agriculture

MDPI

Article

Improved Apple Fruit Target Recognition Method Based on YOLOv7 Model

Huawei Yang [1,2,3], Yinzeng Liu [3], Shaowei Wang [3], Huixing Qu [1], Ning Li [3], Jie Wu [1], Yinfa Yan [1], Hongjian Zhang [1], Jinxing Wang [1,*] and Jianfeng Qiu [2,*]

[1] College of Mechanical and Electrical Engineering, Shandong Agricultural University, Tai'an 271002, China; yhw105@163.com (H.Y.); huixing0219@163.com (H.Q.); cangzhibiwujie@163.com (J.W.); sd28@163.com (Y.Y.); zhanghongji_an@163.com (H.Z.)
[2] College of Radiology, Shandong First Medical University, Tai'an 271000, China
[3] Shandong Academy of Agricultural Machinery Sciences, Jinan 250010, China; lyz19971024@163.com (Y.L.); itismyway163@163.com (S.W.); palm06@163.com (N.L.)
* Correspondence: jinxingw@163.com (J.W.); jfqiu100@163.com (J.Q.)

Abstract: This study proposes an improved algorithm based on the You Only Look Once v7 (YOLOv7) to address the low accuracy of apple fruit target recognition caused by high fruit density, occlusion, and overlapping issues. Firstly, we proposed a preprocessing algorithm for the split image with overlapping to improve the robotic intelligent picking recognition accuracy. Then, we divided the training, validation, and test sets. Secondly, the MobileOne module was introduced into the backbone network of YOLOv7 to achieve parametric fusion and reduce network computation. Afterward, we improved the SPPCSPS module and changed the serial channel to the parallel channel to enhance the speed of image feature fusion. We added an auxiliary detection head to the head structure. Finally, we conducted fruit target recognition based on model validation and tests. The results showed that the accuracy of the improved YOLOv7 algorithm increased by 6.9%. The recall rate increased by 10%, the mAP1 algorithm increased by 5%, and the mAP2 algorithm increased by 3.8%. The accuracy of the improved YOLOv7 algorithm was 3.5%, 14%, 9.1%, and 6.5% higher than that of other control YOLO algorithms, verifying that the improved YOLOv7 algorithm could significantly improve the fruit target recognition in high-density fruits.

Keywords: deep learning; apple; object detection; data augmentation

Citation: Yang, H.; Liu, Y.; Wang, S.; Qu, H.; Li, N.; Wu, J.; Yan, Y.; Zhang, H.; Wang, J.; Qiu, J. Improved Apple Fruit Target Recognition Method Based on YOLOv7 Model. *Agriculture* **2023**, *13*, 1278. https://doi.org/10.3390/agriculture13071278

Academic Editor: Maciej Zaborowicz

Received: 16 May 2023
Revised: 14 June 2023
Accepted: 18 June 2023
Published: 21 June 2023

1. Introduction

China is the world's largest apple producer. The country's planting area reached 2,088,080 ha in 2021, and the output reached 45,973,400 t, accounting for more than 50% of the world's apple output. However, the lack of labor force in Chinese orchards, high labor intensity, and low efficiency in apple picking are increasingly prominent. The research momentum of intelligent fruit-picking technology has significantly increased with the recent development of emerging technologies such as machine vision, robotics, and artificial intelligence [1–7]. The fruit growth density of apple orchards is high because low anvil-dense planting is an unstructured scene [8]. Many overlapping occlusions, leaf occlusions, branch occlusions, and other problems are remarked, resulting in the difficulty of detection, recognition, and low precision of fruit target identification problems.

Given the above problems, scholars specializing in fruit recognition and detection have conducted significant research and proposed many new algorithms. In terms of traditional digital image processing, Bulanon et al. [9] performed threshold segmentation processing to enhance the color difference of the apple image's red channel and extract the apple fruit target. The processing recognition rate reached 88.0%, but its recognition rate in the backlight environment was only 18.0%. Gongal et al. [10] converted the captured red, green, and blue (RGB) images into HIS images and conducted histogram equalization processing. Then, the

processed images were converted into RGB images and achieved threshold segmentation (OTSU). Finally, Hough transforms, and Blob analysis were used for the recognition of apple targets. Based on the R-G color features of apple fruit, Lv et al. [11] used OTSU for image segmentation and determined the center of the apple target through the center principle. The experimental results showed that the OTSU recognition time was reduced by 36%. Mai et al. [12] extracted apple fruit shapes based on Log-Hough transform for apple recognition. The experimental results show that the average recognition rate of the method can reach 91.6% in the case of fruit occlusion, overlap and color variation. Si et al. [13] proposed a normalized red-green difference $(r - g)/(R + G)$ apple segmentation method, using the random ring method to conduct template matching and complete the recognition and positioning of the apple fruit. The experiment showed that the recognition rate of this method reached 92%, and the measurement error was less than 2 cm. However, traditional digital image processing methods have low accuracy and are susceptible to environmental noise, resulting in poor recognition performance and inaccurate object edges.

