remote sensing

MDPI

Article

A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5

Bin Yan [1,2], Pan Fan [1,2], Xiaoyan Lei [1,2], Zhijie Liu [1,3] and Fuzeng Yang [1,3,4,*]

[1] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China; yanbin@nwafu.edu.cn (B.Y.); fanpan@nwafu.edu.cn (P.F.); lxy950429@nwafu.edu.cn (X.L.); liuzhijie@nwsuaf.edu.cn (Z.L.)
[2] Shannxi Key Laboratory of Apple, Yangling 712100, China
[3] Apple Full Mechanized Scientific Research Base of Ministry of Agriculture and Rural Affairs, Yangling 712100, China
[4] State Key Laboratory of Soil Erosion and Dryland Farming on Loess Plateau, Yangling 712100, China
* Correspondence: yangfzkm@nwafu.edu.cn

Abstract: The apple target recognition algorithm is one of the core technologies of the apple picking robot. However, most of the existing apple detection algorithms cannot distinguish between the apples that are occluded by tree branches and occluded by other apples. The apples, grasping end-effector and mechanical picking arm of the robot are very likely to be damaged if the algorithm is directly applied to the picking robot. Based on this practical problem, in order to automatically recognize the graspable and ungraspable apples in an apple tree image, a light-weight apple targets detection method was proposed for picking robot using improved YOLOv5s. Firstly, BottleneckCSP module was improved designed to BottleneckCSP-2 module which was used to replace the BottleneckCSP module in backbone architecture of original YOLOv5s network. Secondly, SE module, which belonged to the visual attention mechanism network, was inserted to the proposed improved backbone network. Thirdly, the bonding fusion mode of feature maps, which were inputs to the target detection layer of medium size in the original YOLOv5s network, were improved. Finally, the initial anchor box size of the original network was improved. The experimental results indicated that the graspable apples, which were unoccluded or only occluded by tree leaves, and the ungraspable apples, which were occluded by tree branches or occluded by other fruits, could be identified effectively using the proposed improved network model in this study. Specifically, the recognition recall, precision, mAP and F1 were 91.48%, 83.83%, 86.75% and 87.49%, respectively. The average recognition time was 0.015 s per image. Contrasted with original YOLOv5s, YOLOv3, YOLOv4 and EfficientDet-D0 model, the mAP of the proposed improved YOLOv5s model increased by 5.05%, 14.95%, 4.74% and 6.75% respectively, the size of the model compressed by 9.29%, 94.6%, 94.8% and 15.3% respectively. The average recognition speeds per image of the proposed improved YOLOv5s model were 2.53, 1.13 and 3.53 times of EfficientDet-D0, YOLOv4 and YOLOv3 and model, respectively. The proposed method can provide technical support for the real-time accurate detection of multiple fruit targets for the apple picking robot.

Keywords: artificial intelligence; convolutional neural network; YOLOv5; object detection; apple picking robot; lightweight; real-time detection

Citation: Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. https://doi.org/10.3390/rs13091619

Academic Editor: Gemine Vivone

Received: 20 March 2021
Accepted: 19 April 2021
Published: 21 April 2021

1. Introduction

Artificial apple picking is a labor-intensive and time-intensive task. Therefore, in order to realize the efficient and automatic picking of apples, to ensure timely harvest of mature fruits, and improve the competitiveness of the apple market, further study of the key technologies of the apple picking robot is essential [1,2]. The intelligent perception and acquisition of apple information is one of the most critical technologies for the apple picking robot, which belongs to the information perception of the front-end part of the

robot. Therefore, to improve the apple picking efficiency of the robot, it is necessary to realize the rapid and accurate identification of apple targets on the tree.

A schematic diagram of the practical harvesting situation that the picking robot confronts in an apple orchard is shown in Figure 1. The robot can realize the picking of apples that are unoccluded or only occluded by leaves. However, the apples which are occluded by branches or occluded by other fruits cannot be harvested by the picking robot, due to the fact that the apples, grasping end-effector and mechanical picking arm of the robot are very likely to be damaged if directly picking apples in the above situations without accurate recognition, resulting in the failure of picking operation. Therefore, it is essential for the picking robot to automatically recognize the apple targets which are graspable or ungraspable.

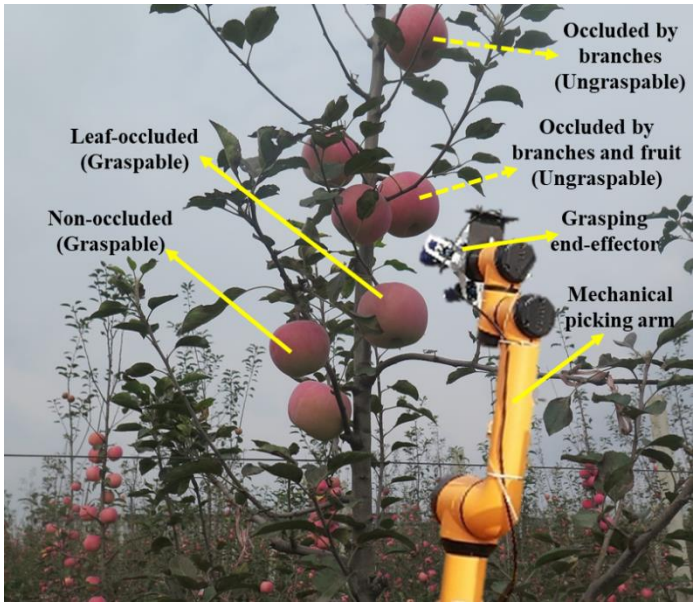


Figure 1. Schematic diagram of practical harvesting situation that picking robot confronts.

However, the existing apple targets recognition algorithms basically focused on the identification of apples in the complex orchard environment (leaf occlusion, branch occlusion, fruit occlusion and mixed occlusion etc.), and recognizes apples in different conditions as one class. Therefore, they are not suitable for application by picking robots. The recognition algorithm for judging whether the apples can be harvested by picking robot has not yet been studied.

With the development of artificial intelligence, in recent years, the artificial neural network has been widely applied in many research fields. For example, in the field of economy, stacking and deep neural network models are deployed separately on feature engineered and bootstrapped samples for estimating trends in prices of underlying stocks during pre- and post-COVID-19 periods [3]; the grey relational analysis (GRA) and artificial neural network models were utilized for the prediction of consumer exchange-traded funds (ETFs) [4]. In the field of industry, a relatively simple fuzzy logic-based solution for networked control system was proposed using related ideas of neural network [5]; a model for the prediction of maximum energy generated photovoltaic modules based on fuzzy logic principles and artificial neural networks was developed [6]; the intelligence of the flexible manufacturing system (FMS) was improved by combining Petri Net [7] and the artificial neural network [8]. In the field of agriculture, a new deep learning architecture called VddNet (Vine Disease Detection Network) was proposed for the detection of vine disease [9]; conifer seedlings in drone imagery were automated, detected using a neural network [10]; the early blight disease was identified in real-time for potato production systems, using machine vision in combination with deep learning [11].

When given sufficient data, the deep learning algorithm can generate and extrapolate new features without having to be explicitly told which features should be utilized and how they can be extracted [12–14]. CNNs (convolutional neural networks) are another variety of algorithm belonging to deep learning technology, which can provide insights into image-related datasets that we have not yet understood, achieving identification accuracies that sometimes surpass the human-level performance [15–17]. One of the most important characteristics of utilizing CNNs in object detection is that the CNN can obtain essential features by itself. Furthermore, it can build and use more abstract concepts [18].

Up till now, there have been many studies in the aspect of apple targets recognition using deep learning technology. Many convolutional neural networks, such as YOLOv2 [19], YOLOv3 [20], LedNet [21], R-FCN [22], Faster R-CNN [23–26], Mask R-CNN [27], DaSNet [28] and DaSNet-v2 [29], were successfully used in apple target recognition. The relevant study status is shown in Table 1.

Table 1. Research on apple target recognition, based on deep learning technology.

Networks Model	Precision (%)	Recall (%)	mAP (%)	F1 (%)	Average Detection Speed (s/pic)	Reference
Improved YOLOv2	—	—	90	—	0.333	[19]
Improved YOLOv3	97	90	87.71	—	0.01669	[20]
LedNet	85.3	82.1	82.6	83.4	0.028	[21]
Improved R-FCN	95.1	85.7	—	90.2	0.187	[22]
Improved Faster R-CNN	89.7	89.9	94.8	89.8	4.412	[25]
Mask R-CNN	85.7	90.6	—	88.1	—	[27]
DaSNet	—	—	83.6	83.2	0.072	[28]
DaSNet-v2	87.3	86.8	88	87.3	0.437	[29]
Faster R-CNN (VGG16)	—	—	89.3	—	0.181	[23]
Faster R-CNN (VGG16)	—	—	87.9	—	0.241	[24]
Faster R-CNN (VGG19)	—	—	82.4	86	0.45	[26]

However, no research work has been reported on the light-weight apple targets recognition algorithm that classifies apples into two categories: graspable (not occluded or only occluded by leaves) and ungraspable (other conditions).

On the other hand, throughout the studies of apple targets recognition based on deep learning, although the recognition accuracy of most existing apple detection models was high, the real-time performance of many of them were insufficient, due to its high complexity, large number of parameters and large size. Therefore, it is essential to design a light-weight apple target detection algorithm, while ensuring the accuracy of fruit recognition, to satisfy the requirements of picking robot for real-time recognition.

In the study, the apple tree fruit was used as the research object. A light-weight apple targets real-time recognition algorithm based on improved YOLOv5s for picking robot was proposed, which can realize the automatic recognition of the apples that can be grasped by picking robot and ungraspable in an apple tree image. The proposed method can provide technical support for real-time accurate detection of multiple fruit targets for the apple picking robot.

