A Counting Method of Red Jujube Based on Improved YOLOv5s

Qiao, Yichen; Hu, Yaohua; Zheng, Zhouzhou; Yang, Huanbo; Zhang, Kaili; Hou, Juncai; Guo, Jiapan

Published in:
Agriculture (Switzerland)

DOI:
10.3390/agriculture12122071

agriculture

Article

A Counting Method of Red Jujube Based on Improved YOLOv5s

Yichen Qiao [1], Yaohua Hu [2,*], Zhouzhou Zheng [1], Huanbo Yang [1], Kaili Zhang [1], Juncai Hou [1,*] and Jiapan Guo [3,4]

[1] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China
[2] College of Optical, Mechanical, and Electrical Engineering, Zhejiang A&F University, Hangzhou 311300, China
[3] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, 9747 AG Groningen, The Netherlands
[4] Data Science Center in Health (DASH), University Medical Center Groningen, University of Groningen, 9713 GZ Groningen, The Netherlands
[*] Correspondence: huyaohua@zafu.edu.cn (Y.H.); houjuncai@nwsuaf.edu.cn (J.H.);
Tel.: +86-15291680166 (Y.H.); +86-18792954818 (J.H.)

Abstract: Due to complex environmental factors such as illumination, shading between leaves and fruits, shading between fruits, and so on, it is a challenging task to quickly identify red jujubes and count red jujubes in orchards. A counting method of red jujube based on improved YOLOv5s was proposed, which realized the fast and accurate detection of red jujubes and reduced the model scale and estimation error. ShuffleNet V2 was used as the backbone of the model to improve model detection ability and light the weight. In addition, the Stem, a novel data loading module, was proposed to prevent the loss of information due to the change in feature map size. PANet was replaced by BiFPN to enhance the model feature fusion capability and improve the model accuracy. Finally, the improved YOLOv5s detection model was used to count red jujubes. The experimental results showed that the overall performance of the improved model was better than that of YOLOv5s. Compared with the YOLOv5s, the improved model was 6.25% and 8.33% of the original network in terms of the number of model parameters and model size, and the Precision, Recall, F1-score, AP, and Fps were improved by 4.3%, 2.0%, 3.1%, 0.6%, and 3.6%, respectively. In addition, RMSE and MAPE decreased by 20.87% and 5.18%, respectively. Therefore, the improved model has advantages in memory occupation and recognition accuracy, and the method provides a basis for the estimation of red jujube yield by vision.

Keywords: count red jujubes; red jujube; improved YOLOv5s; ShuffleNet V2 Unit; Stem; BiFPN

check for updates

Citation: Qiao, Y.; Hu, Y.; Zheng, Z.; Yang, H.; Zhang, K.; Hou, J.; Guo, J. A Counting Method of Red Jujube Based on Improved YOLOv5s. *Agriculture* **2022**, *12*, 2071. https://doi.org/10.3390/agriculture12122071

Academic Editors: Vadim Bolshev, Vladimir Panchenko and Alexey Sibirev

Received: 10 October 2022
Accepted: 30 November 2022
Published: 2 December 2022

1. Introduction

Chinese red jujube is a kind of characteristic fruit which is famous for its various nutritional ingredients [1]. With the increasing demand for red jujubes, it is more and more important to count red jujubes so as to provide a basis for the estimation of jujube yield through vision. Due to the increasing supply of red jujubes, the count of red jujubes will play an important role in the planting and production management. Therefore, it is of great significance to realize the count of red jujubes, and it will help improve the utilization rate of red jujubes. However, the development of artificial intelligence, it provides a new way to solve the problem of low fruit production efficiency [2].

It is an important task of orchard management to estimate the fruit yield by counting the number of fruits. Deep learning has become a potential tool for counting the number of fruits, and It enables automatic feature extraction from data sets. At the same time, by extracting the basic parameters of crop growth, intelligent agricultural technology enables farmers to estimate crop yield, thus reasonably arranging the production and processing of red jujubes [3]. Machine learning methods, such as the Watershed algorithm [4] and the kalman filter algorithm [5], are widely used to count fruit. However, because the supervised learning method in machine learning can't capture the nonlinear relationship between input and output variables and the uncertainty of the crop environment, it is difficult for traditional machine learning methods to develop a reliable crop counting model. However, in recent years, the progress of technology has made it possible to develop advanced crop counting models by using deep learning. Shileiliu et al. [6] proposed a light target detection YOLOv5-CS model, which could realize the object detection and accurate counting of green citrus in the natural environment. The map of the model was 98.23%. ZhangYanchao et al. [7] used the YOLOX target detection network to detect and count the holly fruits, and the map was 95%.

Owing to the improvement of computer hardware and the development of computer vision technology, deep learning has been widely used in various industries [8–10]. Object detection algorithm based on deep learning mainly includes One-Stage and Two-Stage. The first type is the detection algorithm based on candidate region, such as R-CNN (Region-Convolutional Neural Networks) [11], Fast R-CNN (Fast Region-Convolutional Neural Networks) [12], Faster R-CNN (Faster Region-Convolutional Neural Networks) [13]. The second kind regards the detection of target position as a regression problem and directly uses CNN (Convolutional Neural Network) for images, such as SSD (Single Shot Multi-Box Detector) [14,15], YOLO (You Only Look Once) [16–19].

Computer vision technology has also been widely used in various fields [20–23]. The image processing technology is one of the key technologies in precision agriculture, and it is mainly used in classification, localization, and yield prediction [24]. Mulyono et al. [25] proposed a texture extraction method based on a gray-level co-occurrence matrix that is followed by a K-nearest neighbor for the classification of litchii. Sutarno et al. [26] adopted similar ideas to extract texture information and then used the learning vector quantization (LVQ) algorithm as the classifier to classify durian based on their color, shape, and texture. The method was difficult to detect the subtle feature changes among different fruits, and the accuracy of fruit classification was 89%. Zhao et al. [27] proposed a matching algorithm that used the sum of absolute transformed differences (SATD) for fruit detection, followed by the support vector machine (SVM) classifier. The accuracy of recognition reached more than 83%. Dorj et al. [4] proposed forecasting the yield of citrus yields. The method preprocessed images by color space conversion and denoising then recognized and detected citrus and counted citrus by the watershed segmentation algorithm. Other researchers have also studied the fruit classification, identification, and count of fruits based on shape invariant moments [28], decision trees [29], and Hough [30] combined with the texture and color of fruits. The above methods use single features or multi-feature combinations with texture features, shape size, and color differences of fruits to recognize fruits. The recognition result is about 93% when the environment is complex, such as light changes, fruit overlap, leaf occlusion, etc. In addition, the traditional machine learning algorithm is limited by the result of the classifier of the algorithm itself, and it is difficult for the algorithm to complete the object detection of fruit in a complex environment [31].

Due to the occlusion of fruit and leaves, the image transformation, and the background switching in complex orchard environments, the deep learning-based object detection algorithm can solve these problems quickly and effectively with its powerful learning ability and feature representation capability. Fu et al. [32] proposed a deep convolutional neural network detection model in which the improved Faster R-CNN was trained end-to-end by using back-propagation, random gradient descent algorithm, and ZFNet (Zeiler and Fergus networks) for kiwifruit detection. The experiment showed that the model could improve the accuracy of fruit recognition to 96%. Liu et al. [33] fused RGB and NIR images to identify kiwifruit by VGG16. The average detection precision of an image was 90.7%, and the detection time was 0.134 s on one image. Wang et al. [34] proposed an improved model of a lightweight detection network of SSD. The model used a modified DenseNet network as the backbone to replace the first three additional layers in SSD and incorporate a multi-level fusion structure. Compared with the original model, the number of parameters of the improved model was reduced by

11.14×10^6, and the average precision was increased by 2.02%. The classical deep learning networks have been successful in fruit identification and detection. There are advantages of high accuracy and efficiency in the identification and detection of fruits. However, the networks are relatively large, which is not conducive to the application of mobile equipment in the agricultural field. Many researchers have already studied the lightweight model. For instance, Li et al. [35] applied the adaptive spatial pyramid to detect the green peppers and the accuracy reached 96.11% in YOLOv4_tiny. Zhang et al. [31] used MobileNet-v3 as the feature extraction network of YOLOv4-LITE. The improved model reduced the model size and improved the detection speed. Therefore, it is feasible to reduce the weight of the model while ensuring the precision of model detection.

The lightweight model will be beneficial to the application of agricultural mobile equipment and realize the intelligence of agricultural equipment. In order to ensure the detection accuracy of the model in complex unstructured orchards and counting fruit, a counting method of red jujube based on improved YOLOv5s was proposed. The main goal of this research was to reduce the size of the model while ensuring its detection accuracy and speed in an embedded device. The effectiveness of counting red jujubes in a complex environment was comprehensively considered from four aspects in this research

(1) ShuffleNet V2 was used as the backbone of the network to extract the feature map and make the model lightweight.
(2) The Stem, a novel data loading module, was proposed to reduce data information loss and improve model detection accuracy.
(3) The original PANet (Path Aggregation Network) structure was improved to BiFPN (Bidirectional Feature Pyramid Network) for multi-scale feature fusion to enhance the model feature fusion capability and improve the model accuracy.
(4) The improved YOLOv5s detection model was used to count red jujubes.

The second section introduced the method of making the dataset, the improved red jujubes detection algorithm, the counting method of red jujubes, and the training of the network. The third section introduced the test results of the model and the analysis compared with other algorithms. In the last section, the counting methods of red jujubes were summarized and prospect.

2. Materials and Methods

In this section, the acquisition and production of the dataset were mainly introduced. Then, a detection algorithm based on the improved yolov5s of red jujube was proposed, and a counting method for red jujubes was presented. Finally, the training method of the network was introduced, as shown in Figure 1.

