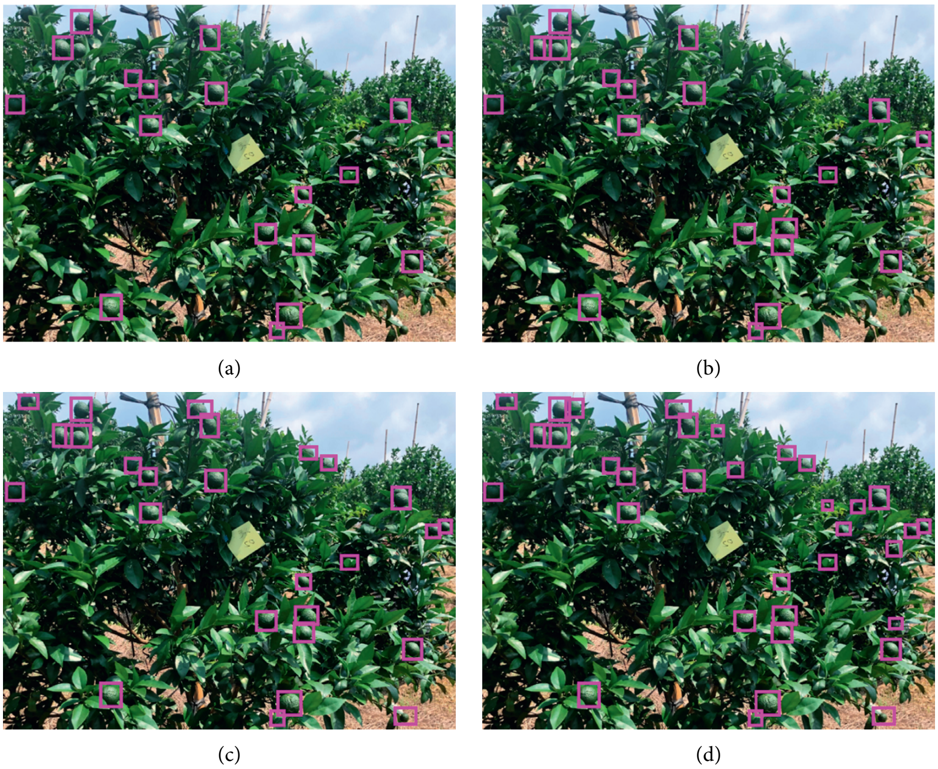


Figure 14: Detection results of fertile orange: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

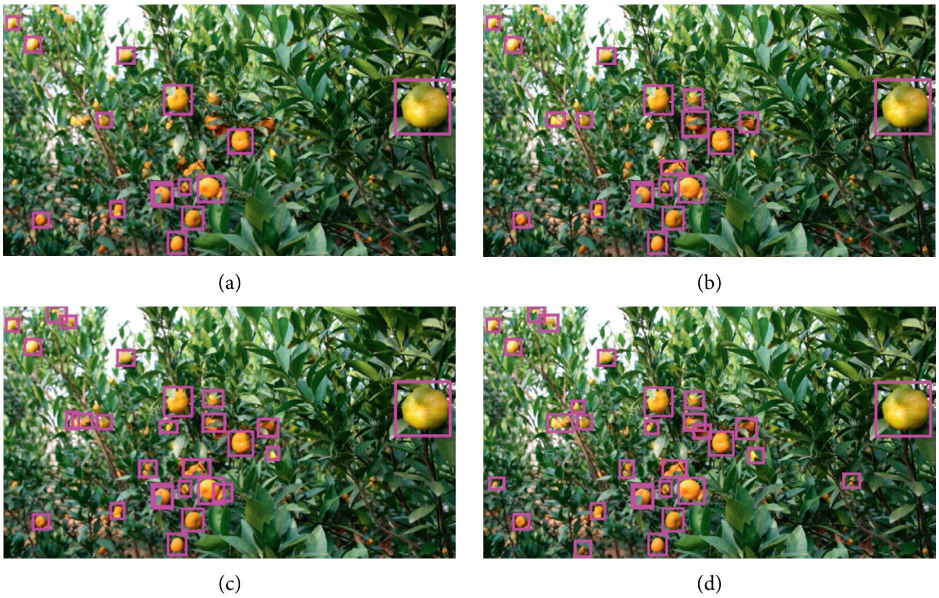


Figure 15: Detection results of Nanfeng tangerine: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

being blocked by leaves. The Faster-RCNN method has a poorer recognition effect on distant fruits when other fruits are present. The experimental results show that the improved YOLOv4 model has a good performance on the detection accuracy of different citrus varieties and has a good generalization ability. In a picture, when a large target and a small target with the same features appear at the same time, the recognition accuracy of the improved YOLOv4 method is also very impressive, and the YOLOv3 method may result in misjudgements.