Recently, deep learning techniques have been widely used for fruit recognition. The most prominent of these is the You Only Look Once (YOLO) algorithm. Many scholars have used the YOLO algorithm for fruit object detection and have achieved notable results. In addition, many scholars have applied the YOLO algorithm to evaluate fruit yields and study plant traits [14–17]. Praveen et al. [18] integrated the adaptive pooling scheme and the attribute augmentation model into the yolov5 architecture and introduced a loss function to obtain an accurate bounding box, thereby maximizing detection accuracy. This model detects smaller objects and improves the feature quality to detect apples in complex backgrounds. The overall accuracy was 0.97, 0.99, and 0.98 in terms of precision, recall, and *F*1-score, respectively. Altaheri et al. [19] used deep learning image processing to improve AlexNet and VGG16 networks and used the convolutional layer to extract image features. The recognition accuracy of this method for unshielded jujube fruit was more than 90%. Ji et al. [20] proposed an improved YOLOX algorithm for apple fruit target detection. Yolox-Tiny network introduced the attention mechanism of the lightweight model Shufflenetv2 and a convolutional block attention module (CBAM) and added the adaptive spatial feature fusion module into the Path Aggregation Network (PANet). The experimental results showed that this network model's average accuracy, precision, recall rate, and *F*1 are 96.76%, 95.62%, 93.75%, and 0.95, respectively. Zhao et al. [21] realized target recognition of apple fruit in a complex environment based on the improved YOLOv3 by combining the residual module in the original trunk network with CSPNet and adding the SPP module to the neck structure to achieve integration of global and local features. Finally, Focal Loss and CIoU Loss were used to optimizing the model, and the experiment showed that the MAP value of the algorithm reached 96.3%. Yang et al. [22] used the improved CenterNet network to identify multi-apple targets in dense scenes quickly. The experimental results show that the average accuracy of this method is 98.9%, and the *F*1 value is 96.39%. However, the training time of the AlexNet and VGG16 network models is long, and they are not easy to deploy. The CenterNet network structure is prone to misjudging the center point for two objects in the same category that is close to each other. The YOLO model has the advantage of high accuracy, a short recognition time and easy deployment. However, it performs poorly on tasks involving object occlusion, object overlap, and small objects in unstructured apple orchard environments. It is thus necessary to modify the original model framework to achieve the accurate recognition of fruit targets.

In summary, the digital image processing method is simple but less robust in unstructured apple orchard environments. Its recognition accuracy is easily perturbed by factors such as illumination, background color, overlapping fruit, branches, and leaves, and its field test results are poor. Deep learning methods can extract the high-dimensional features of fruits to resist illumination, overlap, and occlusion effects [23,24], are robust in detecting apple targets and have high recognition accuracy. The YOLO algorithm has the advantages of easy deployment, easy training, and high recognition efficiency to meet the requirements of the real-time detection of apple targets in an unstructured orchard envi-

ronment [25–31]. To address the problems of low accuracy in apple fruit target recognition caused by high fruit density, occlusion, and overlapping, this study proposes an improved detection method for apple fruit targets based on the YOLOv7 model. MobileOne module is introduced to enhance the hyper-parametric fusion of the recognition network model and reduce the computation of the network model.

Moreover, the SPPFCSPC module is introduced to improve the fusion speed of image features. The module also improves the robustness of the recognition model in the natural environment, growth conditions, and other settings. Finally, the SPPFCSPC module improves the recognition accuracy of the entire model for complex apple targets.

2. Image Data Collection and Preprocessing

2.1. Apple Image Data Collection

In this study, the Honor 30S mobile phone was used for image acquisition. The focal length is 17–80 mm, the aperture value is f/1.8, and the maximum image resolution is 3456 × 4608 pixels. This paper collected the image data of 2-year-old red Fuji apples from the Fruit Science Demonstration Base of Shandong Academy of Agricultural Sciences (117°13′6.24972″ E, 36°28′36.05484″ N). Moreover, RGB images were collected at the apple orchard of Lanting New Village, Langao Town, Longkou City, Yantai City (120°35′56.48″ E, 37°37′31.30″ N). The mobile phone adopted multi-angle and multi-distance shooting, and its distance from the apple fruit is 0.5 to 1.5 m when capturing the image. Various images with uneven illumination, backlight, mutual occlusion of fruits, and occlusion of leaves and branches were collected and saved in JPG format with an image resolution of 3456 × 4608 pixels, as shown in Figure 1. From these images, 474 image samples were finally obtained.

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

Figure 1. Orchard image: (**a**) Uneven illumination, (**b**) Backlight, and (**c**) Smooth light.

2.2. Dataset Production

The YOLO algorithm adjusted the input image to a 1:1 image and compressed it. In the compression process, detailed information that consisted of a small proportion of pixels in the original image would be lost [32,33]. Therefore, we used the image segmentation method to adjust the image size and divide the original image into several sub-images with the same resolution to reduce the loss of semantic information caused by image compression. The size of the input image was set to 640 × 640 pixels. After testing, if the image was divided into 640 × 640 pixels, some images would not be able to fully display the detection target, which would affect the training effect of the recognition model. Therefore, this study divided the collected image into several sub-images with a resolution of 1280 × 1280 pixels. To avoid incomplete targets caused by image splitting, this study proposes an overlapping image segmentation method. Firstly, the threshold of 280 × 280 pixels was determined by measuring the pixel size of a single apple in the collected image. Secondly, the image was segmented into sub-images of 1280 × 1280 pixels

by forward or upward filling. Finally, the original image was divided into 20 sub-images so that each target in the original image could be marked. The results are shown in Figure 2. After manually screening the split images, 1245 images were finally selected as the original images for training.