2. Materials and Methods

2.1. Apple Images Acquisition

2.1.1. Materials and Image Data Acquisition Methods

In the study, fruits in Fuji apple trees of fusiform cultivation mode in a modern standard orchard were used as research object, and the original images of apple trees from

the standardized modern orchard in Agricultural Science and Technology Experimental Demonstration Base of Qian County in Shaanxi Province and the Apple Experimental Station of Northwest A&F University in Baishui county of Shaanxi Province were collected. In fusiform cultivation mode, the row spacing of apple trees is about 4 m. The plant spacing is about 1.2 m, and the tree height is about 3.5 m, which is suitable for apple picking robot to operate in the orchard. The images of apple trees were obtained on sunny and cloudy days. The shooting phase included morning, noon and afternoon. The images were captured by Canon Powershot G16 camera, with a variety of angles selected for image acquisition at different shooting distances (0.5–1.5 m), and in total, 1214 apple images were obtained, including the following conditions: apples occluded by leaves, apples occluded by branches, mixed occlusion, overlap between apples, natural light angle, backlight angle, and side light angle, etc. (Figure 2). Furthermore, the environment factors such as cloudy, shadows, high light, low light, reflections were also considered in image capture. The resolution of the captured images is 4000 × 3000 pixels, and the format is JPEG.

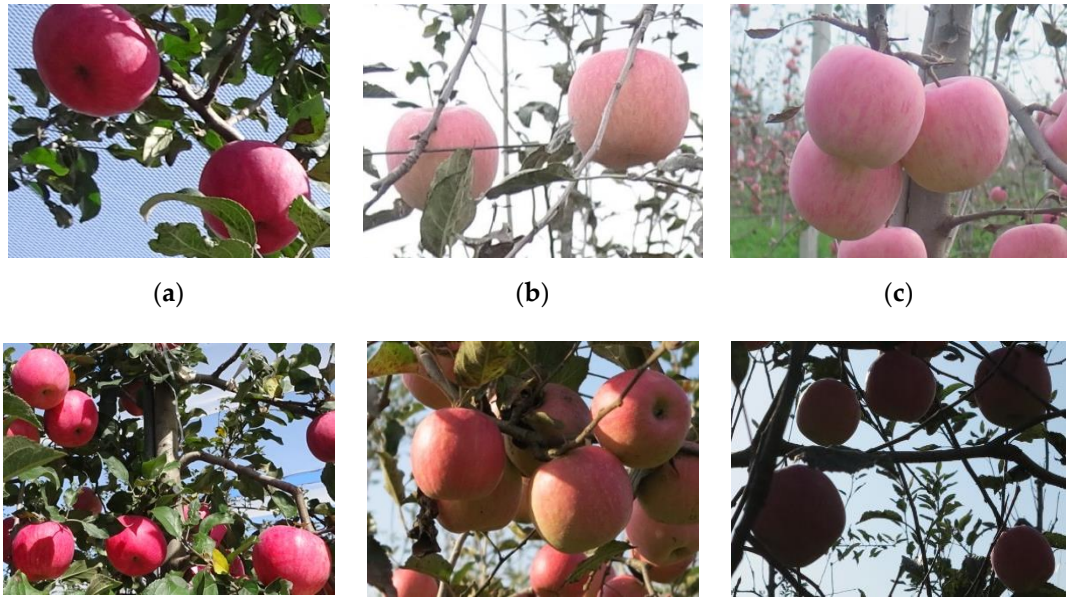


Figure 2. Apple images in different conditions. (**a**) Apples occluded by leaves (**b**) Apples occluded by branches (**c**) Overlapped apples. (**d**) Frontlight angle (**e**) Sidelight angle (**f**) Backlight angle.

2.1.2. Preprocessing of Images

The generation of the target detection model based on deep learning is realized on the basis of the training of a large number of image data, therefore, the augmentation of 1214 collected apple images is necessary.

Firstly, 200 images (100 of sunny days and 100 of cloudy days) were randomly selected from 1214 images as the test set, and the rest of the 1014 images were utilized as the training set. The detail of distribution for the image samples in test set is shown in Table 2. Secondly, in order to improve the training efficiency of apple targets recognition model, the original 1014 images of training set were compressed, and the length and width of them were compressed to 1/5 of the original one. Thirdly, the image data annotation software called 'LabelImg' was used to draw the outer rectangular boxes of the apple targets in the compressed apple tree images to realize the manual annotation of the fruits. Images were labeled based on the smallest surrounding rectangle of apples, to ensure the rectangle contains background area as little as possible. Among them, the apples in the image that were unoccluded or only occluded by leaves were labeled as 'graspable' class, and the apples in other conditions were labeled as 'ungraspable' class. The XML format files were generated after the annotation were saved. Finally, in order to enrich the image data of the

training set, data enhancement processing was carried out to the data set to better extract the features of apples belonging to different labeled categories and avoid the over-fitting of the model obtained from training.

Table 2. Detailed information of images in test set.

Test Set	Sunny	Cloudy	Total
Number of images	100	100	200
Graspable apple	482	525	1007
Ungraspable apple	766	563	1329

Due to the uncertain factors, such as illumination angle and weather, resulting in the light environment of image acquisition is extremely complex; in order to improve the generalization ability of apple targets detection model, several image enhancement methods were utilized for the 1014 images of training set respectively based on MATLAB (version 2016, the MathWorks Inc., Natick, MA, USA) software and its related image processing functions. The image enhancement methods include image brightness enhancement and reduction, horizontal mirroring, vertical mirroring, multi-angle rotation (90°, 180°, 270°) etc. In addition, considering the noise generated by the image acquisition equipment in the process of image acquisition and the blur of the captured images caused by the shaking of the equipment or the branches, Gaussian noise with variance of 0.02 was added to the images, and the motion blur processing was carried out. Detailed procedures of image enhancement methods are illustrated in the following.

Image brightness enhancement and reduction: Firstly, the original image is converted to HSV space by using 'rgb2hsv' function; secondly, the V component (brightness component) of the image is multiplied by different coefficients; finally, the synthesized HSV space image is converted to RGB space by using 'hsv2rgb' function, realizing the brightness enhancement and reduction of the image. In the study, three brightness intensities can be generated utilizing brightness enhancement, including (H + S + 1.2 × V), (H + S + 1.4 × V) and (H + S + 1.6 × V); two brightness intensities can be generated using brightness reduction, including (H + S + 0.6 × V) and (H + S + 0.8 × V).

Image mirroring (horizontal and vertical mirror) was implemented using the Matlab function 'imwarp'. The horizontal mirroring was implemented by transforming the left and right sides of the image centering on the vertical line of the image. The vertical mirroring was implemented by transforming the upper and lower sides of the image centering on the horizontal centerline of the image.

For image rotation, the Matlab function 'imrotate' was used to rotate the raw image, and 90°, 180°, and 270° of rotation were achieved by changing the function parameter 'angle', respectively. The transformed images can improve the detection performance of the model by correctly identifying the apples of different orientations.

Four kinds of motion blur processing were employed to make the convolutional network model have strong adaptability with the blurred images. A predetermined two-dimensional filter was created using the Matlab function 'fspecial'. LEN (length, represents pixels of linear motion of camera) and THETA (θ, represents the angular degree in a counter-clockwise direction) of the motion filter were set as (6, 30), (6, −30), (7, 45) and (7, −45), respectively. Then, the Matlab function 'imfilter' was used to blur the image with the generated filter.

Furthermore, the addition of Gaussian noise with variance of 0.02 to the raw images was implemented using Matlab function 'imnoise'.

The final training sets consist of 16,224 images used as the final training set data for training of apple targets recognition model, including 15,210 enhanced images and 1014 raw images. The detailed distribution of training set data is shown in Figure 3. There was no overlap between the training set and the test set.

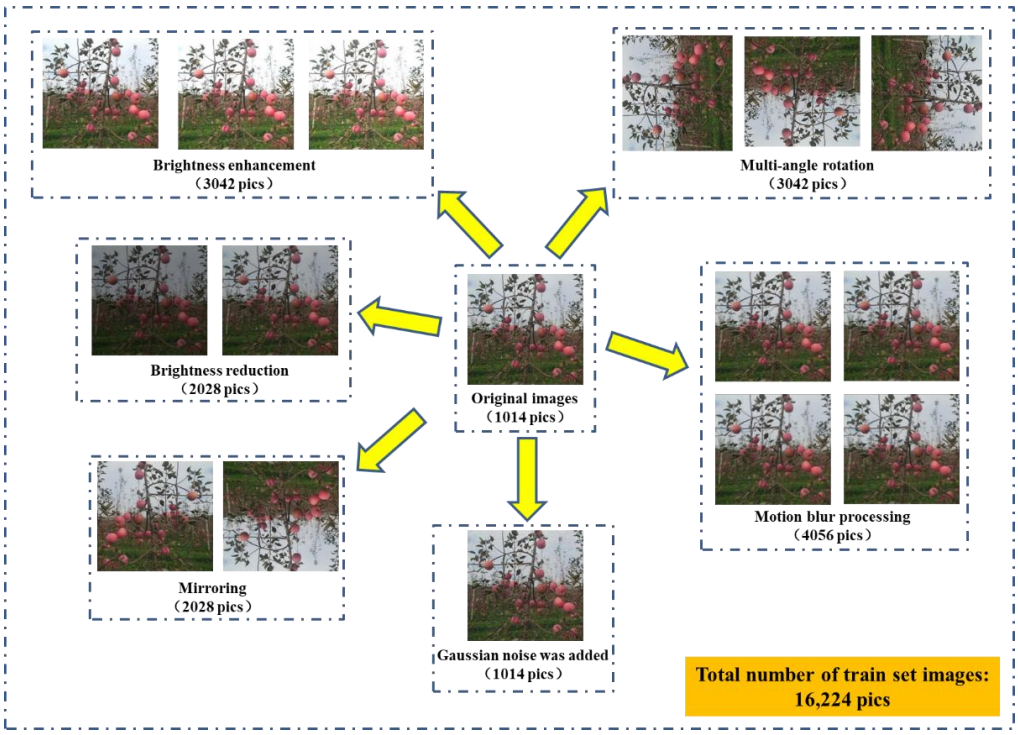


Figure 3. The distribution of training set data.