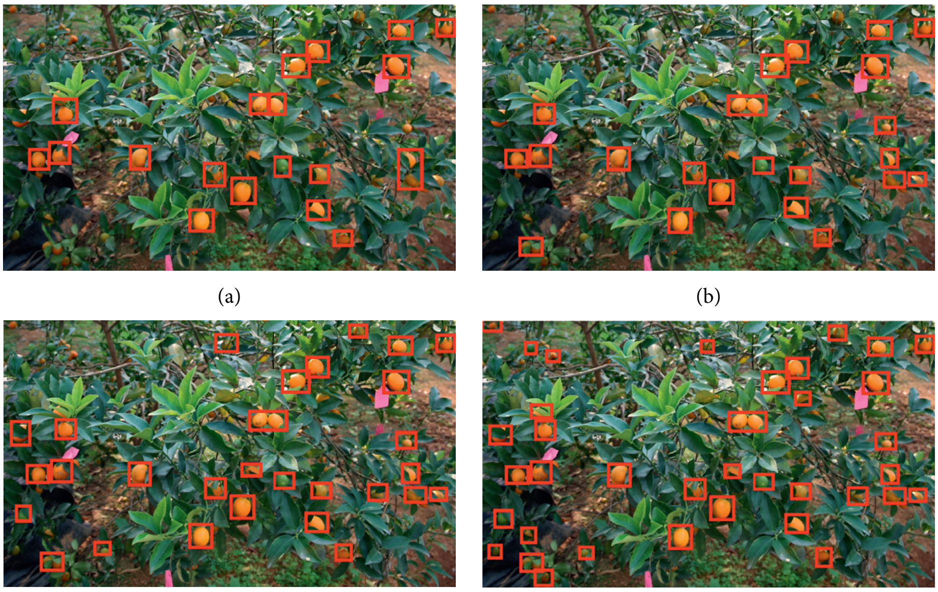


Figure 16: Detection results of kumquat: (a) YOLOv3; (b) Faster-RCNN; (c) YOLOv4; (d) improved YOLOv4.

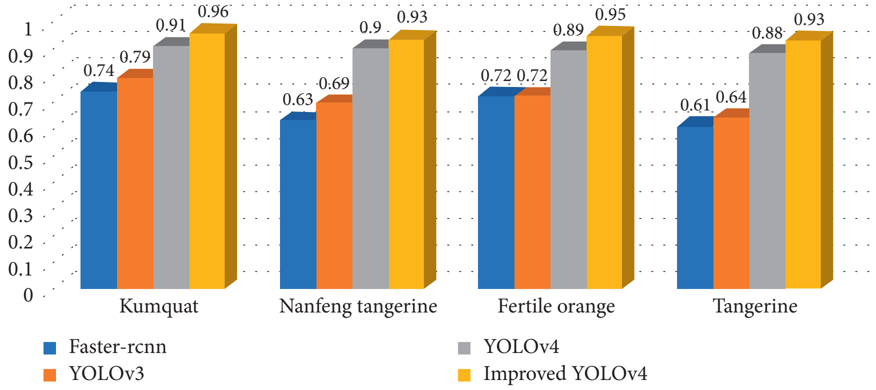


Figure 17: AP values of four methods for four types of citrus.

5. Conclusions

Based on four deep learning algorithms, Faster-RCNN, YOLOv3, YOLOv4, and improved YOLOv4, this paper presents the results of target recognition of four types of citrus in different periods. Experiments were conducted on three different datasets. The conclusions are as follows.

For citrus fruits with insignificant colour characteristics during their growth period and smaller individuals, the improved YOLOv4 method performs best among the four algorithms, followed by YOLOv4. For citrus with bright colour characteristics during maturity, larger individuals, and citrus with relatively dense growth, the improved YOLOv4 algorithm is also superior to the other three methods in accuracy and detection speed.

In the recognition of small targets (highly blocked), the improved YOLOv4 algorithm performs best, followed by YOLOv4; in the recognition of sparse large targets (partially blocked), the performance of the improved YOLOv4 algorithm is also the best.

Among the four algorithms, the improved YOLOv4 algorithm has the highest accuracy and generalization ability. In the detection of multiple varieties of citrus, it has a good performance and does not affect the detection accuracy of the algorithm.

In summary, in our experiment, the overall recognition performance presented by the improved YOLOv4 is optimal. Although the time and space spent on training the model are larger than those for the other models, the space and time can be reduced by pruning later. Finally, it is easy to apply these algorithms to different scenarios.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Shanghai Agriculture Applied Technology Development Program (no. G2015060402), the National Natural Science Foundation of China (no. 61762013), the Science and Technology Foundation of Guangxi Province (no. 2018AD19339), and Research Fund of Guangxi Key Lab of Multi-Source Information Mining & Security (No. 20-A-02-02). The authors also thank the Guangxi Scholarship Fund of the Guangxi Education Department for the support.

References

[1] J. J. Guo, Z. P. Gao, J. L. Xia et al., "Comparative analysis of chemical composition, antimicrobial and antioxidant activity of citrus essential oils from the main cultivated varieties in China," *Lebensmittel-Wissenschaft & Technologie*, vol. 97, pp. 825–839, 2018.