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

Figure 2. Apple image data enhancement: (**a**) Original images and (**b**) Diagram showing the effect of the overlapping image segmentation method.

LabelImg software, the open-source data labeling tool created by Tzutalin with the help of dozens of contributors, was used to annotate the original image, and the change save format was set to PASCAL VOC mode to generate an XML annotation file. This annotation information included the image file name, image size, pixel coordinates of the upper left corner, and pixel coordinates of the lower right corner. The annotated images must be randomly divided into three categories: training, validation, and test set to verify the performance of the training model in a ratio of 8:1:1.

2.3. Apple Image Data Amplification

PyTorch and OpenCV were used to conduct image enhancement processing on training, validation, and test sets to realize the expansion of the dataset, reduce network overfitting, and improve the generalization ability of the recognition model. Image enhancement methods included multi-angle rotation (45°, 90°, 180°), mirror image (horizontal, vertical, and diagonal), gray equalization, brightness (brightness value increased by 50%; brightness value decreased by 50%), and reverse color [34,35]. The image data enhancement effect is shown in Figure 3. Finally, the number of training, validation, and test set samples was 9950, 1250, and 1250, respectively. The platform for image preprocessing included a notebook computer equipped with a CPU of Intel Core i7-7500U (2.70 GHz), 8 GB of RAM, and an NVIDIA GTX 940MX 4 GB GPU, running on a Windows 10 64-bit system. Software tools included CUDA 11.3, PyTorch 1.10.0, PyCharm Community Edition 2020.2.1, Python 3.8, and OpenCV 4.5.3.

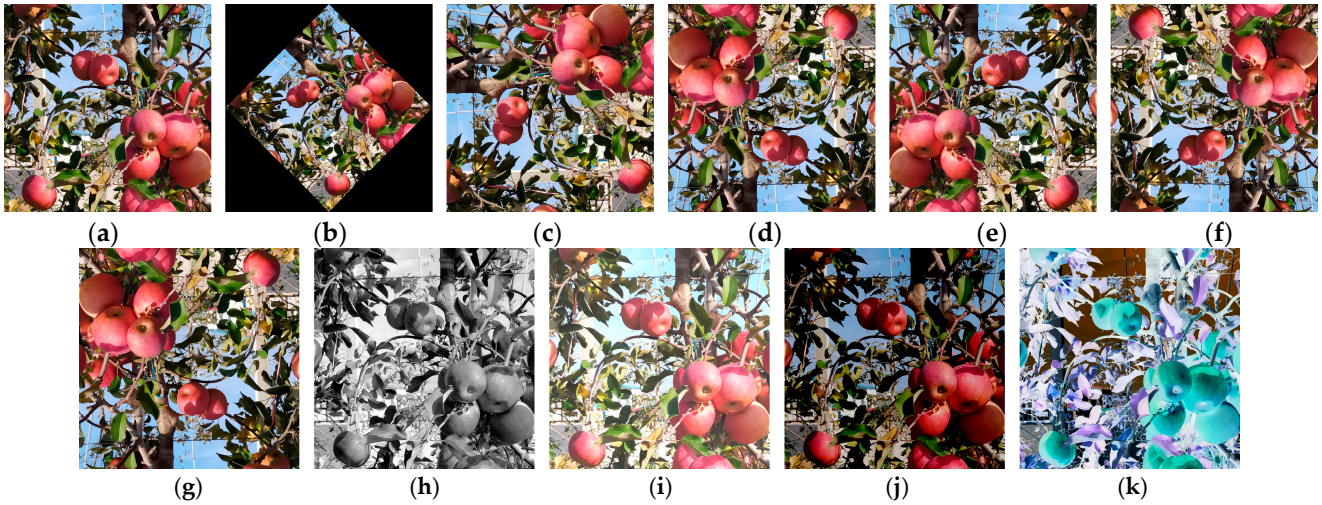

Figure 3. Apple image data enhancement: (**a**) Original images, (**b**) 45-degree rotation, (**c**) 90-degree rotation, (**d**) 180-degree rotation, (**e**) Horizontal mirror, (**f**) Vertical mirror, (**g**) Diagonal mirror, (**h**) Gray equalization, (**i**) Brightness value increased by 50%, (**j**) Brightness value decreased by 50%, and (**k**) Reverse color.

The XML annotation file was converted into the TXT file required for YOLO training. The annotation information included the category code and the relative horizontal center coordinate x_center. This relative vertical center coordinates y_center, the relative width w of the annotation box, and the relative height h of the annotation box. Each line of annotation information in this file type represents an annotation box.

3. Design and Training of the Identification Model

In this section, the design and training of the recognition model are described. The MobileOne-YOLOv7 network is presented in Section 3.1. The MobileOne network is described in Section 3.2. The SPPFCSPC module is presented in Section 3.3. Then, the addition of an auxiliary detection head is described in Section 3.4. In Section 3.5, the multi-scale training method is described, and the optimization of the loss function is presented in Section 3.6. Finally, the training platform and data analysis are described in Section 3.7.