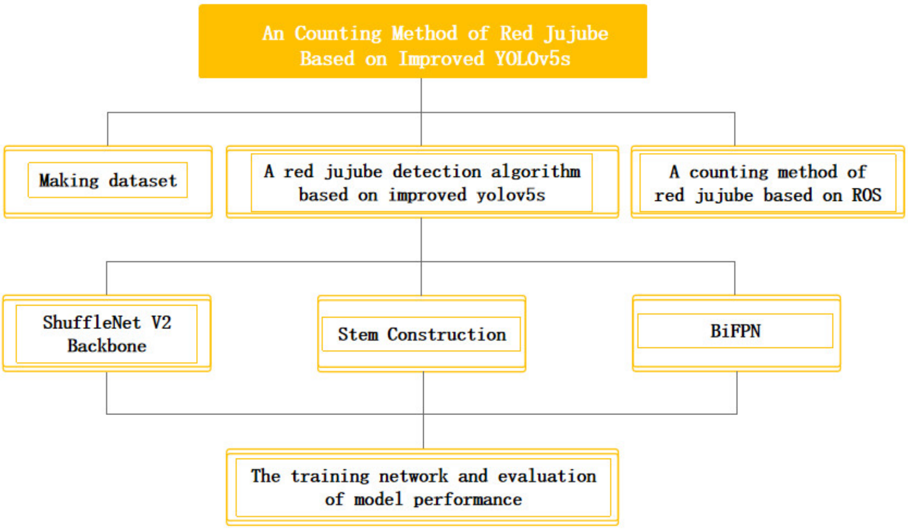

An Counting Method of Red Jujube Based on Improved YOLOv5s
Making dataset
A red jujube detection algorithm based on improved yolov5s
A counting method of red jujube based on ROS
ShuffleNet V2 Backbone
Stem Construction
BiFPN
The training network and evaluation of model performance

Figure 1. A counting method of red jujube based on improved YOLOv5s.

2.1. Image Data Acquisition

The dataset of red jujube, including Jun jujube and Gray jujube, in this study, was collected from a red jujube orchard from 5 October to 9 October in Alar City, Xinjiang, China.

Images of Jun jujube and Gray jujube were taken in a jujube orchard of the 13th company of a group in Alar City, Xinjiang Uygur Autonomous Region. In order to ensure the reliability of the experimental results, the jujube image dataset was collected, which was under different illumination at 9:00 a.m., 3:00 p.m., and 9:00 p.m. for red jujubes. The resolution of the images was 1080 × 1920 pixels, with a total of 1026 original images, which included illumination changes, leaf shading, and fruit overlap. In order to improve the robustness of this model, each image contained one or more different scenarios. The distribution of the dataset is shown in Table 1.

Table 1. Distribution of dataset of red jujubes.

Dataset	Grey Jujube	Jun Jujube	Total Number
illumination change images	136	190	326
leaf shading images	132	225	357
fruit overlap images	139	204	343
Total Number	407	619	1026

2.2. Data Preprocessing and Augmentation

The collection of data sets would affect the recognition effect of the target detection model. The more sufficient and comprehensive the data set is, the better the generalization ability and robustness of the model. Therefore, the number of samples could be expanded by data amplification. In order to truly simulate the shooting of red jujube in a complex environment and apply it to the detection network, this research used Opencv in python to compress and cut the images into 640 × 640. Then, the images were randomly enhanced by different image processing methods [36], such as rotating 180, mirroring, adding salt and pepper noise which set the threshold to 0.5, and changing the image brightness by setting the threshold to 1.3 and 0.7, as shown in Figure 2. Repeated random image processing on an image many times. After enhancement, a total of 10,000 images were obtained as the data set of the model.

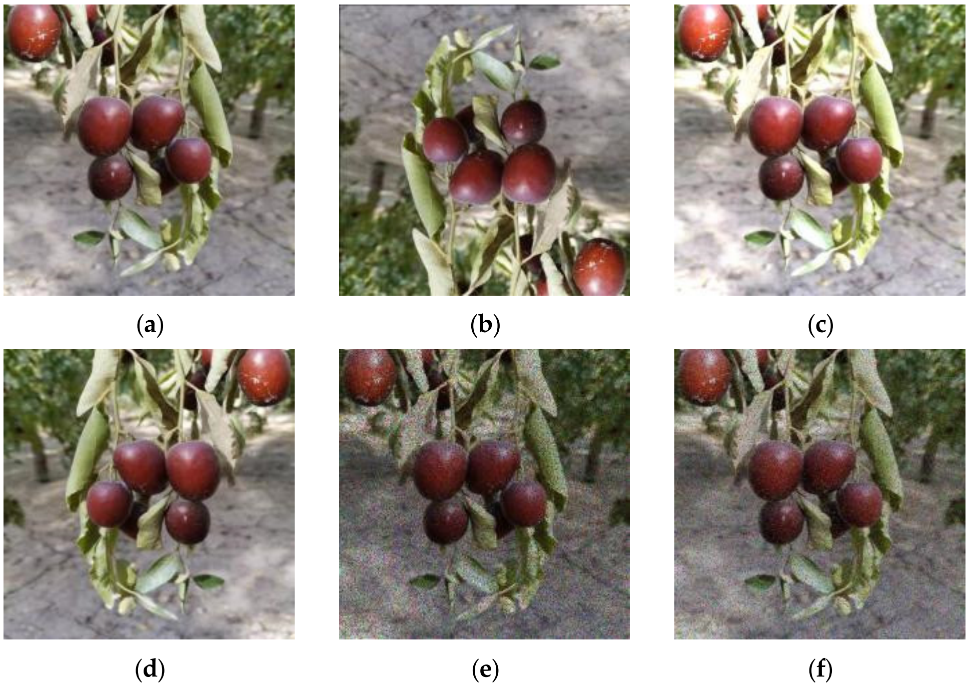
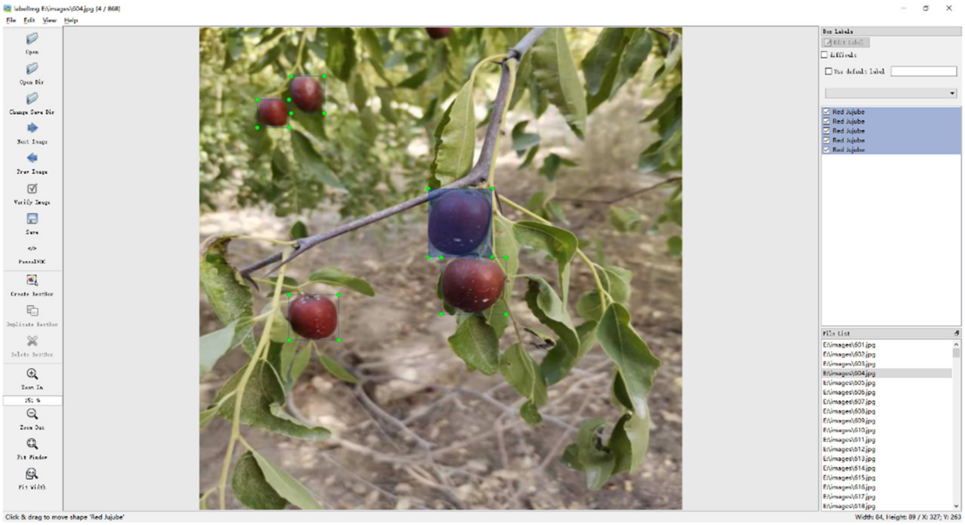


Figure 2. Image sample after data preprocessing and augmentation. (**a**) original image, (**b**) rotating by 180°, (**c**) Increasing brightness, (**d**) mirroring image, (**e**) adding noise, (**f**) reducing brightness.

2.3. *Images Annotation and Dataset Division*

In this research, LabelImg was used to label red jujube in the data set with artificial rectangular boxes, as shown in Figure 3. The dataset was divided into 80% training datasets, 10% validation datasets, and 10% test datasets. The final image samples of the training set, verification set, and test set are 8000, 1000, and 1000 respectively.



labelImg E:\images\604.jpg [4 / 868]

☐ difficult

☐ Use default label

☑ Red Jujube
☑ Red Jujube
☑ Red Jujube
☑ Red Jujube
☑ Red Jujube

E:\images\601.jpg
E:\images\602.jpg
E:\images\603.jpg
E:\images\604.jpg
E:\images\605.jpg
E:\images\606.jpg
E:\images\607.jpg
E:\images\608.jpg
E:\images\609.jpg
E:\images\610.jpg
E:\images\611.jpg
E:\images\612.jpg
E:\images\613.jpg
E:\images\614.jpg
E:\images\615.jpg
E:\images\616.jpg
E:\images\617.jpg
E:\images\618.jpg

Click & drag to move shape 'Red Jujube'

Width: 84, Height: 89 / X: 327; Y: 263

Figure 3. LabelImg data set annotation.

2.4. *Methodologies*

The Yolo series are effective in single-stage object detection, and their miniature detection models guarantee higher accuracy, taking into account faster speed and fewer parameters. Therefore, the lightweight detection models of the Yolo series are more suitable to be applied to embedded devices to develop mobile agricultural equipment. However, due to the complexity of the agricultural production environment and the harsh working environment, it is difficult to meet the agricultural production for the simple detection algorithm. Based on YOLOv5s, the original backbone network was replaced by the ShuffleNet V2 backbone network in this research, which significantly reduced the number of parameters of the network. The Foucs were replaced by the Stem to resist partial information missing from the feature map. PANet was replaced by BiFPN to enhance the model feature fusion capability and improve the model accuracy. Finally, the improved YOLOv5s detection network was used to identify the image and count red jujubes.