[2] J. J. Zhuang, S. M. Luo, C. J. Hou, Y. Tang, Y. He, and X. Y. Xue, "Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications," *Computers and Electronics in Agriculture*, vol. 152, pp. 64–73, 2018.

[3] Z. Zhai, Z. Xu, X. Zhou et al., "Recognition of hazard grade for cotton blind stinkbug based on Naive Bayesian classifier," *Transactions of the CSAE*, vol. 31, no. 1, pp. 204–211, 2015.

[4] X. Huang, G. Li, C. Ma et al., "Green peach recognition based on improved discriminative regional feature integration algorithm in similar background," *Transactions of the CSAE*, vol. 34, no. 23, pp. 142–148, 2018.

[5] C. Wang, X. Li, Z. Wu et al., "Machine vision detecting potato mechanical damage based on manifold learning algorithm," *Transactions of the CSAE*, vol. 30, no. 1, pp. 245–252, 2014.

[6] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: a review," *Computers and Electronics in Agriculture*, vol. 116, pp. 8–19, 2015.

[7] G. Lin, Y. Tang, X. Zou, J. Li, and J. Xiong, "In-field citrus detection and localisation based on RGB-D image analysis," *Biosystems Engineering*, vol. 186, p. 3444, 2019.

[8] G. Lin, Y. Tang, X. Zou, J. Cheng, and J. Xiong, "Fruit detection in natural environment using partial shape matching and probabilistic Hough transform," *Precision Agriculture*, vol. 21, no. 1, pp. 160–177, 2020.

[9] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, pp. 1–12, 2017.

[10] F. Li, Y. Jiang, T. Li, Y. Feng, and S. Chen, "Design of a robot end effector with measurement system for precise pick-and-place of square objects," *Procedia Manufacturing*, vol. 48, pp. 172–180, 2020.

[11] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] Y. Osako, H. Yamane, S.-Y. Lin, P.-A. Chen, and R. Tao, "Cultivar discrimination of litchi fruit images using deep learning," *Scientia Horticulturae*, vol. 269, p. 109360, 2020.

[13] A. Koirala et al., "Deep learning–method overview and review of use for fruit detection and yield estimation," *Department of Agriculture and Food*, vol. 162, pp. 219–234, 2019.

[14] A. Koirala, K. B. Walsh, fnm Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of "MangoYOLO"" *Precision Agriculture*, vol. 20, no. 6, pp. 1107–1135, 2019.

[15] J. Xiong, Z. Zheng, J. Liang, Z. Zhong, B. Liu, and B. Sun, "Citrus detection method in night environment based on improved YOLO v3 network," *Transactions of the CSAE*, vol. 51, no. 4, pp. 199–206, 2020.

[16] H. Kang and C. Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors*, vol. 19, no. 20, p. 4599, 2019.

[17] M. Chen, Y. Tang, X. Zou et al., "Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology," *Computers and Electronics in Agriculture*, vol. 174, p. 105508, 2020.

[18] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.

[19] T. Haas, C. Schubert, M. Eickhoff, and H. Pfeifer, "BubCNN: bubble detection using Faster RCNN and shape regression network," *Chemical Engineering Science*, vol. 216, Article ID 115467, 2020.

[20] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: an improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.

[21] D. Wu, Q. Wu, X. Yin et al., "Lameness detection of dairy cows based on the YOLOv3 deep learning algorithm and a relative step size characteristic vector," *Biosystems Engineering*, vol. 189, pp. 150–163, 2020.

[22] A. Bochkovskiy, C.-Y. Wang, H. Yuan, and M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[23] C. Y. Fu, W. Liu, A. Ranga et al., "DSSD: deconvolutional single shot detector," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, p. 1701, Honolulu, HI, USA, July 2017.

[24] J. DAI and J. TONG, "Galaxy morphology classification using deep residual networks," *Progress in Astronomy*, vol. 36, pp. 384–397, 2018.

[25] D. Misra, "Mish: a self regularized non-monotonic neural activation function," 2019, https://arxiv.org/vc/arxiv/papers/1908/1908.08681v2.pdf.

[26] Y. Chen, P. Zhang, Z. Li et al., "Stitcher: feedback-driven data provider for object detection," 2020, https://arxiv.org/abs/2004.12432.

[27] D. Font, T. Pallejà, M. Tresanchez et al., "A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm," *Sensors*, vol. 14, no. 7, pp. 11557–11579, 2014.

[28] Z. Liu, J. Li, Z. Shen et al., "Learning efficient convolutional networks through network slimming," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[29] S. S. Mehta, C. Ton, S. Asundi, and T. F. Burks, "Multiple camera fruit localization using a particle filter," *Computers and Electronics in Agriculture*, vol. 142, pp. 139–154, 2017.

[30] B. Jiang, Q. Wu, X. Yin et al., "FLYOLOv3 deep learning for key parts of dairy cow body detection," *Computers and Electronics in Agriculture*, vol. 166, pp. 172–180, 2019.

[31] L. P. Osco, M. D. S. D. Arruda, J. Marcato Junior et al., "A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 160, pp. 97–106, 2020.