2.2. Improvment of YOLOv5s Network Architecture

2.2.1. YOLOv5s Network Architecture

YOLOv5 network [30,31] is the latest product of the YOLO architecture series. The detection accuracy of this network model is high, and the inference speed is fast, with the fastest detection speed being up of 140 frames per second. On the other hand, the size of the weight file of YOLOv5 target detection network model is small, which is nearly 90% smaller than YOLOv4, indicating that YOLOv5 model is suitable for deployment to the embedded devices to implement real-time detection. Therefore, the advantages of YOLOv5 [31] network are its high detection accuracy, lightweight characteristics, and fast detection speed at the same time.

Since the accuracy, real-time performance and lightweight aspect of the fruit detection model are essential to the accuracy and efficiency of the fruit targets recognition for apple picking robot, this study intends to improve the fruit targets recognition network for the apple picking robot based on the YOLOv5 architecture. The YOLOv5 architecture contains four architectures, specifically named YOLOv5s [31], YOLOv5m [31], YOLOv5l [31] and YOLOv5x [31], respectively. The main difference among them is that the amount of feature extraction modules and convolution kernel in the specific location of the network is different. The size of models and the amount of model parameters in the four architectures increase in turn.

Since there were two varieties of targets to be identified in this study, and the recognition model has high requirements for real-time performance and lightweight properties. Therefore, the accuracy, efficiency and size of the recognition model were considered comprehensively in the study, and the improved design of the apple targets recognition network was carried out based on the YOLOv5s [31] architecture (Figure 4).

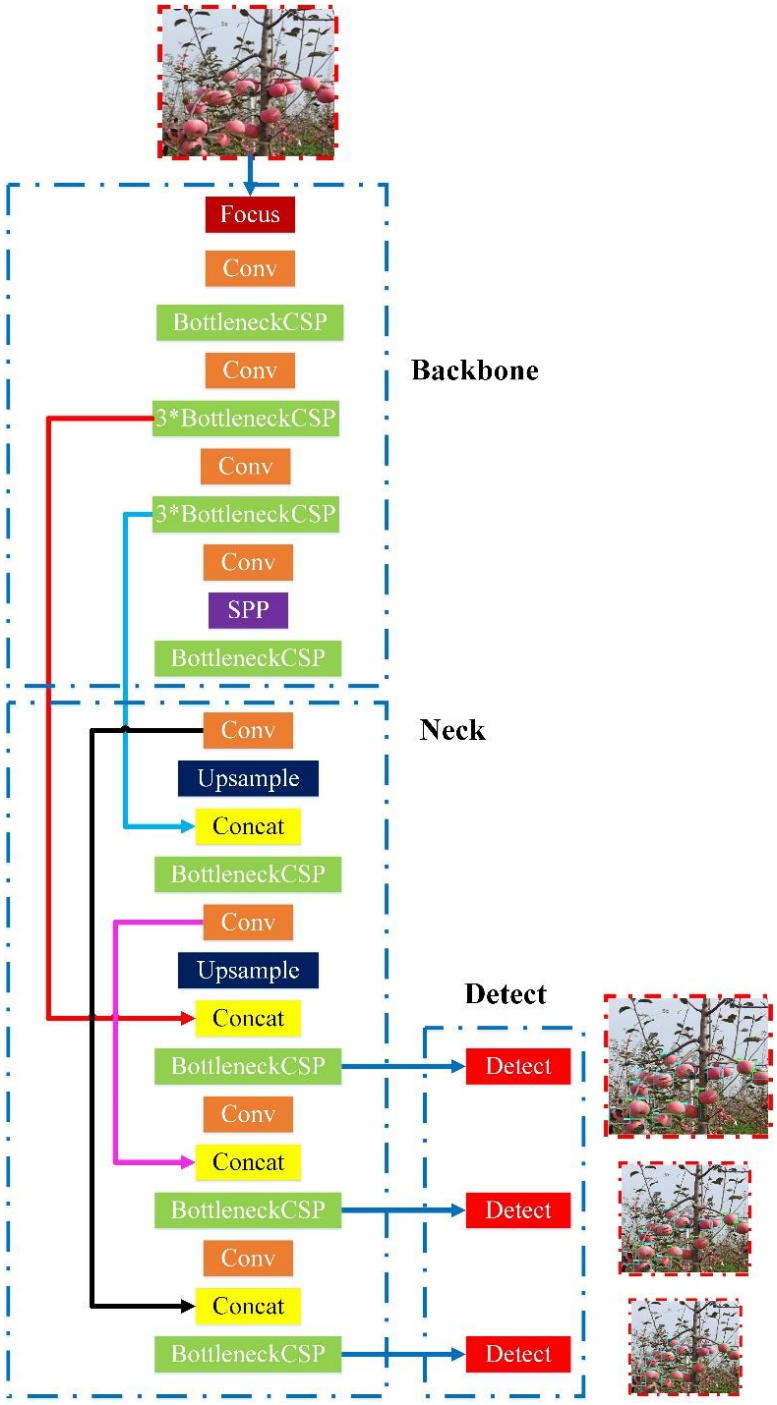


Figure 4. Architecture of original YOLOv5s network.

The YOLOv5s [31] framework mainly consists of three components, including: backbone network, neck network and detect network. Backbone network is a convolutional neural network that aggregates different fine-grained images and forms image features. Specifically, the first layer of the backbone network is the focus module (Figure 5), which is designed to reduce the calculation of the model and accelerate the training speed. Its functions are as follows: Firstly, the input 3 channel image (the default input image size of YOLOv5s [31] architecture is $3 \times 640 \times 640$) was segmented into four slices with the size of $3 \times 320 \times 320$ per slice, using a slicing operation. Secondly, concat operation was utilized to connect the four sections in depth, with the size of output feature map being $12 \times 320 \times 320$, and then through the convolutional layer composed of 32 convolution

kernels, the output feature map with a size of 32 × 320 × 320 was generated. Finally, through the BN layer (batch normalization) and the Hardswish activation functions, the results were output into the next layer.

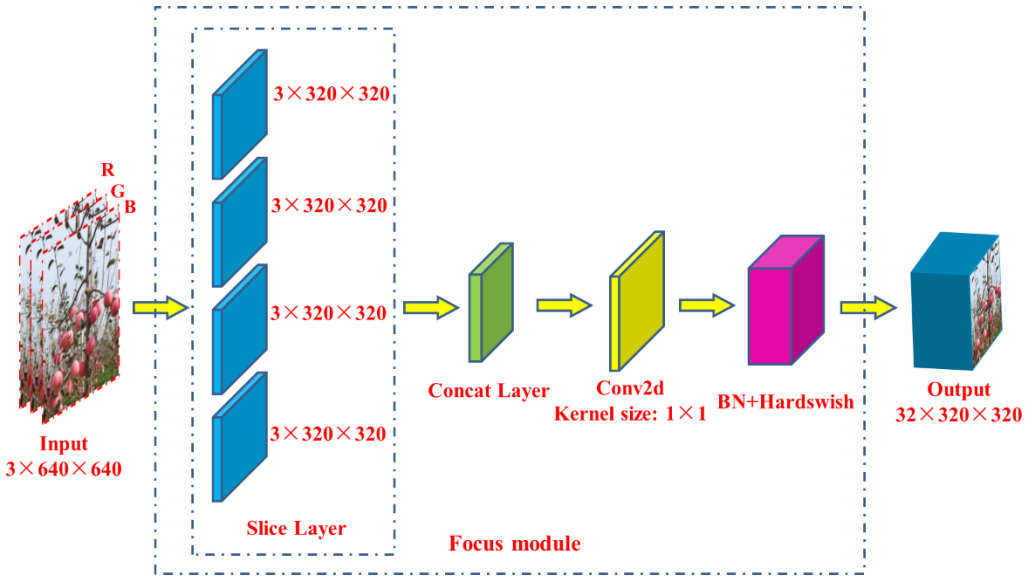


Figure 5. Structure of Focus module.

The third layer of the backbone network is the BottleneckCSP module (Figure 6), which is designed to better extract the deep features of the image. The BottleneckCSP module is mainly composed of a Bottleneck module, which is a residual network architecture that connects a convolutional layer (Conv2d + BN + Hardswish activation function) whose convolution kernel size is 1 × 1, with a convolutional layer whose convolution kernel size is 3 × 3. The final output of the Bottleneck module is the addition of the output of this part and the initial input through the residual structure.

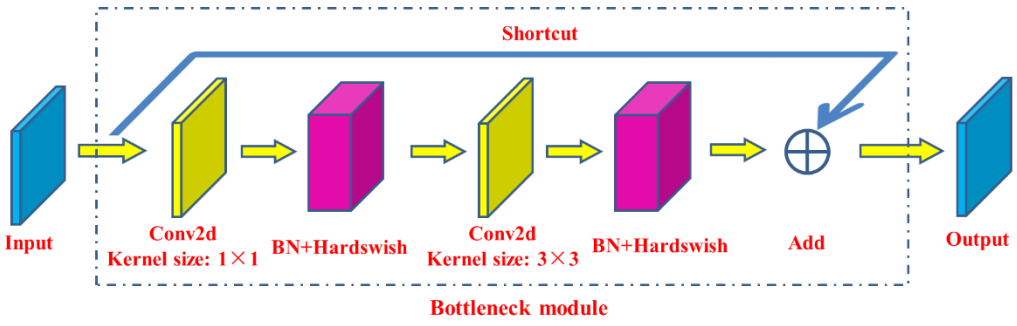


Figure 6. Structure of Bottleneck module.