2.4.1. Yolov5s Network

YOLOv5 is improved by adding some new ideas on the basis of YOLOv4, and its detection accuracy and speed have been greatly improved. The YOLOv5 can be divided into four types according to the size of the model: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, among which the YOLOv5s model is the smallest. YOLOv5s mainly consists of four parts: Input, Backbone, Neck, and Prediction.

In order to improve the speed and accuracy of the network, the Mosaic data augmentation is used in the YOLOv5 to stitch images by random cropping, scaling, and lining up. YOLOv5s uses adaptive anchor box calculation to set the initial anchor boxes for different datasets and calculates the difference between the bounding boxes and the ground truth. YOLOv5s updates the anchor boxes in the reverse iteration to adaptively calculate the best anchor box for different training sets. To adapt different sizes of images in the dataset, YOLOv5 uses adaptive image scaling to fill the scaled image with the least amount of black edges, which reduces the computation and improves the speed. Backbone will perform information extraction on the feature maps. It mainly includes Focus, CBS, and C3. The input image is sliced by the Focus and convolved by one convolution with 32 kernels, as shown in Figure 4. CBS consists of a convolution, a batch normalization, and the SiLU. The SiLU is defined as follows:

$$SiLU(x) = \frac{x}{1 + exp(-x)} \tag{1}$$

where, x represents the feature map.

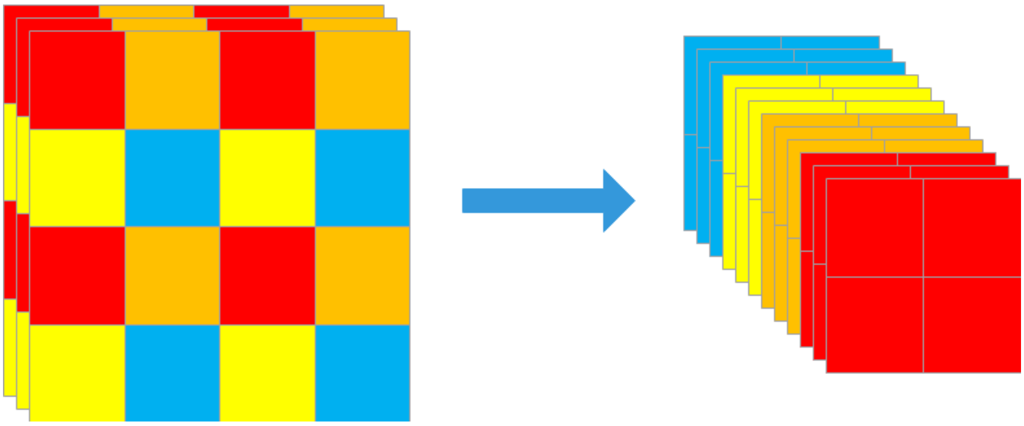


Figure 4. Foucs structrue.

As a new structure of BottlenneckCSP, C3 contains 3 CBS modules and several Bottlenecks. The C3 is used repeatedly in YOLOv5s to extract more information. As shown in Figure 5, the SPP (spatial pyramid pooling) introduces three different pooling kernels of 5 × 5, 9 × 9, and 13 × 13, and it connects different feature maps to expand the respective field, which effectively separates the most important features and improves the accuracy of the model.

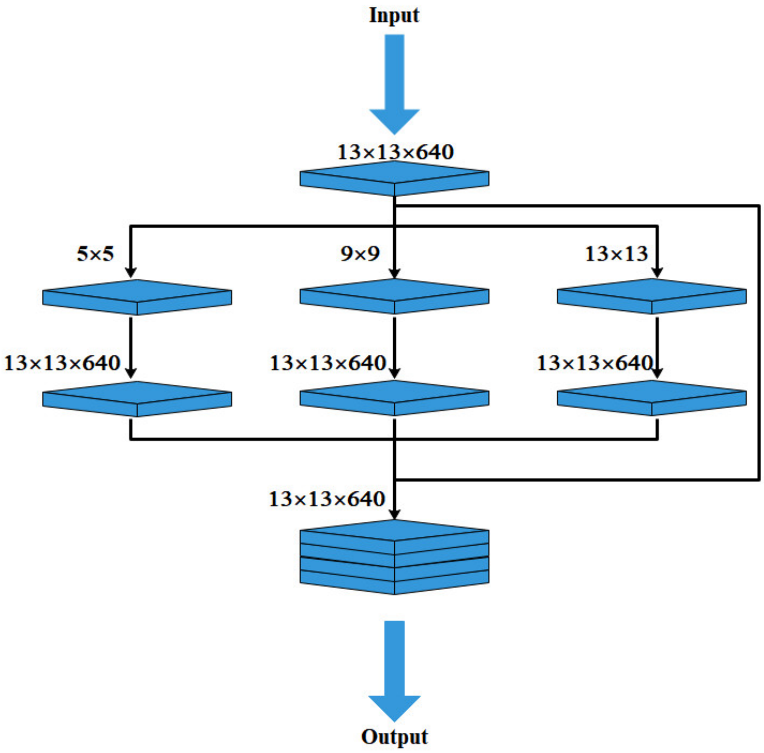


Figure 5. SPP structure.

To utilize most of the backbone information, the Neck of YOLOv5 uses the FPN + PAN. Feature Pyramid Network (FPN) solves the problem of different input feature map sizes by constructing an image pyramid on the feature map. PAN, as the innovative point of path aggregation network (PANet) [37], downsamples the image from FPN and then performs concat on the image. To improve the ability of image recognition and localization, FPN acquires the semantic features of the image from the top, while PAN gets the localization features of the image from the bottom.

There are some regression loss functions used in object detection tasks, such as the Smooth Loss function [16], IOU Loss function [38], GIOU Loss function [39], DIOU Loss function [40], and CIOU_Loss function [41]. In the Prediction, YOLOv5 uses CIOU_Loss as the loss function of the Bounding box. The CIOU_Loss function is defined as follows:

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

where, IOU represents the intersection ratio of the prediction box to the object box. b represents the center point of the prediction box. b^{gt} represents the center point of the object box. $\rho^2(b, b^{gt})$ represents Euclidean distance squared between the center point of the prediction box and the center point of the object box. c represents the diagonal length of the two closed boxes. α represents a positive trade-off parameter. v represents the consistency of the aspect ratio.

2.4.2. ShuffleNet V2 Backbone

YOLOv5s reduces the parameters of the model by C3 and improves the speed of the model, but the C3 is very complicated, with a large amount of calculation and still needs a lot of memory. The YOLOv5 lightweight model based on ShuffleNet V2 was designed, which greatly reduced the model parameters. The ShuffleNet V2 backbone was designed by using ShuffleNet V2 Units [42], and the backbone of the original model was replaced by the ShuffleNet V2 backbone.

As a lightweight convolutional neural network that is suitable for application to mobile devices, ShuffleNet V2 was first proposed in 2018. Compared with ShuffleNet V1, ShuffleNet V2 adopts the way of channel Shuffle, which divides the characteristic channels into two parts, ensuring that the input and output channels are the same, One part enters the bottleneck, and the other part does not run. Excessive point convolution will increase computational complexity. ShuffleNet V2 replaces the grouped point convolution with the standard point convolution. ShuffleNet V2 puts the channel shuffle after the dimensional stacking to prevent fragmentation of the model. ShuffleNet V2 replaces element-wise operators with concat to reduce the time of model detection. The basic model units of ShuffleNet V2 are divided into two types. The ShuffleNet V2 Units are shown in Figure 6. ShuffleNet V2 introduces channel shuffle. First, the channels of the input feature map are divided into two branches. The two branches directly connect to the concat. There are two 1 × 1 point convolution layers and a 3 × 3 group convolution layer with a stride size of 2 in the other branch. The convolution layers contain a batch normalization layer and ReLu. The other basic model unit of ShuffleNet V2 differs from the previous model, where two convolution layers: a 3 × 3 group convolution layer with a stride of 2 and a 1 × 1 point convolution layer. Finally, two images of branches of the same size were spliced together. In order to extract information on different-size feature maps, the ShuffleNet V2 backbone was designed to replace the backbone by using 16 ShuffleNet V2 Units in YOLOv5s.

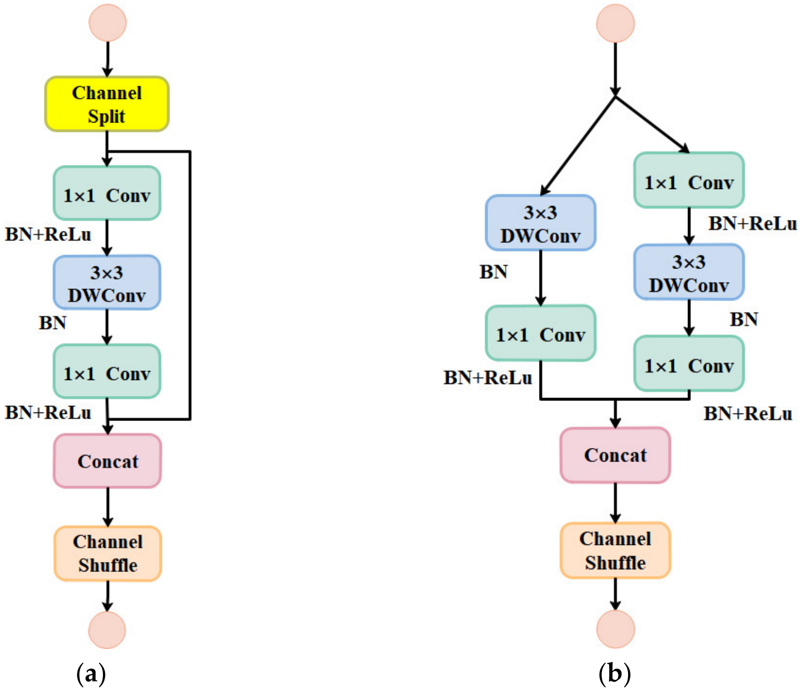


Figure 6. The structure of ShuffleNet-v2 Units. (**a**) the structure of ShuffleNet-v2 Unit1. (**b**) the structure of ShuffleNet-v2 Unit2.