3.1. MobileOne-YOLOv7 Network

YOLO is an object detection and classification algorithm that can quickly and accurately locate and identify multiple objects in a single image. The basic concept of the YOLO algorithm is to divide the entire image into grids, with each grid responsible for detecting one object in the image. An object may fall within multiple grids, and each grid predicts the object's category and position through a convolutional neural network. The YOLO algorithm integrates the classification and detection tasks and unifies them through the probabilities of classification and the confidence of the bounding boxes to calculate the final probability of the object. Compared to traditional algorithms, the YOLO algorithm has many advantages compared to other algorithms. Because it uses a fully convolutional neural network, it has fewer network parameters and a shorter training time. Furthermore, the network framework of the YOLO algorithm is simple and easy to understand and implement, and it can adapt to different types of image detection tasks. At the same time, the YOLO algorithm can independently detect different scales and types of objects, achieving good detection accuracy, and the accuracy of the object bounding box's position and size is no worse than that of traditional methods. The COCO128 dataset was used to test and compare the index values of different versions of the YOLO model. As can be seen from Table 1, compared with YOLOv3, YOLOv4, YOLOv5s, and YOLOv6, YOLOv7 had the highest values of AP@0.50 and AP@0.50–0.95. Moreover, its speed and accuracy in the 5 FPS to 160 FPS exceeded all known detectors [36,37]. Thus, we chose YOLOv7 to

improve the design of a new recognition network since the recognition model's detection accuracy and real-time performance were directly related to the accuracy and efficiency of apple target recognition of the picking robot.

Table 1. Test results of YOLOv3-YOLOv7 on the COCO128 dataset.

Model	Size (Pixel)	Date	Framework	Backbone	mAP@0.50	mAP@0.50–0.95
YOLOv3	640	2018	Darknet	Darknet53	83.4	51.9
YOLOv4	640	2020	Darknet	CSPD Darknet53	81.6	56.5
YOLOv5s	640	2020	PyTorch	Modified CSP v7	85.7	56.7
YOLOv6s	640	2022	PyTorch	EfficientRep	87.7	58.4
YOLOv7	640	2022	PyTorch	RepConvN	92.0	62.8

The YOLOv7 network model included six architectures: Yolov7-Tiny, YOLOv7, Yolov7-x, Yolov7-E6, Yolov7-D6, and Yolov7-E6e. The main differences were the number of convolutional layers and the number of auxiliary probes.

The YOLOv7 network comprised a Backbone network, a neck network, and a head network. The backbone network comprised the Conv + BN + Silu (CBS) module, efficient aggregation network (ELAN) module, MaxPool (MP) module, and SPPCSPC module. The neck network adopted the PANet network structure [38,39], which was a top-down and bottom-up bidirectional fusion backbone network. It realized the multi-scale fusion of the network by aggregating the characteristics between different backbone network layers and detection layers. The head network comprised three detection heads of different dimensions. The network structure of YOLOv7 is shown in Figure 4.