The initial input of the BottleneckCSP module (Figure 7) is input into two branches, and the amount of channels of feature maps is halved through the convolution operation in two branches. Then, through the Bottleneck module and Conv2d layer in branch two, the output feature map of branch one and two is connected in depth, utilizing concat operation. Finally, the output feature map of the module was obtained after passing through the BN layer and Conv2d layer successively, and the size of this feature map is the same as that of the input of the BottleneckCSP module.

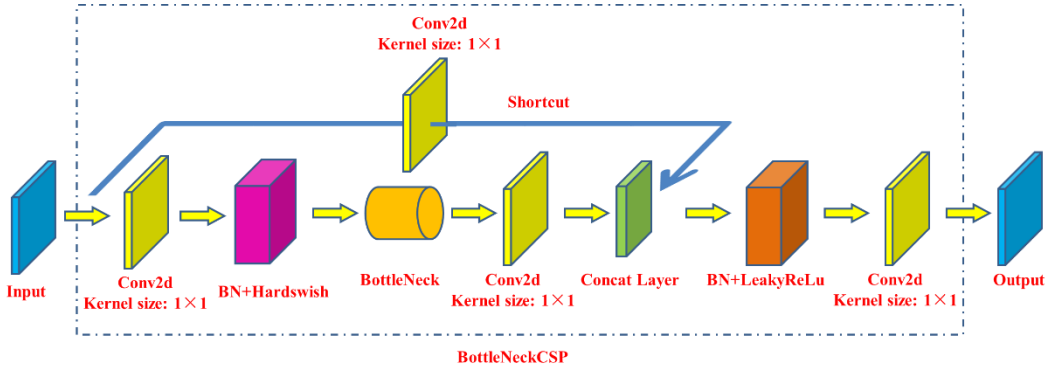


Figure 7. Structure of BottleneckCSP module.

The ninth layer of the Backbone network is SPP module (spatial pyramid pooling) (Figure 8), which is designed to improve the receptive field of the network by converting any size of feature map into a fixed-size feature vector. The size of the input feature map of the SPP module belonged to YOLOv5s [31] is 512 × 20 × 20. Firstly, the feature map with a size of 256 × 20 × 20 is output after a pass through the convolutional layer; the convolution kernel size is 1 × 1. Then, this feature map and the output feature map that are subsampled through three parallel Maxpooling layers (maximum pooling layer) are connected in depth, and the size of the output feature map is 1024 × 20 × 20. Finally, the final output feature map with a size of 512 × 20 × 20 is obtained after a pass through the convolutional layer with a 512 convolution kernel.

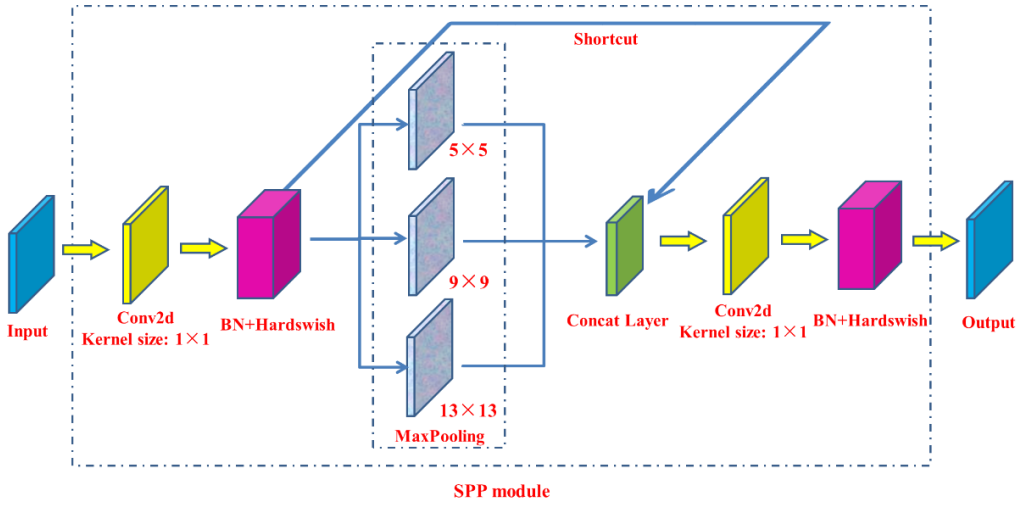


Figure 8. Structure of SPP module.

The neck network is a series of feature aggregation layers of mixed and combined image features, which is mainly used to generate FPN (feature pyramid networks), and then the output feature map is transmitted to the detect network (prediction network). Since the feature extractor of this network adopts a new FPN structure which enhances the bottom-up path, the transmission of low-level features is improved, and the detection of objects with different scales is enhanced. Therefore, the same target object with different sizes and scales can be accurately recognized.

The detect network is mainly used for the final detection part of the model, which applies anchor boxes on the feature map output from the previous layer, and outputs a vector with the category probability of the target object, the object score, and the position of the bounding box surrounding the object. The detection network of YOLOv5s [31] architecture is composed of three detect layers, whose input is a feature map with dimensions of 80 × 80, 40 × 40 and 20 × 20 respectively, used to detect the image objects of different

sizes. Each detect layer finally outputs a 21-channel vector ((2 classes + 1 class probability + 4 surrounding box position coordinates) × 3 anchor boxes), and then the predicted bounding boxes and categories of the targets in the original image were generated and labeled, implementing the detection of the apple targets in the image.

2.2.2. Improvement of Backbone Network

Since the recognition algorithm for the apple picking robot not only needs to accurately identify apple targets in a variety of situations in the complex orchard environment, the size of the model also needs to be compressed as much as possible to facilitate its deployment in hardware devices; later, the backbone network of YOLOv5s architecture was optimized and improved in this study. Under the premise of ensuring the detection accuracy, the amount of the network weight parameters and its volume were reduced, to realize the lightweight and improved design of the fruit targets recognition network for the apple picking robot.

The backbone network of the YOLOv5s architecture contains four BottleneckCSP modules, which contain multiple convolutional layers according to Section 2.2.1. Although the convolution operation can extract the features in the image, the convolution kernel contains a large number of parameters, which leads to a large number of parameters in the recognition model. Therefore, the improved design of the BottleneckCSP module was executed in this study. The convolutional layer on the bridge branch of the original module was removed, and the input feature map of the BottleneckCSP module was directly connected with the output feature map of another branch in depth, which effectively reduced the number of parameters in the module. The architecture of the improved BottleneckCSP module is shown in Figure 9, named BottleneckCSP-2. On the other hand, in order to recoup the limitation of BottleneckCSP-2, which may cause deficiency in the extraction of deep features in the image, due to its lightweight characteristics, four parts of the original backbone network where the BottleneckCSP module was used were replaced with four coterminous BottleneckCSP-2 modules in the study.

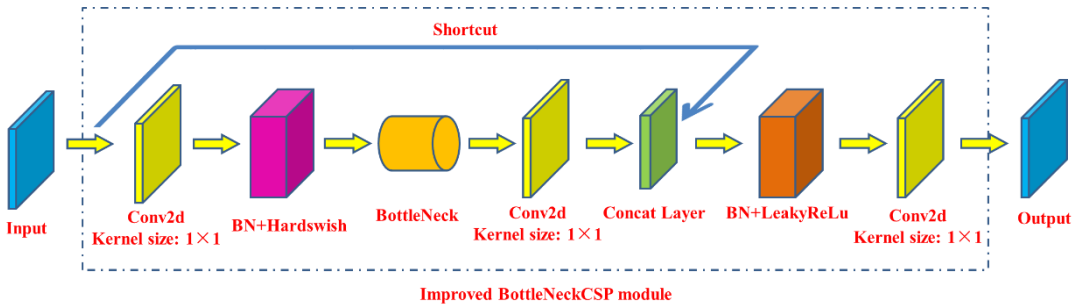


Figure 9. Structure of improved BottleneckCSP module (BottleneckCSP-2 module).

Since the shape and color of apples are different from the background objects in the image, to improve the detection accuracy of apple targets, the attention mechanism [32] in the machine vision was utilized in the design of the fruit targets recognition network to better extract the features of apple targets. The SE module (squeeze and networks, SENET) [33] is a kind of visual attention mechanism network, wherein a novel feature re-calibration strategy, illustrating the importance of each feature channel, is automatically obtained through learning, and then useful features are promoted, and unimportant features are suppressed accordingly. Since the computation of this module is small, and the module can effectively improve the expression ability of the model and optimize the content learned, it is embedded in the backbone network of the improved designed YOLOv5s architecture in this study, to improve the detection accuracy of the model.

2.2.3. Improvement of Fusion Feature Layer

The fusion of feature maps of different scales is a significant way to improve the recognition performance of the target detection network. The purpose of feature fusion is to combine features extracted from images into a feature with more discriminant ability than input features. The low-level feature map has a higher resolution and contains more location and detailed information about the target object. However, due to less feature extraction through the convolutional layer, the semantics of the low-level feature map are low, and the feature map contains more noise. The semantic information of high-level feature map is rich, but the feature map resolution is low, and the perception ability of details in the image is relatively insufficient. Therefore, the effective fusion of high-level and low-level features is key to improve the detection performance of the model.

Based on the lightweight improvement design on original YOLOv5s architecture backbone in Section 2.2.2, combined with the dimensions of the feature maps output from layers of the improved network architecture, the fusion of layers 4 and 15, 6 and 11, 10 and 21 of the original YOLOv5s architecture was changed to the fusion of layers 5 and 18, 8 and 14, 13 and 24 of the network architecture designed in this study. On the other hand, after analyzing the captured apple tree image, for the whole image, most of the sizes of apple targets that needed to be recognized belonged to the medium size. Due to the output feature map of the 23rd layer of the improved designed network architecture in the study is used as the input feature map of the target detection layer for medium-sized objects, therefore, to promote the accuracy of detection for apple, make up for the loss of spatial information caused by the low resolution of high-level features, the bridge fusion (feature fusion of the 14th layer and the 18th layer) of the feature maps that input to the medium-size target detection layer in the original YOLOv5s architecture, was improved and optimized, and the output of the feature extraction layer with a larger perception field in the lower layer was fused with the output of the feature extraction layer was located before the medium-sized target detection layer. In other words, the output feature maps of the 14th layer and the 21st layer of the improved designed network are fused together. The architecture of the improved design lightweight fruit recognition network for apple picking robot is shown in Figure 10.