2.4.3. Stem Construction

Inception-v4 [43] was proposed in 2017, which confirmed that residual connectivity largely accelerated the training speed of Inception networks. With reference to the design idea of Inception-v4, the Stem was proposed to rapidly reduce the resolution of the input feature maps, ultimately achieving a top-5 error rate of 3.08% on ILSVRC. The feature map is continuously reduced from 299 × 299 to 35 × 35 by Stem in the InceptionV4 network, and it has many convolution layers, which is better for complex task feature extraction. However, the task is simpler to detect a single target of red jujube, which will cause excessive calculation. The Stem is shown in Figure 7. In order to reduce the parameters of the model, the model could be pruned. Inspired by the idea of fast feature map resolution reduction, four CBS were adopted to make the size of the feature map to be suitable for the network, where 3 × 3 convolutions with the stride of 2 were used in the first and third CBS and 1 × 1 convolution was used in the second and fourth CBS. In contrast to the Foucs, which sliced the feature map into 32 small feature maps before image concat, the Stem used two 3 × 3 convolutions with the stride of 2 to reduce the feature map sizes and concated it with the feature map of the maximum pooling layer, so that the number of parameters was reduced while improving the feature extraction ability of the network and improving the accuracy.

2.4.4. BiFPN

With the deepening of the network level, the semantic information of image features gradually changes from a low dimension to a high dimension. As shown in Figure 8, the PANet structure was used to fuse the multi-scale features of images in the original YOLOv5s detection network. In order to improve the detection accuracy of red jujubes, the BIFPN network, a weighted bidirectional feature pyramid network, was applied to the detection of red jujubes. Compared with the traditional feature fusion network, BiFPN introduced weight to make it more sensitive to important features and makes better use of feature information of different sizes.

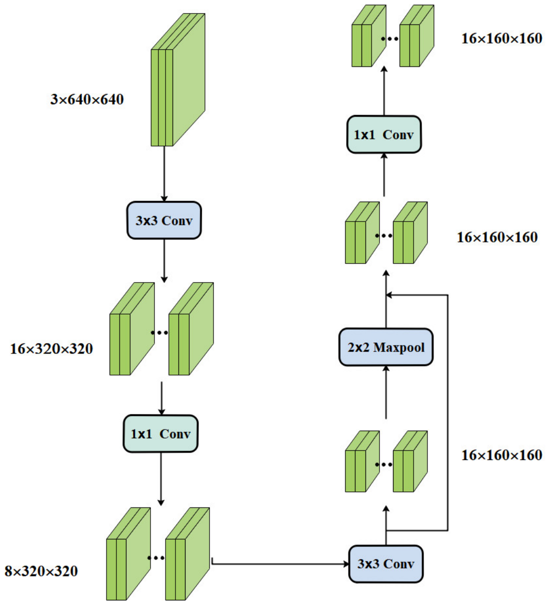


Figure 7. The structure of the Stem.

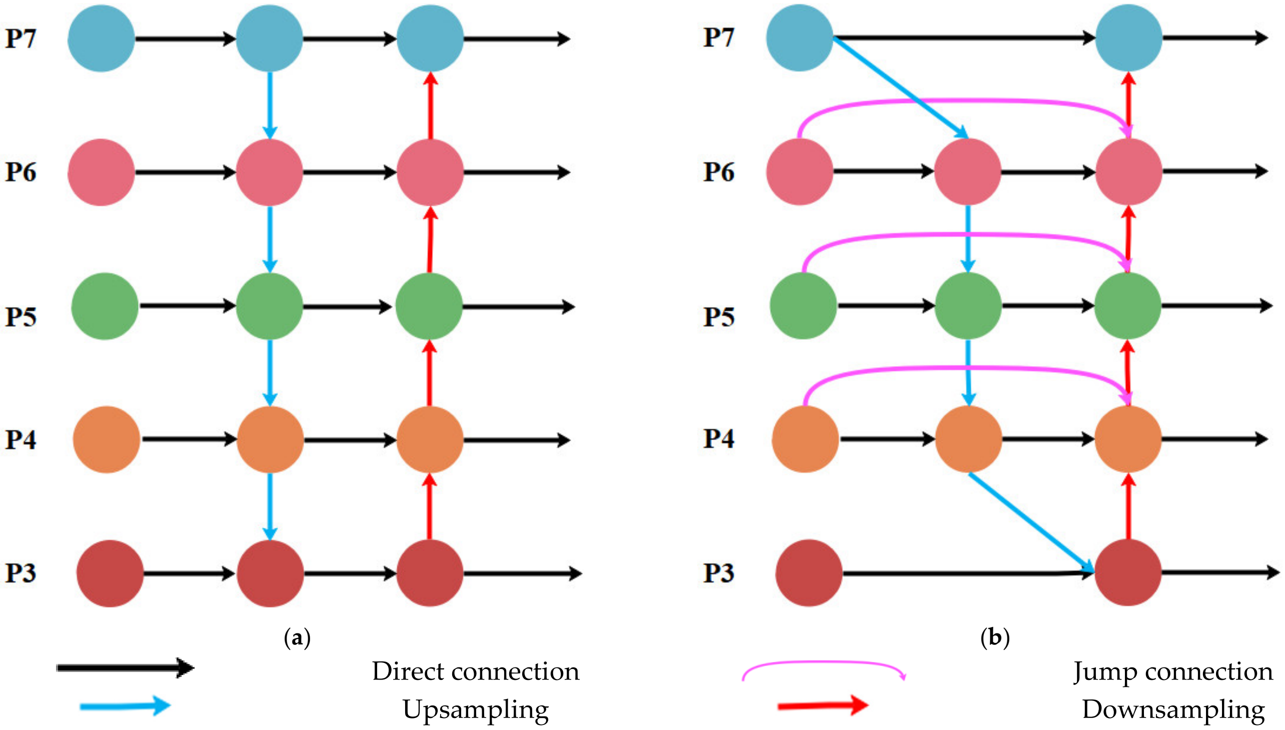


Figure 8. Bi-directional feature fusion network. (**a**) PANet with bi-directional feature fusion network, (**b**) BiFPN with bi-directional feature fusion network.

In this research, BiFPN was introduced in the neck of YOLOv5s, as shown in Figure 9. Because the node, which had only one input edge and no ability of feature fusion, made little contribution to the feature fusion of the network. Therefore, deleting this node had little effect on network feature fusion. When the original input node and the output node were in the same layer, an extra edge was added between the output node and the input node, and feature fusion was realized without increasing too much computational overhead. Different from the PANet structure of YOLOv5s, when performing feature fusion, each bidirectional path was used as a feature network layer, and the feature network layer was reused at the same layer, thus realizing a higher level of feature fusion.

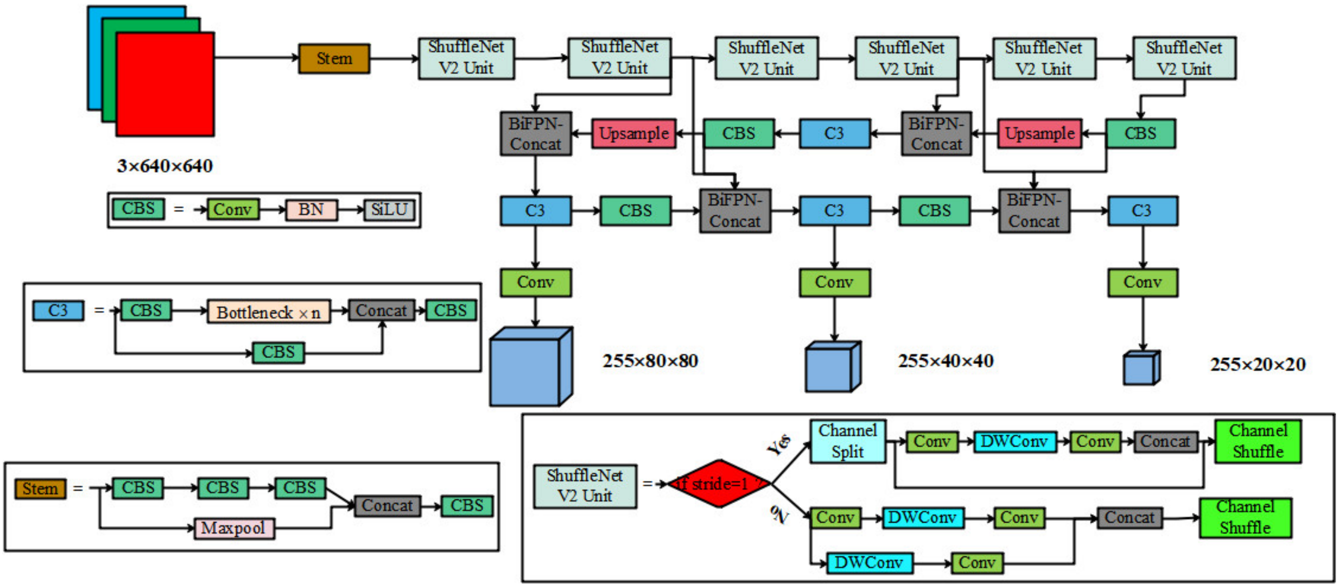
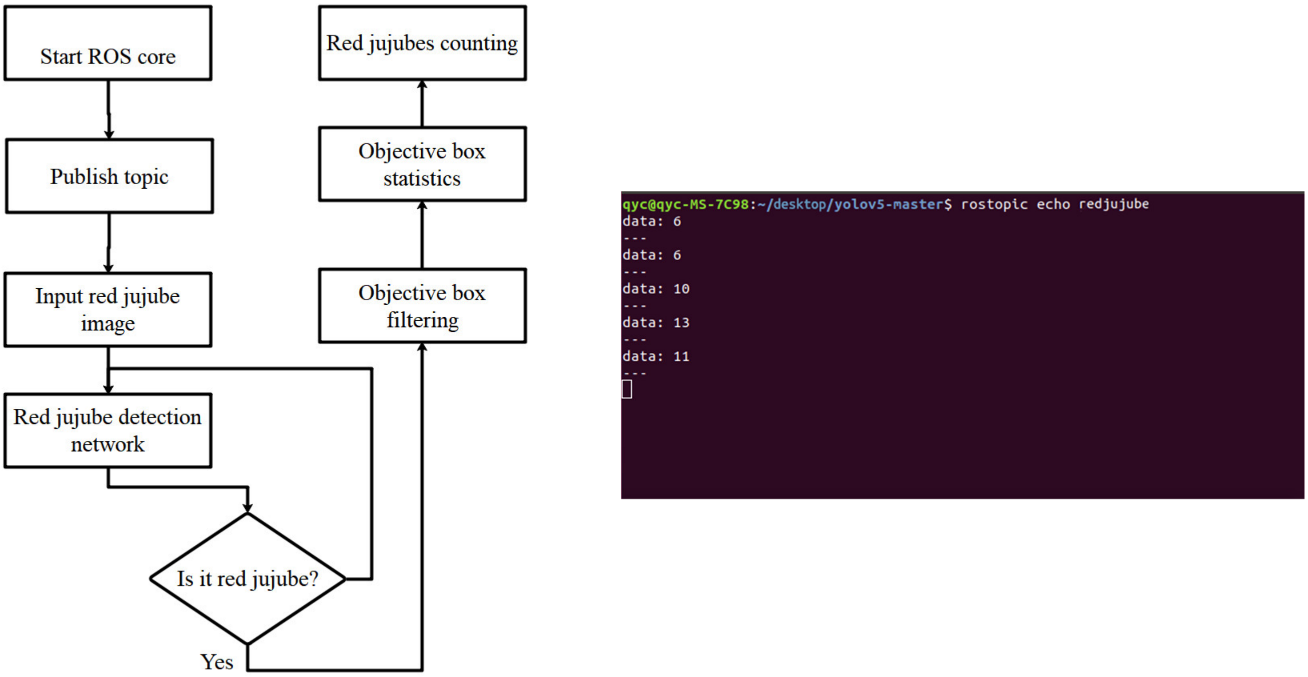