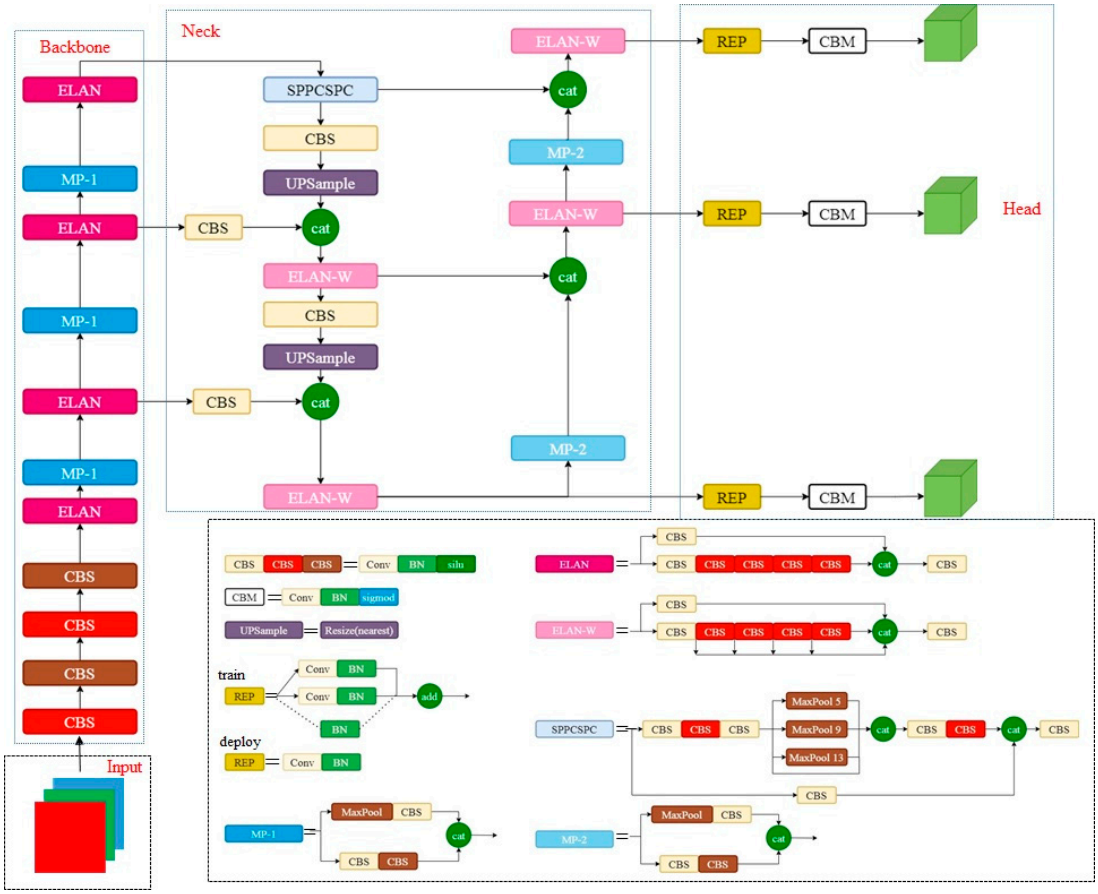

Figure 4. YOLOv7 structure diagram.

To address the problems of the network model being lightweight and the over-parameterization of the backbone, this study introduced the MobileOne module to replace the last ELAN module in the backbone, as shown in the red box in Figure 5. To improve the calculation speed of the network, the SPPCSPC module in the neck was improved and replaced with the SPPFCSPC module, as shown in the green box in Figure 5. At the same time, to enhance the recognition accuracy of the new network in the complex environment of the apple orchard, the first ELAN module in the backbone introduced a head for small target detection, as shown in the blue box in Figure 5.

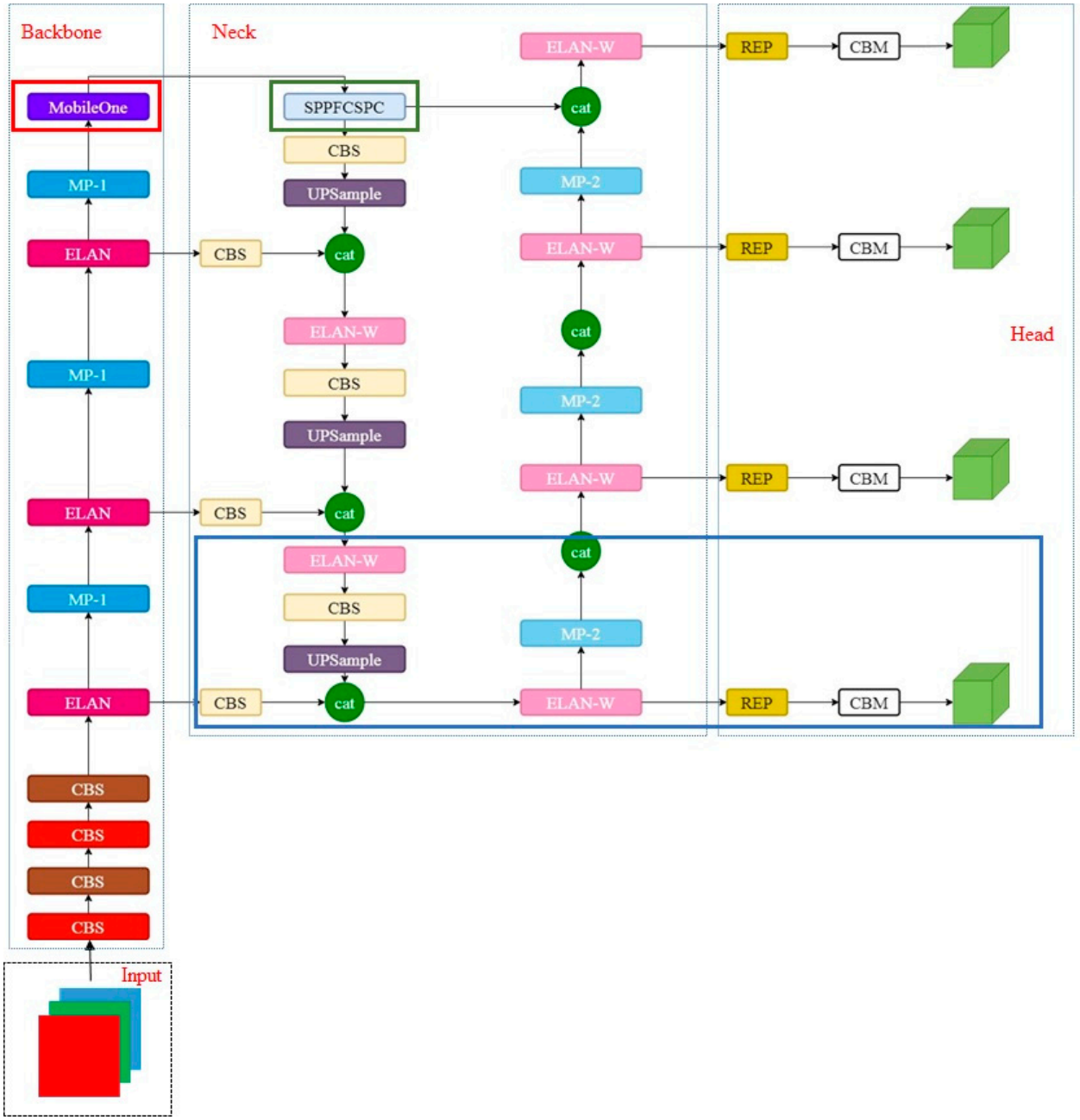

Figure 5. Improved YOLOv7 structure diagram.