2.2.4. Improvement of Initial Anchor Box Size

Initial detection anchor boxes of three sizes of original YOLOv5s framework are set for each feature map of three sizes (80×80, 40×40, 20×20), which are 10×13, 16×30, 33×23; 30×61, 62×45, 59×119; 116×90, 156×198, 373×326 respectively, and the three feature maps are input to the multi-scale detection layer, utilized for the detection of small, medium and large objects, respectively. However, since the distance between apples on the trees in the distant planting row in the background and the apple picking robot is too long, those apples cannot be regarded as effective targets to be picked. In order to avoid the false recognition of small apples in the background of the image, and improve the identification accuracy of apples in the foreground, based on the analysis of the size of apples in the foreground, the size of small apples in the background of image, and the size of the image, the sizes of initial anchor boxes belonging to the target detection layer for small and medium scales in original YOLOv5s network were improved. Combined with the length-width ratios of the apple targets in the image, the length-width ratios of the initial anchor boxes were set to about 1/1, which were modified to 80×70, 75×75, 85×100; 95×110, 130×110, and 115×125 respectively, to achieve the accurate identification of apple targets.

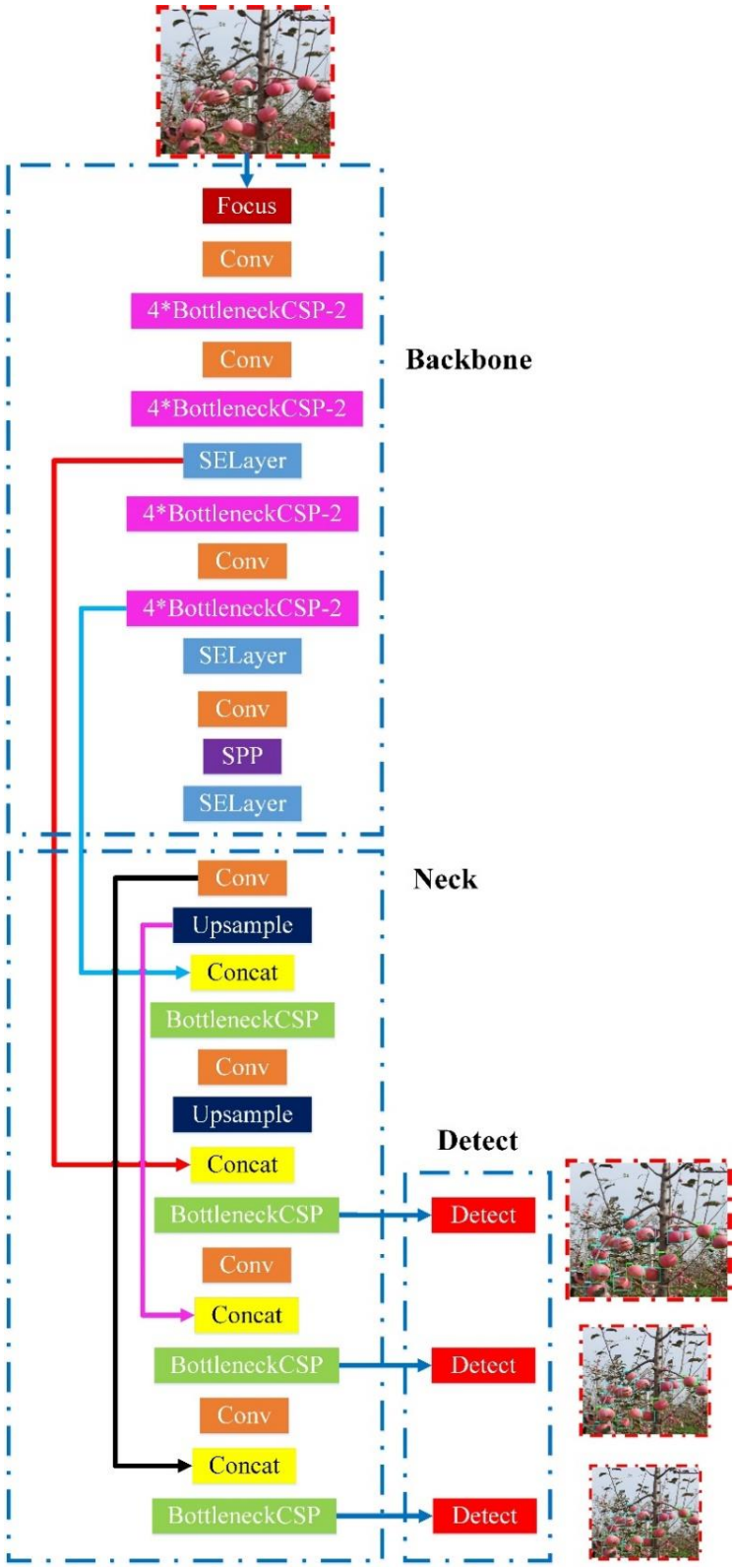


Figure 10. Architecture of improved YOLOv5s network.

2.3. Network Training

2.3.1. Training Platform

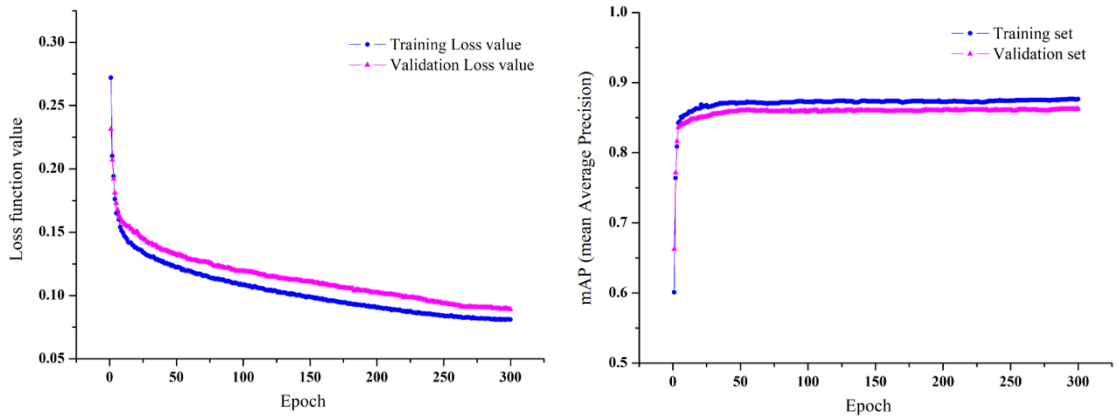
Based on the Lenovo Legion Y7000P personal computer (Intel (R) Core (TM) I7-9750H CPU, 2.6GHz, 16GB memory; NVIDIA Geforce RTX 2060 GPU, 6GB video memory), the

PyTorch deep learning framework was built under the Windows 10 operating system in the study, and Python language was utilized to write the program code and call CUDA, Cudnn, OpenCV and other required libraries, to achieve the training and testing of the fruit target recognition model for the apple picking robot.

In this study, the improved YOLOv5s network was trained by stochastic gradient descent (SGD) in an end-to-end way. The batch size of the model training was set to 4, and each time, the regularization was done by the BN layer to update the weight of model. The momentum factor (momentum) was set to 0.937, and the decay rate (decay) of weight was set to 0.0005. The initial vector and IOU (intersection over union) threshold were all set to 0.01, and the enhancement coefficient of hue (H), saturation (S) and lightness (V) were set to 0.015, 0.7 and 0.4, respectively. The number of training epochs was set to 300. After training, the weight file of the recognition model obtained was saved, and the test set was utilized to evaluate the performance of the model. The final output of the network is the location boxes of the two varieties of apple targets recognized (the prediction box of fruit location) and the probability of belonging to a specific category.

2.3.2. Training Results

The training loss and validation loss curves are shown together in Figure 11a, indicating that the loss value decreased rapidly in the first 100 epochs of network training, and basically, tends to be stable after 250 epochs of training. Therefore, the model output after 300 epochs of training was determined as the fruit target recognition model for the apple picking robot in this study. Furthermore, the training set mAP (mean average precision) and validation set mAP are shown together in Figure 11b. The above charts indicate that the model was trained well without overfitting.



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

Figure 11. Network training results. (**a**) Training and validation loss (**b**) mAP of training and validation sets.

2.4. Test and Evaluation of Model

2.4.1. Evaluation Indicators of Model

In the study, objective evaluation indicators such as Precision (1), Recall (2), mAP (mean average precision) (3) and F1 score (4) were used to evaluate the performance of the trained apple targets recognition model. The calculation equations are as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$mAP = \frac{1}{C}\sum_{K=i}^{N} P(k)\Delta R(k) \tag{3}$$

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{4}$$

where, TP means that the number of correctly identified two varieties of apple targets; FP means that the number of misidentified background as apple targets; FN represents the number of unidentified apple targets; C represents the number of apple target categories; N represents the number of IOU thresholds, K is the IOU threshold, $P(k)$ is the precision, and $R(k)$ is the recall.

2.4.2. Determination of Prediction Confidence Threshold

After the confidence of the target was obtained by the recognition model, the identified targets would be filtered using the preset threshold. The precision and recall of the detection results are different based on the same recognition model utilizing different thresholds for prediction. If the confidence threshold of the recognition model was not set appropriately, the prediction results were shown in Figure 12: The small apple targets in the distant background would be detected by mistake (labeled by yellow ellipse in Figure 12a), if the confidence threshold were set too low. The apple target in the foreground would not be recognized (labeled by yellow ellipse in Figure 12b) if the threshold were set too high. Therefore, it is essential to determine an appropriate confidence threshold for the recognition model, and then the apple targets can be recognized accurately based on the predicted confidence from the model.