Figure 9. The structure of the improved YOLOv5s model.

2.4.5. Counting Method of Red Jujube

The counting method of red jujubes was based on the improved jujube target detection algorithm. This research used ROS to count red jujubes. The detection steps were as follows: (1) Starting ROS core and publishing topics; (2) the improved YOLOv5s were used to detect the target of jujube fruit, and the target detection frame and corresponding features were obtained; (3) counting the number of target detection frames, as shown in Figure 10a. The detection results are shown in Figure 10b.



```
qyc@qyc-MS-7C98:~/desktop/yolov5-master$ rostopic echo redjujube
data: 6
---
data: 6
---
data: 10
---
data: 13
---
data: 11
---
```

(**a**) (**b**)

Figure 10. Counting method of red jujube. (**a**) the process of the red jujube counting method; (**b**) the results of the jujube counting method.

2.5. Test Platform

The experiment was conducted on an improved YOLOv5s architecture with Pytorch based on Python 3.8. The details of the experimental setup are shown in Table 2.

Table 2. Experimental environment.

Configuration	Parameter
CPU	Intel(R) Core(TM) i7-10700K
GPU	NVIDIA GeForce RTX 3070
Accelerated environment	CUDA11.1 CUDNN8.2.1
Development environment	Pycharm2021.3.2
Operating system	Windows 10

The batch size was 4, and the epochs were 400. The adaptive matrix estimation algorithm (Adam) was used to optimize the model. The initial learning rate was 0.001, and the momentum was 0.9. The weight of the model was saved once every training session, and the best weight was also saved.

2.6. Evaluation of Model Performance

In order to evaluate the performance of our model of red jujube, Precision (P), Recall (R), Average Precision (AP), Parameters, Model Size, and detection speed (Fps) were chosen in the article, root mean square error (RMSE) and average absolute percentage error rate (MAPE) were used as evaluation indexes of jujube quality where Recall, Precision, F1-score, RMSE, and MAPE were defined as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \times 100\% \tag{3}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times 100\% \tag{4}$$

$$\mathrm{F1-score} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(\mathrm{y_i}-\hat{\mathrm{y}}_\mathrm{i})^2} \tag{6}$$

$$\mathrm{MAPE} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \frac{|(\mathrm{y_i}-\hat{\mathrm{y}}_\mathrm{i})/\mathrm{y_i}|}{\mathrm{m}} \times 100\% \tag{7}$$

where, TP represents the number of true positive samples, FP represents the number of false positive samples, and FN represents the number of false negative samples. The variable $\mathrm{y_i}$ represents the actual number of red jujubes in each image, $\hat{\mathrm{y}}_\mathrm{i}$ represents the number of red jujubes predicted by each image model, and m represents the number of image samples.

3. Results and Discussion

3.1. Performance Comparison Using the Different Improve Method

As shown in Table 3, Recall and Precision were based on a 0.5 threshold. As one of the important indicators for evaluating the model, the area of the Precision-Recall curve was larger, and the AP of the model was higher.

Table 3. The model performance with a different module.

Model	Precision (%)	Recall (%)	F1-Score (%)	AP (%)	Parameters	Model Size (KB)	Fps
YOLOv5s	89.10	90.30	89.70	95.60	7,063,542	14,052	35.10
YOLOv5s + Stem	87.60	93.90	90.60	96.00	7,281,341	14,026	38.40
YOLOv5s + BiFPN	88.60	90.90	89.70	95.30	7,063,542	14,052	39.40
YOLOv5s + ShuffleNet V2	83.80	91.60	87.50	94.00	490,205	1322	35.50
YOLOv5s + Stem + BiFPN	89.70	94.50	92.00	96.20	7,281,341	14,026	39.40
YOLOv5s + Stem + ShuffleNet V2	93.70	89.20	91.40	95.90	441,606	1149	36.30
YOLOv5s + BiFPN + ShuffleNet V2	83.40	92.10	87.50	94.10	490,205	1322	35.50
Our model	93.40	92.30	92.80	96.20	441,606	1149	36.50

ShuffleNet V2 was used as the backbone network of the network, resulting in a reduction in model parameters by 14.41 times and an increase in Fps from 35.10 to 35.47. The improved network could reduce model parameters and increase detection speed. BiFPN was applied to the red jujube detection network. The experimental result showed that BiFPN improved the average accuracy of the network without increasing the parameters of the network. At the same time, it improved the detection speed of the model, with the average accuracy increased by 0.20% and the Fps increased to 39.40. Therefore, BiFPN could enhance the feature fusion ability of YOLOv5s and speed up the detection speed of the model. The Focus was replaced by the Stem, and the improved network has been improved in Recall, F1-score, AP, model size, and Fps, among which the Recall has increased by 3.600%. So, Stem is more effective than Focus in jujube detection. Compared with YOLOv5s, the AP increased by 0.6%, but the parameters increased, which increased the calculation pressure of testing equipment when Stem and BiFPN were used at the same time. When Stem and ShuffleNet V2 were applied at the same time, compared with YOLOv5s, the parameters were greatly reduced, but the detection accuracy was also lower. Our method not only reduced the model parameters but also improved the detection accuracy. The parameters and model size of the improved model was 6.25% and 8.33% of the original network, respectively. The Precision, Recall, F1-score, AP, and Fps were increased by 4.30%, 2.00%, 3.10%, 0.60%, and 3.99%, respectively.

As a lightweight network model, YOLOv5s has high accuracy and can meet the detection of small targets in complex environments, but it is difficult to be satisfied with the identification and localization of red jujubes under limited computation. When locating and recognizing overlapping fruits, the original YOLOv5s tended to easily identify two red jujubes that were mutually obscured as the one red jujube, as shown in Figure 11b. The main reason was that the differences were small between mutually obscured fruits, and the original YOLOv5s did not extract enough feature information about them, causing false detection. In recognition of small red jujube targets, the original YOLOv5s easily missed the red jujubes that were obscured by a large area of leaves or caused by the camera being too far away, as shown in Figure 11e. The main reason was that the environment of outdoor was complex, and the discrimination of the red jujubes was large. The improved model could accurately detect red jujubes and could also accurately identify the blocked jujubes, as shown in Figure 11c, and the number of missed jujubes was obviously less than the original YOLOv5s, as shown in Figure 11f.