3.2. MobileOne Network

MobileOne Network is an efficient Apple, Inc. neural network backbone developed for mobile devices [40–42]. The most significant differences between this network and ACNet, DBBNet, and RepVGG are as follows: the reasoning time is less than 1 ms, and the accuracy

rate can reach 75.9% when applied to ImageNet and when using the over-parametric and depth-separable convolution structure.

The left part of Figure 6 is a complete building block module structure composed of deep convolution and point convolution. The deep convolution is a grouping convolution, and the number of groups is the same as the input channel. Point convolution is a 1 × 1 convolution whose primary function is to freely change the number of output channels and channel fusion of the deep convolution output feature map.

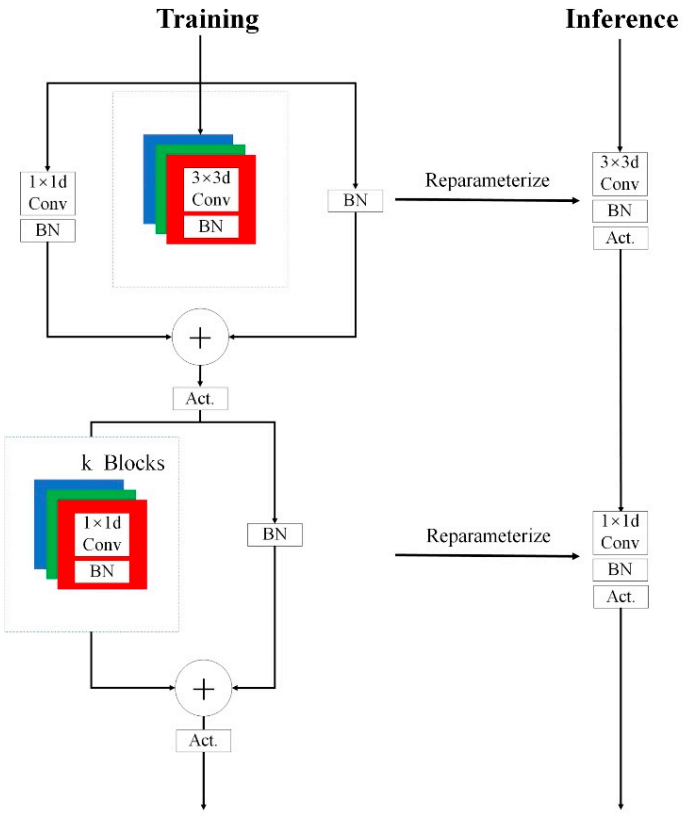

Figure 6. MobileOne module structure diagram.

The deep convolution module in Figure 6 comprises 1 × 1 convolution, over-parameterized 3 × 3 convolution, and batch normalization (BN) layers. However, both 1 × 1 and 3 × 3 convolutions are deep convolutions (grouping convolution). The point convolution module comprised over-parameterized 1 × 1 convolution and BN layers. During the network model training, the MobileOne network included stacked building blocks using the parameterized method at the end of the training. The MobileOne module was introduced for feature extraction because of the lightweight and over-parameterized features of the MobileOne network.

3.3. SPPFCSPC Module

SPPCSPS module was used in the original YOLOv7 network to obtain different receptive fields through maximum pooling and enlarge the receptive fields to improve the network model adaptability for the image of different resolutions. SPPCSPC module uses three convolution kernels of various sizes, such as 5 × 5, 9 × 9, and 13 × 13. The maximum pooling of four scales had four receptive fields to reveal the distinction between small and large targets. However, the Spatial Pyramid Pooling Connected Spatial Pyramid Convolution (SPPCSPC) structure increased the network computation using convolution kernels of different sizes for parallel pooling operation. Therefore, the Spatial Pyramid Pooling-Fast Connected Spatial Pyramid Convolution (SPPFCSPC) module was adopted for pooling operation. As shown in Figure 7, the module replaced three convolution kernels of parallel

maximum pooling operation in the original SPPCSPC structure with three serial maximum pooling operations of the same convolution kernels. This replacement improved the speed while keeping the receptive field unchanged.

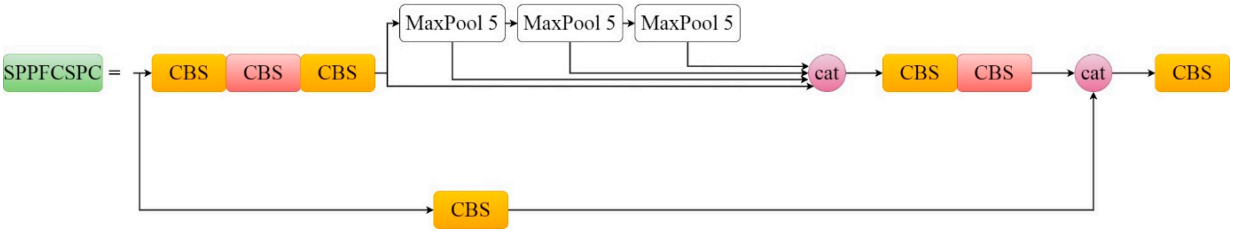

Figure 7. SPPFCSPC module structure diagram.