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

Figure 12. Impact of confidence threshold on detection result. (**a**) Threshold is too low (**b**) Threshold is too high.

Based on the trained apple targets detection model, by adjusting the confidence threshold, the optimal prediction threshold was determined by combing with the actual demand of the fruit target recognition task for the apple picking robot, and comparing the recogni-

tion precision, recall and mAP of the detection results using different threshold values for two varieties of targets in the test data set, including 200 images. The precision, recall and mAP of the model utilizing different confidence thresholds are shown in Figure 13.

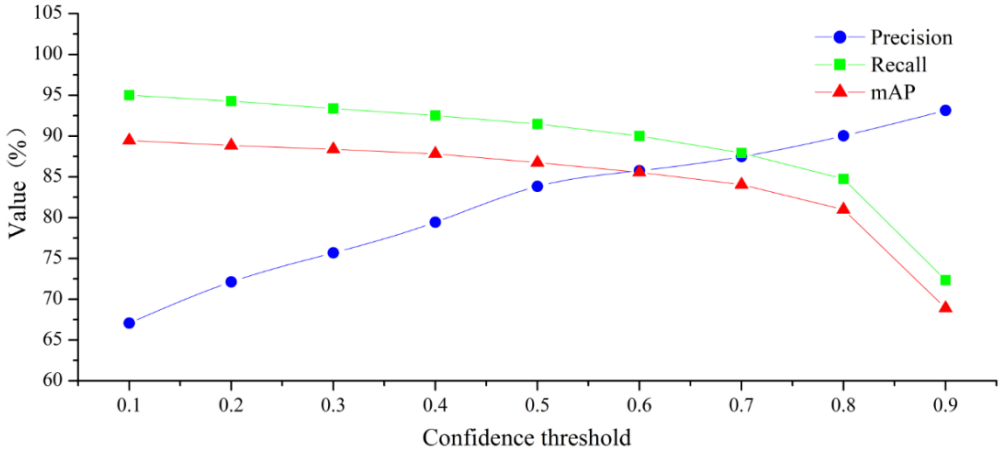


Figure 13. Changes in performance parameters of model with different confidence thresholds.

For the picking robot's apple targets recognition task, only apples in the foreground (within the scope that the grasping end-effector can pick) need to be detected. Therefore, to eliminate the interference from other small apples of the background, precision was considered first between the precision and recall of the recognition model. On the other hand, mAP, which is used to evaluate the performance of the model, should be referred to when selecting the threshold, since both precision and recall can be reflected in it.

According to the analysis of Figure 13, if the confidence threshold value was set to lower than 0.5, the precision was relatively low, less than 80%; if the confidence threshold value was set higher than 0.5, the mAP was decreased gradually. Therefore, considering the recognition precision and mAP of the model comprehensively, when the confidence threshold was set at 0.5, the performance of the model is the best, with the precision, recall, mAP being 83.83%, 91.48%, and 86.75%, respectively.

3. Results

3.1. Results and Analysis of Apple Targets Detection

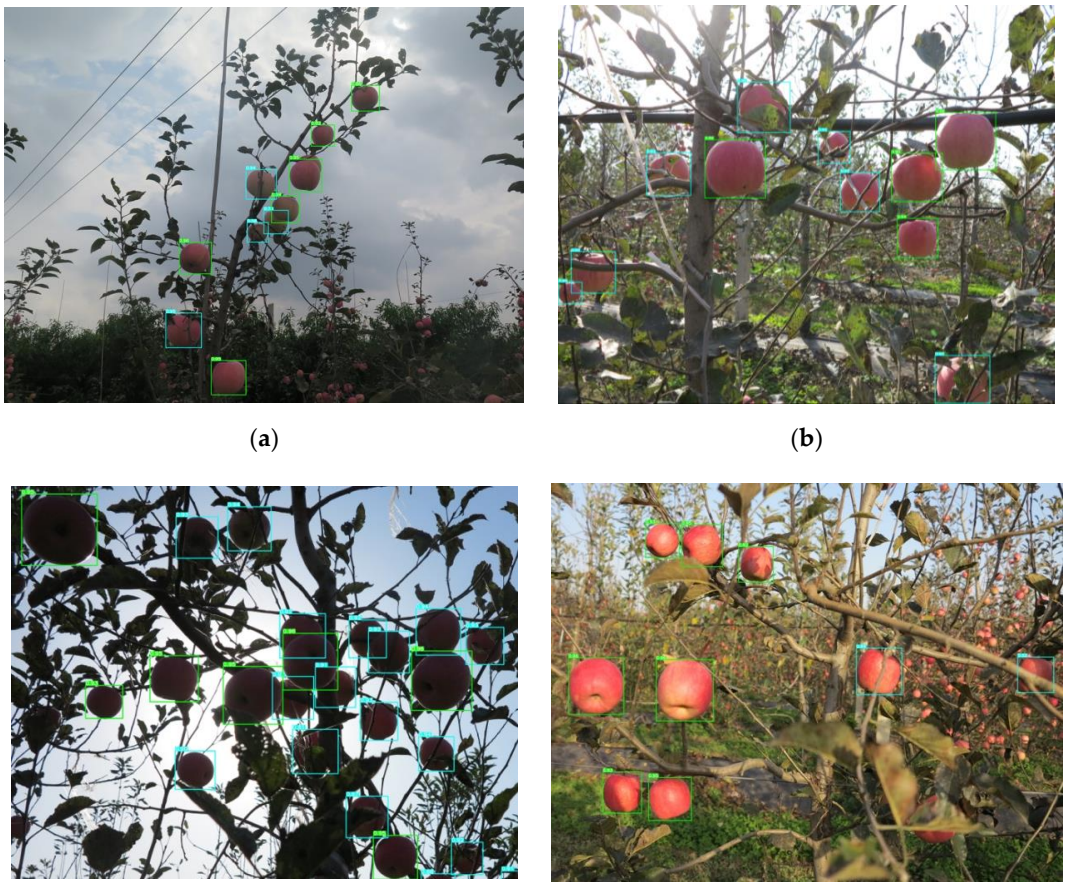
In order to verify the performance of the fruit, a real-time recognition model for the apple picking robot based on the improved design of YOLOv5s in the study, the recognition results of the model on 200 images of the test set were further analyzed. There are a total of 2336 apple targets in 200 test set images, among which the target number of graspable fruits is 1007, and the target number of ungraspable fruits is 1329. The specific recognition results of the method proposed in the study are shown in Table 3, which indicates that, for the graspable fruits, the precision, recall, mAP value and F1 score of the proposed model were 85.51%, 94.33%, 89.23% and 89.70%, respectively; for ungraspable fruits, the identification results were 82.56%, 89.32%, 84.87% and 85.81%, respectively. The overall identification precision, recall, mAP and F1 were 83.83%, 91.48%, 86.75% and 87.49%, respectively.

Table 3. Recognition results of apple targets using improved YOLOv5s network.

Test Set	Number	Precision (%)	Recall (%)	mAP (%)	F1 (%)
Graspable apple	1007	85.51	94.33	89.23	89.70
Ungraspable apple	1329	82.56	89.32	84.87	85.81
Total	2336	83.83	91.48	86.75	87.49

Examples of the recognition results of the proposed model for graspable and ungraspable fruits in different weather and lighting conditions are shown in Figure 14. Green

boxes were used for the label graspable fruits, while blue boxes were used for the label ungraspable fruits. As can be seen in Figure 14, the proposed recognition model is not only suitable to detect the images captured under uniform illumination on cloudy days, but also applicable to detect the images captured under sunny conditions. Moreover, the two varieties of fruit targets could also be well recognized under frontlight, backlight and sidelight conditions on a sunny day, utilizing the proposed model.



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

(**c**) (**d**)

Figure 14. Recognition results of apples using improved YOLOv5s network. (**a**) Cloudy (**b**) Sidelight of sunny (**c**) Backlight of sunny (**d**) Frontlight of sunny.

3.2. Comparison with the Recognition Results Using Different Object Detection Algorithms

In order to further analyze the recognition performance of the proposed algorithm for apple targets, the improved YOLOv5s network was compared with the original YOLOv5s, YOLOv3, YOLOv4 and EfficientDet-D0 network on 200 images of test set in the study. The mAP value and average recognition speed of the model were taken as evaluation indicators. The recognition results, size and the number of parameters of each network model are shown in Table 4.

According to Table 4, the mAP value of the improved YOLOv5s recognition model proposed in the study was the highest, which was 5.05% higher than that of the original YOLOv5 network, 14.95%, 4.74% and 6.75% higher than that of the YOLOv3, YOLOv4 and EfficientDet-D0 network respectively, indicating that the algorithm proposed was the best for apple targets recognition among five methods. For the recognition speed of the model, the average detection speed of the improved YOLOv5s model was 0.015 s per image (66.7 fps) (fps, frames per second), which was 2.53, 1.13 and 3.53 times of EfficientDet-D0, YOLOv4 and YOLOv3 network respectively, indicating that the proposed model can satisfy the requirements of the picking robot for real-time apple recognition. On the other hand,

it can be seen from Table 4 that the size of the improved YOLOv5s recognition model proposed in the study was only 12.7 MB, accounting for 90.71%, 5.4%, 5.2% and 84.67% of the original YOLOv5s, YOLOv3, YOLOv4 and EfficientDet-D0 networks, respectively. It showed that the proposed network cannot only ensure the recognition accuracy, but also realize the lightweight properties of the network effectively.

Table 4. Performance comparison of five object detection networks.