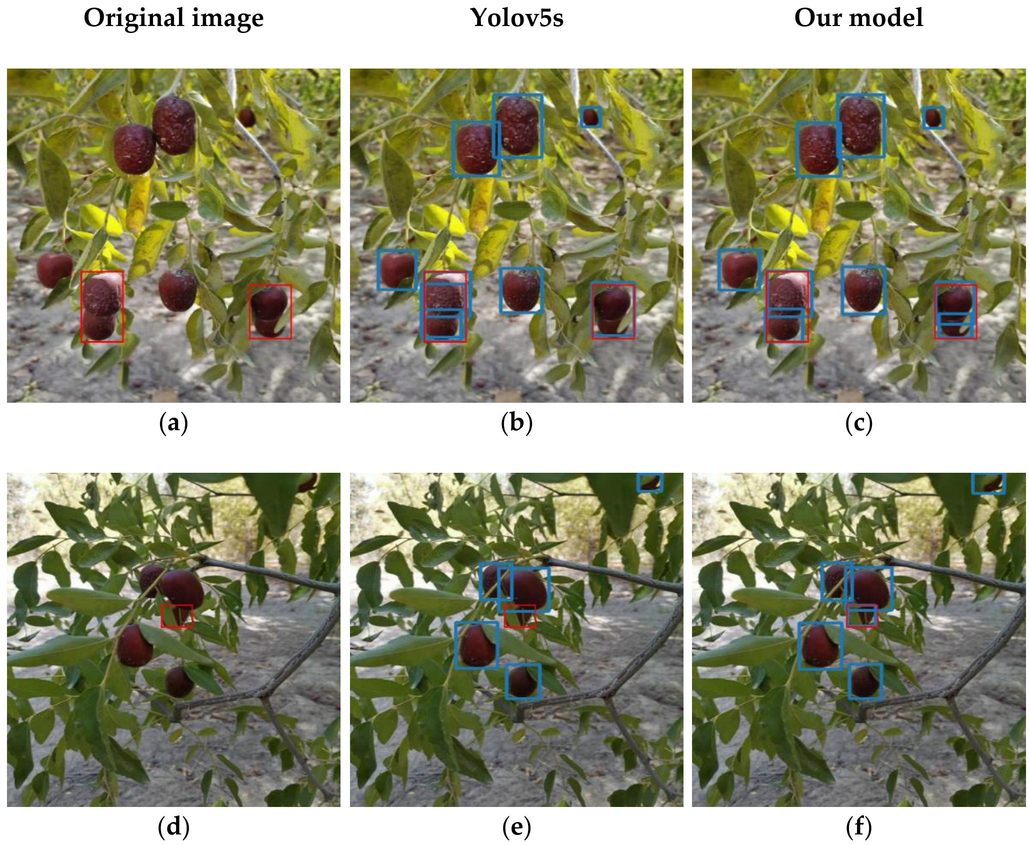


Figure 11. The results of different algorithms for the recognition of red jujube. (**a**) the original image of a dense jujube sample. (**b**) the original model to dense jujube detection image. (**c**) the improved model to dense jujube detection image. (**d**) the original image of leaf-obscured jujube. (**e**) the original model to leaf-obscured jujube detection image. (**f**) the improved model to leaf-obscured jujube detection image. Where the red boxes are the label boxes marked manually, and the blue boxes are the test results of model test.

3.2. Performance Comparison Using the Different Lightweight Backbone Networks

In order to embed mobile devices, the ShuffleNet V2 backbone network was used in YOLOv5s in this research. MoblieNet V3, as the improved version of MoblieNet V1 and MoblieNet V2, has a large improvement in detection efficiency. In order to verify the detection performance of the improved model, the MoblieNet V3 network was used as the backbone of YOLOv5s to compare the improved YOLOv5s, which used the ShuffleNet V2 backbone network and YOLOv5s. The results show that after adding MoblieNet V3 as the backbone, the network has a large improvement in Precision, but a large decrease in Recall, resulting in the improved YOLOv5s, which is used the MoblieNet V3 backbone network and the original YOLOv5s in the same AP, as shown in Table 4. In addition, there is a phenomenon of missing the detection of jujube fruit, as shown in Figure 12. The improved YOLOv5s, which is used in the MoblieNet V3 backbone network, has a significant reduction in parameters and Model Size with YOLOv5s. Therefore, using a lightweight network as the backbone reduces the size of the model while maintaining accuracy.

Table 4. The comparison of different backbone networks.

Model	Precision (%)	Recall (%)	AP (%)	Parameters	Model Size (KB)	Fps
YOLOv5s	89.1	90.3	95.6	7,063,542	14,052	35.1
MoblieNet V2-YOLOv5s	81.2	90.3	93.6	2,917,046	5423	23.4
MoblieNet V3-YOLOv5s	94.2	85.8	95.6	3,538,532	7189	22.2
Ghost-YOLOv5s	85.4	92.3	93.4	3,897,605	8492	23.2
ShuffleNet V2-YOLOv5s	83.8	91.6	94.0	490,205	1149	35.5

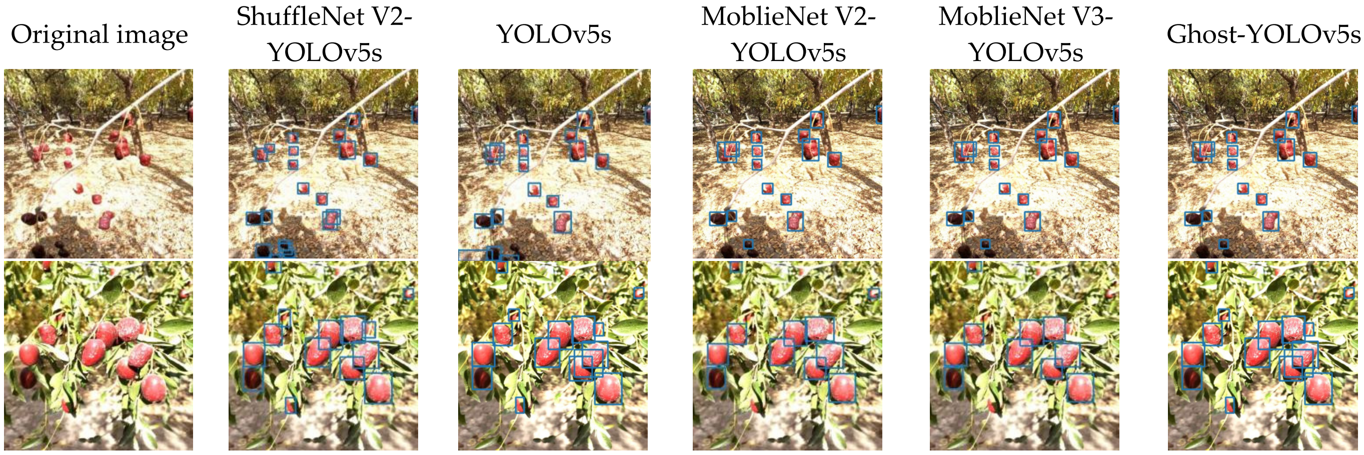


Figure 12. Test results of different lightweight backbone networks. Where the blue boxes are the test results of model test.

The Precision and AP of using ShuffleNet V2 as the backbone network were slightly lower than that of the original YOLOv5s and the improved YOLOv5s using MoblieNet V3 as the backbone network. However, using ShuffleNet V2 as the backbone network could provide a more comprehensive red jujube detection. When MoblieNet V2 and GhostNet were used as a backbone, some red jujubes were missed, as shown in Figure 12. Compared with the other four detection models, the number of parameters using ShuffleNet V2 as the backbone network was only 7.14% of YOLOv5s, and the number of parameters was obviously smaller than other networks. The detection speed using the ShuffleNet V2 backbone network model was also faster than other detection networks, as shown in Table 4. using ShuffleNet V2 as a backbone not only greatly reduced the number of model parameters but also improved the detection speed, which was more suitable for red jujubes counting and related embedded mobile devices.

3.3. Performance Comparison in Counting Jujubes Using the Different Algorithms

To verify the effectiveness of improved YOLOv5s for target detection, YOLOv3-tiny, YOLOv4-tiny, Faster R-CNN, SSD, YOLOvx-tiny, and YOLOv7-tiny were selected to compare with improved YOLOv5s. This research experimented with the selected comparison models using datasets of the same size and the same training and test sets. In order to ensure the reliability of the test, the epoch was set to 400, and the batch size was set to 4. In this research, three orchard jujube images were selected to test the yield estimation method. The comparison results are shown in Table 5. The P-R curve of the models is shown in Figure 13.

Table 5. Detection results of red jujubes with different target detection algorithms.

Model	The Number of Actual Jujube						The Number of Predicted Jujube						Precision (%)	Recall (%)	AP (%)	RMSE	MAPE (%)	Model Size (KB)
	1	2	3	4	5	6	1	2	3	4	5	6						
YOLOv5s	10	15	15	9	10	6	9	16	14	8	8	6	89.10	90.30	95.60	1.15	9.07	14,052
YOLOv4-tiny							10	15	11	9	8	6	91.60	89.40	95.90	1.83	7.78	103,012
YOLOv3-tiny							10	14	11	7	8	6	92.30	88.70	95.50	2.04	12.59	481,391
YOLOvx-tiny							8	10	11	7	7	6	86.60	91.30	95.70	3.11	22.04	19,901
YOLOv7-tiny							10	11	12	7	8	6	89.20	90.50	95.10	2.35	14.81	23,674
SSD							8	11	14	7	8	6	88.30	87.10	90.50	2.19	15.93	92,782
Faster R-CNN							9	12	13	7	7	6	64.00	89.30	87.90	2.12	15.93	110,773
Our Model							10	15	13	9	9	6	93.40	92.30	96.20	0.91	3.89	1149

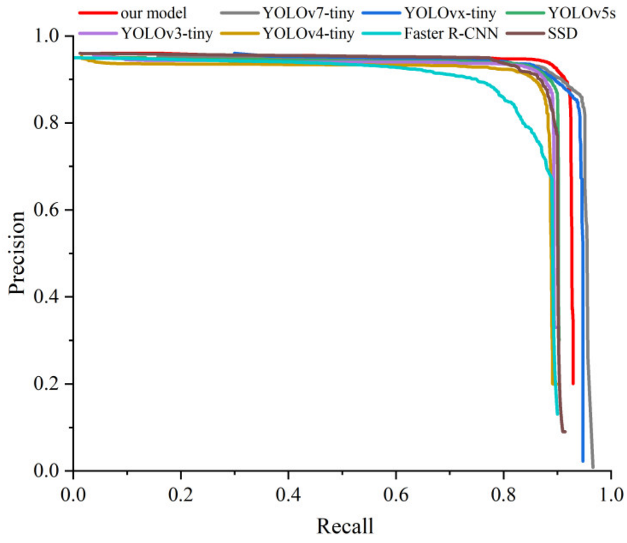


Figure 13. The PR curve of red jujubes with different target detection algorithms.