3.4. Add an Auxiliary Detection Head

This study added a prediction head to detect objects of different scales, improve the model's accuracy, and identify the location information of the small-scale apple target in a high-density complex environment. As shown in Figure 8, the neck network adopted the PANet structure, added an up-sampling, and added a forward splicing operation based on the original YOLOv7.

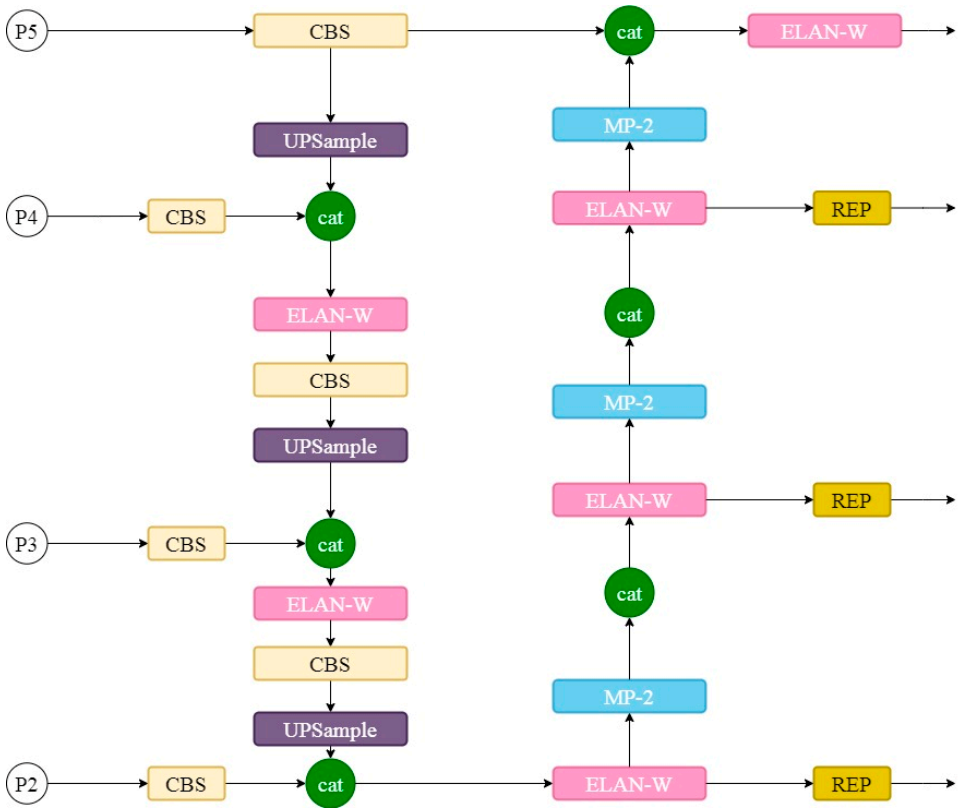

Figure 8. PANet structure.

3.5. Multi-Scale Training

The multi-scale feature extraction method [43–46] improved the identification model training accuracy. This method scaled the input image at different scales and then extracted each scale's image features after scaling. Finally, all the obtained scale features were used to build the feature pyramid and input it into the neural network model. However, this method inputted the images of different scales into the recognition network model in parallel, reduced the computer processing speed and raised the computer performance

requirement. The multi-scale training method was based on the multi-scale feature extraction method. Firstly, a reasonable number of scales were fixed. In model training, a proportion was randomly selected for each iteration, and the input image was scaled down or enlarged according to the proportion. However, different proportions were used in the training process to improve the robustness of the network model and avoid excessive calculation of the problem in multi-scale feature extraction. Thus, our image input network was 640 × 640 pixels. The image was randomly scaled and input into the network model according to the three fixed ratios of 0.5, 1, and 2.

3.6. Optimization of Loss Function

The original YOLOv7 network adopted the Complete-Intersection-over-Union (CioU) Loss function, added the loss of the detection frame size based on the Distance-Intersection-over-Union (DioU) Loss function, added the loss of length and width, and made the prediction frame more consistent with the real frame. The calculation formulas [47–49] are as follows:

$$L_{CIoU} = 1 - CIoU, \tag{1}$$

$$CIoU = IoU - \frac{D^2}{C^2} - \alpha v, \tag{2}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2, \tag{3}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v}, \tag{4}$$

where D is the Euclidean distance between the prediction frame's center point and the real frame's center point, furthermore, C is the Euclidean distance of the diagonal of the minimum external rectangle between the prediction frame and the real frame; w^{gt} and h^{gt} are the width and height of the real box, respectively; and w and h are the width and height of the forecast box, respectively.

Thus, the CioU Loss function formula considered the overlap area, center point distance, and aspect ratio of boundary box regression. However, the aspect ratio (v) difference was not the real difference between width, height, and confidence, as this effectively hinders the similarity of model optimization. Given this problem, the Efficient-Intersection-over-Union (EioU) Loss was proposed to replace this study's original CioU Loss function. Based on CioU, EIoU disintegrated the influence factor of aspect ratio to calculate the length and width of the target and anchor frames, respectively. The goal was to minimize the difference between the width and height of the predicted frame and the real frame to accelerate the convergence speed, as shown in the following equation:

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(w, w^{gt}\right)}{C_w^2} + \frac{\rho^2\left(h, h^{gt}\right)}{C_h^2}, \tag{5}$$

where C_w and C_h are the width and height of the minimum external frame covering the two boxes, respectively.