Object Detection Networks	mAP (%)	Average Detection Speed (s/pic)	Number of Parameters	Size of Model (MB)
YOLOv5s	81.7	0.013	7.25×10^6	14.0
YOLOv3	71.8	0.053	6.15×10^7	235.0
YOLOv4	82.01	0.017	6.39×10^7	244.0
EfficientDet-D0	80.0	0.038	3.83×10^6	15.0
Our network	86.75	0.015	6.52×10^6	12.7

Overall, the model proposed in the study was the most lightweight among the five network models, with the highest mAP value. The recognition speed of the proposed model is faster than that of the EfficientDet-D0, YOLOv3 and YOLOv4 networks. Although the recognition speed was slightly lower than that of the original YOLOv5s network, the average recognition speed could reach 66.7 fps, which can satisfy the requirements of real-time apple recognition.

Examples of recognition results of the five network models are shown in Figure 15, which are the identification results of apple targets in cloudy and sunny conditions. As shown in Figure 15a, the identification results of the improved YOLOv5s network proposed in the study were accurate, without misrecognition or missed recognition. It can be seen that, in cloudy conditions, the misrecognition that the ungraspable apple was identified as a graspable one occurred in YOLOv5s, EfficientDet-D0, and YOLOv3 networks (labeled by yellow ellipse in Figure 15b(1),c(1),d(1)). Moreover, missed recognition that the ungraspable apple was not identified occurred in EfficientDet-D0, YOLOv3 and YOLOv4 networks (labeled by white ellipse in Figure 15c(1),d(1),e(1,2)). For sunny condition, the misrecognition that the ungraspable apple was identified as a graspable one occurred in EfficientDet-D0, YOLOv3 and YOLOv4 network (labeled by yellow ellipse in Figure 15c(2),d(2),e(2)), and the misrecognition that the graspable apple was identified as an ungraspable one occurred in EfficientDet-D0 (labeled by pink ellipse in Figure 15c(2)).

(**1**) Cloudy (**2**) Sunny

(**a**)

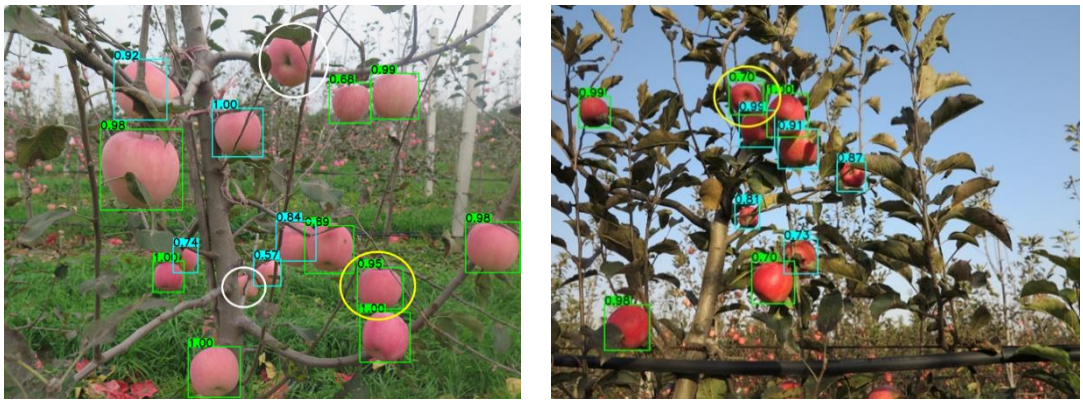
Figure 15. *Cont.*

(**1**) Cloudy (**2**) Sunny

(**b**)

(**1**) Cloudy (**2**) Sunny

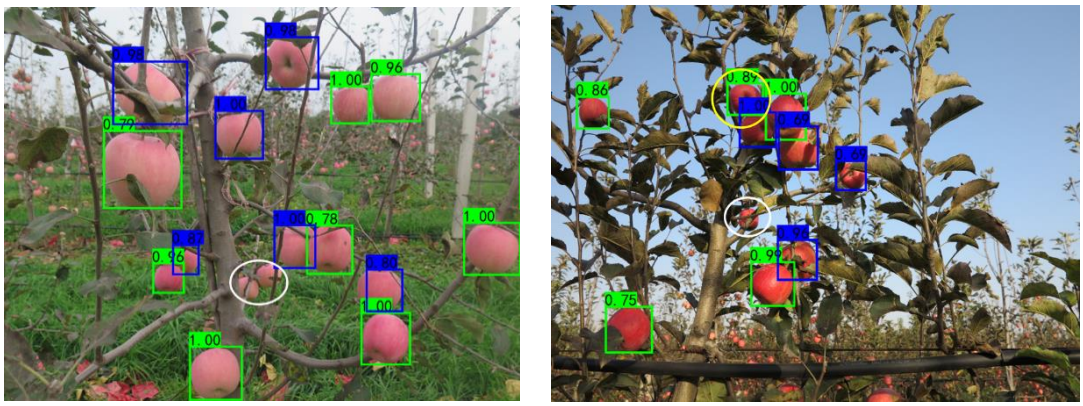
(**c**)



(**1**) Cloudy (**2**) Sunny

(**d**)

Figure 15. *Cont.*



(**1**) Cloudy (**2**) Sunny

(**e**)

Figure 15. Examples of the recognition results of five network models. (**a**) Improved YOLOv5s (**b**) YOLOv5s (**c**) EfficientDet-D0 (**d**) YOLOv3 (**e**) YOLOv4.

4. Discussion

Most of the existing recognition algorithms for fruits on apple trees regarded apple targets in the complex orchard environment (leaf occlusion, branch occlusion, mixed occlusion and fruit occlusion etc.) as one class. However, there are few studies on the multi-classification recognition of apple targets. Faster R-CNN network was used by Gao et al. [24] to identify four types of apple targets under different conditions on apple trees, including non-occluded fruit, leaf-occluded fruit, branch/wire-occluded fruit, and fruit-occluded fruit. In order to verify the recognition performance of the algorithm proposed in the study, the recognition results of the proposed algorithm were compared with those of the algorithm proposed by Gao et al. [24], shown in Table 5.

Table 5. Performance comparison with multi-class detection method for apple.

Detection Method	mAP (%)	mAP of Different Classes (%)				Detection Speed (s/pic)	Size of Model (MB)
		Graspable		Ungraspable			
		Non-Occluded	Leaf-Occluded	Branch/Wire-Occluded	Fruit-Occluded		
Faster R-CNN(VGG16)	87.9	90.9	89.9	85.8	84.8	0.241	512
Our model	86.75	89.23		84.87		0.015	12.7

Considering more categories division for apple targets will lead to the increasement of recognition model size, increasing the difficulty of network identification and detection time, since the algorithm proposed in the study was a lightweight fruit targets real-time detection method for apple picking robot, therefore, the classification of the apples into two categories including graspable apple (not occluded or only occluded by leaves) and ungraspable one (other conditions) was already satisfied with actual requirements.

As can be seen from Table 5, the mAP of the compared algorithm for the four types of fruits was 90.9%, 89.9%, 85.8% and 84.8% respectively, and the overall mAP was 87.9%, which was 1.15% higher than that of the recognition algorithm proposed in the paper (86.75%). However, the size of the recognition model in Gao's paper was relatively large, with a weight file of 512 MB; the size of the light-weight recognition model proposed in this paper is 12.7 MB, accounting for only 2.48% of the size of model in Gao's paper. Significantly, the lightweight of the model size is conducive to the deployment of model in hardware devices later. On the other hand, the apple recognition speed is significant for the picking robot, which influences the working efficiency of the robot. The recognition speed of the model in Gao's paper was 0.241 s per image (4.15 fps), which was not satisfied with

the requirements of real-time apple targets recognition. However, the average detection speed of the model proposed in the study was 0.015 s per image (66.7 fps), which was 16.07 times the detection speed in Gao's paper, satisfying the requirements of picking robot for real-time apple recognition.

Furthermore, it is notable that the recognition results of YOLOv4 algorithm in Section 3.2 shows that the detection performance of the model is good, since its mAP was the highest among the four contrasted algorithms, even higher than the mAP of original YOLOv5s, which reflects the excellent target detection ability of the model. However, the size of the model is relatively large, reaching 244 MB, which may increase the deployment cost of the recognition algorithm in the embedded devices of picking the robot vision system.

The YOLOv5 network contains four varieties of architectures (YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x) with different sizes, indicating the strong flexibility of this model.

Therefore, users can choose the specific model with appropriate size for development and application based on the actual task requirements. The main consideration of the selection and design of recognition algorithm in this study was its future application environment which was the application deployment of the detection algorithm on the apple picking robot to realize real time fruit targets recognition. The advantages of lightweight (very small model size) and high detection speed of YOLOv5s network will reduce the deployment cost of the recognition model, indicating the great potential of the detection model based on improved YOLOv5s for deployment in the embedded devices of picking robot vision system.

Overall, the strengths of the proposed apple detection algorithm are reflected in the following points: firstly, it is able to automatically recognize the graspable and ungraspable apple targets in an image which has not been studied before; secondly, the detection performance, especially the detection speed of the improved designed YOLOv5s model is excellent, which is suitable for the real-time apple recognition of the picking robot; finally, the size of the proposed detection model is very lightweight, indicating that it has great potential to be deployed in hardware devices, which is essential for the wide application of the detection algorithm, it is related to the cost of picking robot's visual system, since the larger the model, the higher the requirement of configuration and computing ability of the hardware equipments.

On the other hand, since the apple picking robot has the ability of working at night, however, the algorithm proposed in the study is designed for fruit recognition in the daytime. Therefore, it is not necessarily suitable for apple target recognition at night, which is a limitation of our detection algorithm. In addition, the recognition object of the paper is a red apple which is widely planted. However, a large number of green apple trees are planted in the same apple orchard at the same time in general, but the algorithm proposed in the paper cannot realize the recognition of green apples, which limits the application scope of the apple picking robot, indicating another limitation of our detection algorithm.

5. Conclusions and Future Work

In order to realize the automatic recognition of graspable and ungraspable fruits for the picking robot in an apple tree image, a light-weight fruit target real-time detection method for the apple picking robot based on improved YOLOv5 was proposed in the study.