The P-R curve is a curve with recall as the horizontal coordinate and precision as the vertical coordinate out of the curve, whose area can show the comprehensive performance of the target detection model for red jujubes. Figure 13. shows that the curve areas of YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOvx-tiny and YOLOv7-tiny are larger than those of SSD and Faster R-CNN. It illustrates that the Yolo series detection networks have higher accuracy and better recognition of red jujubes. YOLOv5s is used as an improved detection network for YOLOv3-tiny and YOLOv4-tiny, but the best detection result is not obtained for red jujubes, as shown in Table 5. The YOLOv4-tiny has better detection results, but the YOLOv5s are smaller in model size and more suitable for being used in agricultural mobile devices. Compared with the classical networks, the improved network not only maintains a better detection performance but also greatly reduces the model size.

Different detection algorithms were used to count red jujubes. YOLOvx-tiny, YOLOv5s, SSD, and Faster R-CNN all showed that the counting results of red jujubes were less than the actual number, as shown in Figure 14 image1. YOLOv7-tiny, YOLOv5s, and Faster R-CNN caused repeated recognition in the process of counting red jujubes, which led to the counting results being higher than the actual number, as shown in Figure 14 image2 and image3. Error counting occurred when SSD counted red jujubes, as shown in Figure 14 image3. When counting image4, only YOLOv4-tiny and Our Model counted accurately. However, Our Model also missed the detection of red jujubes, but compared with other algorithms, the number of missed detection was less, as shown in Figure 14 image5. When counting the Shaded red jujubes, all algorithms could count effectively, as shown in Figure 14 image6.

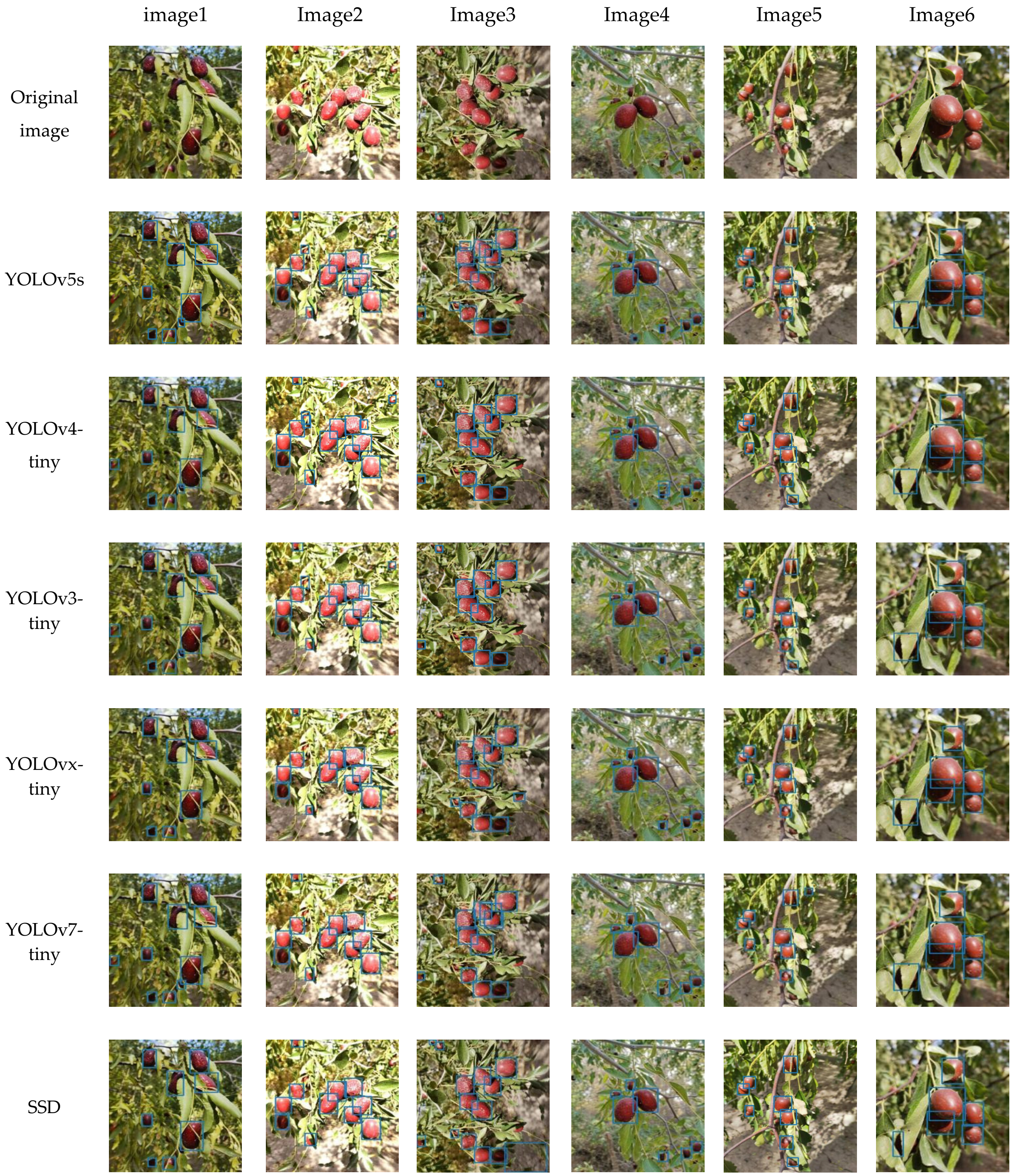


Figure 14. *Cont.*

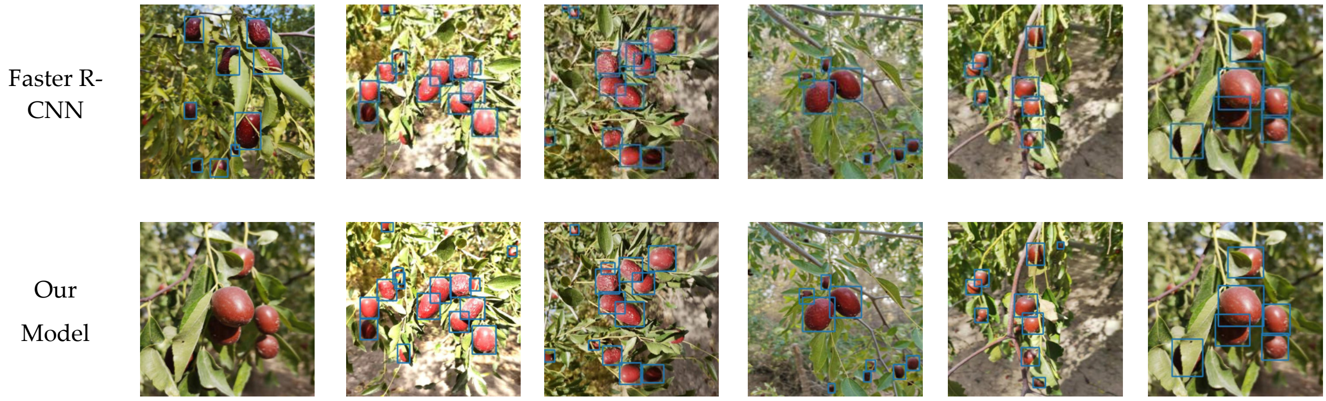


Figure 14. Test results of different algorithms. Where the blue boxes are the test results of model test.

According to the experimental results, In the detection of red jujube, YOLOv5s, YOLOv4-tiny, and Faster R-CNN all miss the detection, which leads to a decrease in the number of red jujubes. YOLOv3-tiny, SSD, and Faster R-CNN all have error recognition, which leads to the increase in the estimation error of jujube yield by the model, as shown in Figure 14. Faster R-CNN, as one of the representative networks of the two-stage detection model, has good overall detection performance for red jujubes, but the AP is lower compared with other detection networks, And RMSE and MAPE are the maximum values, as shown in Table 5. This difference is mainly manifested in the recognition difficulty of fruits with large leaves shading and poor recognition of overlapping fruits. The reason for the difference is that Faster R-CNN does not build an image feature pyramid and cannot sufficiently extract features for small targets, resulting in insensitivity to small target recognition. For both the single-stage detection model Yolo series and SSD, the overall performance is better than Faster R-CNN. Comparing SSD and YOLOv5s, the Precision is reduced by 0.80%. The recall is reduced by 3.20%, and AP is reduced by 5.10%, RMSE is increased by 45.75%, MAPE is increased by 6.86%. The main reasons are: (1) Since YOLOv5s introduces the FPN + PAN, while the detection layer is fused by three levels of feature layers, while all six feature pyramid layers of SSD come from the last layer of FCN, YOLOv5s is better than SSD in detecting red jujubes. (2) Due to the limited number of red jujube and the severe occlusion between red jujubes, it is difficult for the model to learn the various states. Compared with YOLOvx-tiny and YOLOv7-tiny, the AP of the improved network increased by 0.50% and 1.10%, respectively, RMSE decreased by 2.2 and 1.44 respectively, and MAPE decreased by 18.15% and 10.92% respectively. Comparing YOLOv5s, we introduce the ShuffleNet V2 backbone to reduce the size of the model, but the feature extraction ability of the model is limited. The idea of resizing images by convolution layer was adopted, and the Stem was added to enhance the feature extraction ability of the network. The improved model overall outperforms YOLOv5s, with Precision, Recall, and AP improving by 4.3%, 2.0%, and 0.6%. In addition, the model size, RMSE, and MAPE decreased by 91.82%, 20.87%, and 5.18%, respectively. The improved model has the highest Precision, Recall, F1-Score, and AP, and the smallest in model size, RMSE, and MAPE among the comparison networks.