In the improved designed network architecture, to realize the light-weight improvement of the network, BottleneckCSP module was improved designed to BottleneckCSP-2 module, which was used to replace the BottleneckCSP module in the backbone architecture of original YOLOv5s network. To acquire the feature of apple targets under different conditions better, SE module was inserted to the improved designed backbone network. To improve the recognition accuracy of apple targets, the bonding fusion mode of feature maps which were input into the target detection layer of medium size in original YOLOv5s network was improved. To avoid the misrecognition of small apples in the background of image, the initial anchor box size of original network was improved. The detection

results of the test set showed that the proposed improved network model can effectively realize the recognition of the fruits that can be grasped by picking robot and ungraspable in the current apple tree image. The total recall, precision, mAP, and F1 of recognition were 91.48%, 83.83%, 86.75%, and 87.49%, respectively. The average detection speed was 0.015 s per image.

The detection results for two varieties of targets of the improved YOLOv5s algorithm proposed in the study were compared with the detection results of another four algorithms on the 200 test set images. The results indicated that, contrasting with original YOLOv5s, YOLOv3, YOLOv4 and EfficientDet-D0 model, the mAP of the proposed improved YOLOv5s model increased by 5.05%, 14.95%, 4.74% and 6.75%, respectively. The size of the model was compressed by 9.29%, 94.6%, 94.8% and 15.3%, respectively. The average recognition speed of the proposed improved model was 0.015 s per image (66.7 fps), which was 2.53, 1.13 and 3.53 times of EfficientDet-D0, YOLOv4 and YOLOv3 network respectively, satisfying the requirements of real-time apple detection.

Future Work

Consider the limitations of the proposed detection algorithm illustrated in 4. In discussion, in terms of improving the application scope of our algorithm and apple picking robot, a large number of green apple image data will be captured. Furthermore, the images of red apples and green apples will also be captured at night by utilizing artificial lighting. All the image samples above will be added to the training sets which are used for training of the detection model, in order to realize automatic recognition of the graspable and ungraspable red or green apple targets in an image in day and night. On the other hand, since the proposed algorithm can realize real time recognition of the graspable and ungraspable apple targets for the picking robot, it can therefore be combined with the movement control strategy of the grasping end-effector in the next step, in order to realize the picking of apples occluded by branches or fruits by adjusting the picking angle and the position of the end-effector. Future work also includes the recognition of fruits on other apple tree varieties or fruits of protected horticulture based on the improved designed detection network architecture. Furthermore, the recognition of target objects in unmanned aerial vehicle (UAV-based) remote sensing images utilizing the detection network architecture proposed in this study is also worth studying in the future.

Author Contributions: All authors contributed extensively to this manuscript. B.Y. contributed to the development of the algorithm, obtaining apples images, programming, and writing. B.Y. also performed the experiments and analyzed the results. X.L. helped in obtaining the images of apples. P.F. contributed to the original draft preparation. F.Y. and Z.L. reviewed and edited the draft. F.Y. provided significant contributions to this development as the lead. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the science and technology projects in Shaanxi Province Development and Application of key equipment for Orchard Mechanization and Intelligence (Grant No. 2020zdzx03-04-01).

Institutional Review Board Statement: The study in the paper did not involve humans or animals.

Informed Consent Statement: The study in the paper did not involve humans or animals.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We sincerely thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

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

References

1. Fu, L.; Gao, F.; Wu, J.; Li, R.; Karkee, M.; Zhang, Q. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Comput. Electron. Agric.* **2020**, *177*. [CrossRef]

2. Zhang, Z.; Igathinathane, C.; Li, J.; Cen, H.; Lu, Y.; Flores, P. Technology progress in mechanical harvest of fresh market apples. *Comput. Electron. Agric.* **2020**, *175*. [CrossRef]
3. Ghosh, I.; Datta Chaudhuri, T. FEB-Stacking and FEB-DNN models for stock trend prediction: A performance analysis for pre and post Covid-19 periods. *Decis. Mak. Appl. Manag. Eng.* **2020**, *4*. [CrossRef]
4. Malinda, M.; Chen, J. The forecasting of consumer exchange-traded funds (ETFs) via grey relational analysis (GRA) and artificial neural network (ANN). *Empir. Econ.* **2020**, *3*. [CrossRef]
5. Precup, R.; Preitl, S.; Petriu, E.; Bojan-Dragos, C.; Szedlak-Stinean, A.; Roman, R.; Hedrea, E. Model-Based fuzzy control results for networked control systems. *Rep. Mech. Eng.* **2020**, *1*. [CrossRef]
6. Mirko, S.; Aleksandar, S.; Đorđe, S. ANFIS model for the prediction of generated electricity of photovoltaic modules. *Decis. Mak. Appl. Manag. Eng.* **2019**, *2*. [CrossRef]
7. Messinis, S.; Vosniakos, G. An agent-based flexible manufacturing system controller with Petri-net enabled algebraic deadlock avoidance. *Rep. Mech. Eng.* **2020**, *1*. [CrossRef]
8. Hu, L.; Liu, Z.; Hu, W.; Wang, Y.; Tan, J.; Wu, F. Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *J. Manuf. Syst.* **2020**, *55*. [CrossRef]
9. Kerkech, M.; Hafiane, A.; Canals, R. VddNet: Vine disease detection network based on multispectral images and depth map. *Remote Sens.* **2020**, *12*, 3305. [CrossRef]
10. Fromm, M.; Schubert, M.; Castilla, G.; Linke, J.; McDermid, G. Automated detection of conifer seedlings in drone imagery using convolutional neural networks. *Remote Sens.* **2019**, *11*, 2585. [CrossRef]
11. Afzaal, H.; Farooque, A.A.; Schumann, A.W.; Hussain, N.; McKenzie-Gopsill, A.; Esau, T.; Abbas, F.; Acharya, B. Detection of a potato disease (early blight) using artificial intelligence. *Remote Sens.* **2021**, *13*, 411. [CrossRef]
12. Abdulridha, J.; Ampatzidis, Y.; Qureshi, J.; Roberts, P. Laboratory and UAV-Based identification and classification of tomato yellow leaf curl, bacterial spot, and target spot diseases in tomato utilizing hyperspectral imaging and machine learning. *Remote Sens.* **2020**, *12*, 2732. [CrossRef]
13. Biffi, L.J.; Mitishita, E.; Liesenberg, V.; dos Santos, A.A.; Goncalves, D.N.; Estrabis, N.V.; Silva, J.d.A.; Osco, L.P.; Ramos, A.P.M.; Centeno, J.A.S.; et al. ATSS deep learning-based approach to detect apple fruits. *Remote Sens.* **2021**, *13*, 54. [CrossRef]
14. Fuentes-Pacheco, J.; Torres-Olivares, J.; Roman-Rangel, E.; Cervantes, S.; Juarez-Lopez, P.; Hermosillo-Valadez, J.; Rendón-Mancha, J.M. Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sens.* **2019**, *11*, 1157. [CrossRef]
15. Hani, N.; Roy, P.; Isler, V. A comparative study of fruit detection and counting methods for yield mapping in apple orchards. *J. Field Robot.* **2020**, *37*, 263–282. [CrossRef]
16. Zhang, T.; Zhang, X. High-Speed ship detection in SAR images based on a grid convolutional neural network. *Remote Sens.* **2019**, *11*, 1206. [CrossRef]
17. Peteinatos, G.; Reichel, P.; Karouta, J.; Andujar, D.; Gerhards, R. Weed Identification in Maize, Sunflower, and Potatoes with the Aid of Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 4185. [CrossRef]
18. Hoeser, T.; Kuenzer, C. Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sens.* **2020**, *12*, 1667. [CrossRef]
19. Bresilla, K.; Perulli, G.D.; Boini, A.; Morandi, B.; Grappadelli, L.C.; Manfrini, L. Single-Shot convolution neural networks for real-time fruit detection within the tree. *Front. Plant Sci.* **2019**, *10*. [CrossRef]
20. Zhao, D.; Wu, R.; Liu, X.; Zhao, Y. Apple positioning based on YOLO deep convolutional neural network for picking robot in complex background. *Trans. Chin. Soc. Agric. Eng.* **2019**, *35*, 164–173. [CrossRef]
21. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*. [CrossRef]
22. Wang, D.; He, D. Recognition of apple targets before fruits thinning by robot based on R-FCN deep convolution neural network. *Trans. Chin. Soc. Agric. Eng.* **2019**, *35*, 156–163. [CrossRef]
23. Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* **2020**, *197*, 245–256. [CrossRef]
24. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Comput. Electron. Agric.* **2020**, 176. [CrossRef]
25. Gene-Mola, J.; Vilaplana, V.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Gregorio, E. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. *Comput. Electron. Agric.* **2019**, *162*, 689–698. [CrossRef]
26. Zhang, J.; Karkee, M.; Zhang, Q.; Zhang, X.; Yaqoob, M.; Fu, L.; Wang, S. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Comput. Electron. Agric.* **2020**, 173. [CrossRef]
27. Gene-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V.; Gregorio, E. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. *Comput. Electron. Agric.* **2020**, 169. [CrossRef]
28. Kang, H.; Chen, C. Fruit detection and segmentation for apple harvesting using visual sensor in orchards. *Sensors* **2019**, *19*, 4599. [CrossRef] [PubMed]
29. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* **2020**, *171*. [CrossRef]

30. Liu, Y.; Lu, B.; Peng, J.; Zhang, Z. Research on the use of YOLOv5 object detection algorithm in mask wearing recognition. *World Sci. Res. J.* **2020**, *6*, 276–284.
31. ultralytics. yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 18 May 2020).
32. Sun, S.; Jiang, M.; Liang, N.; He, D.; Long, Y.; Song, H.; Zhou, Z. Combining an information-maximization-based attention mechanism and illumination invariance theory for the recognition of green apples in natural scenes. *Multimed. Tools Appl.* **2020**, *79*, 28301–28327. [CrossRef]
33. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]