4. Conclusions

In this research, a counting method of red jujube based on improved YOLOv5s was proposed for achieving accurate detection and counting red jujubes while reducing the model size in a complex environment. In order to reduce the number of parameters, ShuffleNet V2 was used as the backbone to make the model lightweight. In addition, the Stem module was designed as an intermediate module between the input and backbone to prevent the information loss caused by the change in feature map size. PANet was replaced by BiFPN for multi-scale feature fusion to enhance the model feature fusion capability and improve the model accuracy. Finally, the improved YOLOv5s detection model was used to count red jujubes. In order to verify the efficiency of the proposed model, YOLOv5s, YOLOv3-tiny, YOLOv4-tiny, SSD, Faster R-CNN, YOLOvx-tiny, and YOLOv7-tiny were used to compare with the improved model. The results showed that the improved model not only greatly reduced the model size but also had better performance in detection results than the comparison networks. Compared with yolov5s, Precision, Recall, and AP are improved by 4.3%, 2%, and 0.6%, respectively. In addition, the model size, RMSE, and MAPE decreased by 91.82%, 42.21%, and 11.47%, respectively. Therefore, the improved YOLOv5s model can not only effectively improve the detection performance of red jujubes but also finish the task of counting red jujubes in agricultural production. The method can provide a basis for estimating the yield of jujube by vision.

In summary, a counting method of red jujube based on improved YOLOv5s was proposed in this research, and the counting effectiveness of the method was verified by experiments. The future work of the red jujube counting method is as follows:

(1) Expand the types of data sets and increase the robustness of the model. There are only two kinds of jujube in the data set used in this research, so it is necessary to add more kinds of jujube fruit data to enhance the robustness of the model.
(2) Construct the model of jujube fruit size and quality. Further, the counting method of red jujubes was used to accurately estimate the yield of red jujubes.

Author Contributions: Data curation, methodology, project administration, writing—original draft, writing—review and editing, Y.Q.; review & editing, supervision, funding acquisition, and project administration, Y.H.; data curation, Z.Z.; formal analysis, H.Y.; formal analysis, K.Z.; review & editing, supervision, funding acquisition, and project administration, J.H. review & editing, supervision, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: Please add: This research was supported by the Talent start-up Project of Zhejiang A&F University Scientific Research Development Foundation (2021LFR066) and the National Natural Science Foundation of China (C0043619, C0043628).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dicianu, D.E.; Butcaru, A.C.; Constantin, C.G.; Dobrin, A.; Stanica, F. Evaluation of some nutritional properties of Chinese jujube (*Zizyphus jujuba* Mill.) fruit organicaly produced in bucharest area. *Sci. Pap. Ser. B Hortic.* **2020**, *64*, 79–84.
2. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
3. Muruganantham, P.; Wibowo, S.; Grandhi, S.; Samrat, N.H.; Islam, N. A Systematic Literature Review on Crop Yield Prediction with Deep Learning and Remote Sensing. *Remote Sens.* **2022**, *14*, 1990. [CrossRef]
4. Dorj, U.-O.; Malrey, L.; Sang-seok, Y. An yield estimation in citrus orchards via fruit detection and counting using image processing. *Comput. Electron. Agric.* **2017**, *140*, 103–112. [CrossRef]
5. Wang, Z.; Kerry, W.; Anand, K. Mango fruit load estimation using a video based MangoYOLO—Kalman filter—Hungarian algorithm method. *Sensors* **2019**, *19*, 2742. [CrossRef]
6. Lyu, S.; Li, R.; Zhao, Y.; Li, Z.; Fan, R.; Liu, S. Green Citrus Detection and Counting in Orchards Based on YOLOv5-CS and AI Edge System. *Sensors* **2022**, *22*, 576. [CrossRef]
7. Zhang, Y.; Zhang, W.; Yu, J.; He, L.; Chen, J.; He, Y. Complete and accurate holly fruits counting using YOLOX object detection. *Comput. Electron. Agric.* **2022**, *198*, 107062. [CrossRef]
8. Li, X.; Du, Y.; Yao, L.; Wu, J.; Liu, L. Design and Experiment of a Broken Corn Kernel Detection Device Based on the YOLOv4-Tiny Algorithm. *Agriculture* **2021**, *11*, 1238. [CrossRef]
9. Gu, Y.; Wang, S.; Yan, Y.; Tang, S.; Zhao, S. Identification and Analysis of Emergency Behavior of Cage-Reared Laying Ducks Based on YOLOv5. *Agriculture* **2022**, *12*, 485. [CrossRef]
10. Zheng, Z.; Yang, H.; Zhou, L.; Yu, B.; Zhang, Y. HLU 2-Net: A Residual U-Structure Embedded U-Net With Hybrid Loss for Tire Defect Inspection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11.

11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
12. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
15. Shen, Z.; Liu, Z.; Li, J.; Jiang, Y.-G.; Chen, Y.; Xue, X. Dsod: Learning deeply supervised object detectors from scratch. In Proceedings of the IEEE International Conference on Computer Vision 2017, Venice, Italy, 22–29 October 2017; pp. 1919–1927.
16. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
19. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* **2020**, *11*, 510. [CrossRef] [PubMed]
21. You, L.; Jiang, H.; Hu, J.; Chang, C.; Chen, L.; Cui, X.; Zhao, M. GPU-accelerated Faster Mean Shift with euclidean distance metrics. *arXiv* **2021**, arXiv:2112.13891.
22. Zhao, M.; Jha, A.; Liu, Q.; Millis, B.A.; Mahadevan-Jansen, A.; Lu, L.; Landman, B.A.; Tyskac, M.J.; Huo, Y. Faster mean-shift: Gpu-accelerated embedding-clustering for cell segmentation and tracking. *arXiv* **2020**, arXiv:2007.14283. [CrossRef]
23. Zhao, M.; Liu, Q.; Jha, A.; Deng, R.; Yao, T.; Mahadevan-Jansen, A.; Tyska, M.J.; Millis, B.A.; Huo, Y. VoxelEmbed: 3D instance segmentation and tracking with voxel embedding based deep learning. In Proceedings of the International Workshop on Machine Learning in Medical Imaging, Strasbourg, France, 27 September 2021; pp. 437–446.
24. Lu, Y.; Young, S. A survey of public datasets for computer vision tasks in precision agriculture. *Comput. Electron. Agric.* **2020**, *178*, 105760. [CrossRef]
25. Mulyono, I.; Lukita, T.; Sari, C.; Setiadi, D.; Rachmawanto, E.; Susanto, A.; Putra, M.; Santoso, D. Parijoto Fruits Classification using K-Nearest Neighbor Based on Gray Level Co-Occurrence Matrix Texture Extraction. *J. Phys. Conf. Ser.* **2020**, *1051*, 012017. [CrossRef]
26. Fauliah, S.P. Implementation of learning vector quantization (lvq) algorithm for durian fruit classification using gray level co-occurrence matrix (glcm) parameters. *J. Phys. Conf. Ser.* **2019**, *1196*, 012040.
27. Zhao, C.; Lee, W.S.; He, D. Immature green citrus detection based on colour feature and sum of absolute transformed difference (SATD) using colour images in the citrus grove. *Comput. Electron. Agric.* **2016**, *124*, 243–253. [CrossRef]
28. Peng, H.; Shao, Y.; Chen, K.; Deng, Y.; Xue, C. Research on multi-class fruits recognition based on machine vision and SVM. *IFAC-PapersOnLine* **2018**, *51*, 817–821. [CrossRef]
29. Wajid, A.; Singh, N.K.; Junjun, P.; Mughal, M.A. Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) 2018, Sukkur, Pakistan, 3–4 March 2018; pp. 1–4.
30. Hussin, R.; Juhari, M.R.; Kang, N.W.; Ismail, R.; Kamarudin, A. Digital image processing techniques for object detection from complex background image. *Procedia Eng.* **2012**, *41*, 340–344. [CrossRef]
31. Zhang, F.; Chen, Z.; Bao, R.; Zhang, C.; Wang, Z. Recognition of dense cherry tomatoes based on improved YOLOv4-LITE lightweight neural network. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 270–278.
32. Fu, L.; Feng, Y.; Majeed, Y.; Zhang, X.; Zhang, J.; Karkee, M.; Zhang, Q. Kiwifruit detection in field images using Faster R-CNN with ZFNet. *IFAC-PapersOnLine* **2018**, *51*, 45–50. [CrossRef]
33. Liu, Z.; Wu, J.; Fu, L.; Majeed, Y.; Feng, Y.; Li, R.; Cui, Y. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE Access* **2019**, *8*, 2327–2336. [CrossRef]
34. Wang, Y.; Xue, J. Lightweight object detection method for Lingwu long jujube images based on improved SSD. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 173–182.
35. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved YOLOv4-tiny model. *Comput. Electron. Agric.* **2021**, *191*, 106503. [CrossRef]
36. Novtahaning, D.; Shah, H.A.; Kang, J.-M. Deep Learning Ensemble-Based Automated and High-Performing Recognition of Coffee Leaf Disease. *Agriculture* **2022**, *12*, 1909. [CrossRef]
37. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
38. Wu, S.; Yang, J.; Wang, X.; Li, X. Iou-balanced loss functions for single-stage object detection. *Pattern Recognit. Lett.* **2022**, *156*, 96–103. [CrossRef]

39. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
40. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
41. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef]
42. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
43. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.