3.7. Model Training

3.7.1. Test Platform

The deep learning training Center workstations in the school laboratory were used to test different algorithms. These workstations were run on a Windows 10 system platform, equipped with an NVIDIA Ge Force RTX2080Ti graphics card and the processor of Intel® Xeon® Silver 4210R. The main frequency of the CPU was 2.40 GHz with 64 GB RAM. The programming language was Python 3.8.5, the CUDA version was 11.3, and the deep

learning framework PyTorch was 1.10.0. All of the algorithms in this paper were trained by transfer learning.

The original "Yolov7.pt" was used as the weight for model training, and the experimental dataset was modified according to the dataset format required by YOLOv7. The initial learning rate, final OneCycleLR learning rate, weight attenuation, stochastic gradient descent momentum, box loss gain, cls loss gain, "batch size" (the number of samples selected in a single training), and "epochs" (learning step size) were set as 0.01, 0.2, 0.0005, 0.937, and 0.0, respectively. Then, 5, 0.3, 16, and 300 were selected for the multi-scale training mode. After the model training was completed, the weight obtained from the training was saved and used to evaluate the performance of the recognition model on the test set. After post-processing operations, such as non-maximum suppression, many redundant detection boxes were eliminated, and the detection result was the network's final output and had the highest confidence score. A concrete flowchart of the algorithm is shown in Figure 9.

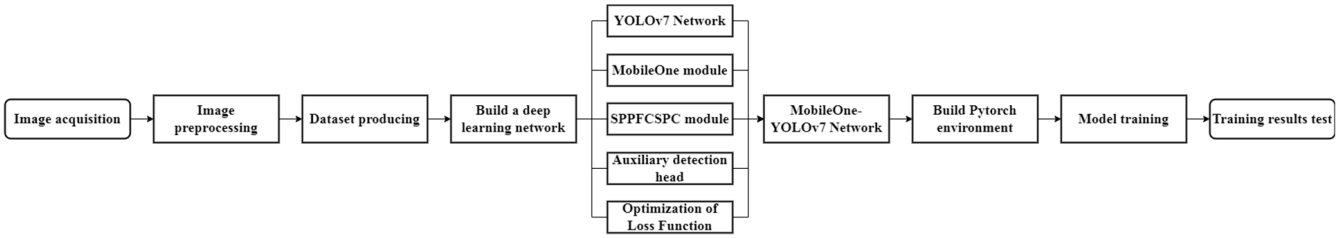

Figure 9. Algorithm flow chart.

3.7.2. Evaluation Indicators

Five index parameters, namely accuracy *P* (Precision, %), recall *R* (%), mean average precision (mAP), and *F*1, were used to evaluate the network model performance [50,51]. Its calculation is shown in the following equations:

$$P = \frac{TP}{TP + FP} \times 100\%, \tag{6}$$

$$R = \frac{TP}{TP + FN} \times 100\%, \tag{7}$$

$$AP = \int_0^1 P(R)dR, \tag{8}$$

$$mAP = \frac{1}{M} \sum_{k=1}^{M} AP(k) \times 100\%, \tag{9}$$

$$F1 = \frac{P \times R \times 2}{P + R}, \tag{10}$$

where *TP* is the positive prediction for a positive sample, *FP* is the negative sample prediction for a positive sample, *FN* is the positive sample prediction for a negative sample, *TN* is the negative sample prediction for a negative sample, *k* represents the current category, and *M* represents the number of categories.

As shown in the accuracy and recall formulas, the accuracy was expressed as the ratio predicted correctly of all predicted positive sample results. The recall was the ratio normally indicated by all the positive samples.

3.7.3. Training Process

In this experiment, the original weight of YOLOv7 was used for training, and 300 rounds were trained. The weight number of each training was recorded as 0–299. Then, the weight with the highest *p*-value was selected for every 50 rounds of training. Thus, we have seven

weight models with the highest weight models. Table 2 shows the evaluation indices of these models, revealing the superposition of training times. The *p*-value of the model showed an increasing trend (each time was not higher than the previous time). As shown in Figure 10, the values of accuracy *p*, recall rate *R*, *F*1, and mAP1 (mAP@0.5) fluctuated significantly in the first 30 iterations, revealing an increasing state was stable after 100 iterations, with slight fluctuation. This is because, in the training process, the learning rate was gradually reduced through the learning rate scheduling strategy. The adjustment process of the learning rate would lead to different convergence speeds of the model at different stages, resulting in fluctuations in the values of various indicators. The value of mAP2 (mAP@0.50–0.95) rose in the first 100 iterations, and after the completion of 100 iterations, the fluctuation range decreased and gradually became stable.

Table 2. Performance indicators of weights at each stage.

Number of Weight Model	*P*/%	*R*/%	mAP1/%	mAP2/%	*F*1/%
49	81.78	79.71	87.48	74.42	80.73
99	87.02	96.26	96.16	88.56	91.41
149	90.69	94.07	95.24	88.19	92.35
199	90.92	97.19	96.95	89.82	93.95
249	91.70	96.43	96.38	90.08	94.01
299	92.25	96.57	97.14	90.23	94.36
283	96.31	92.96	97.28	91.76	94.61

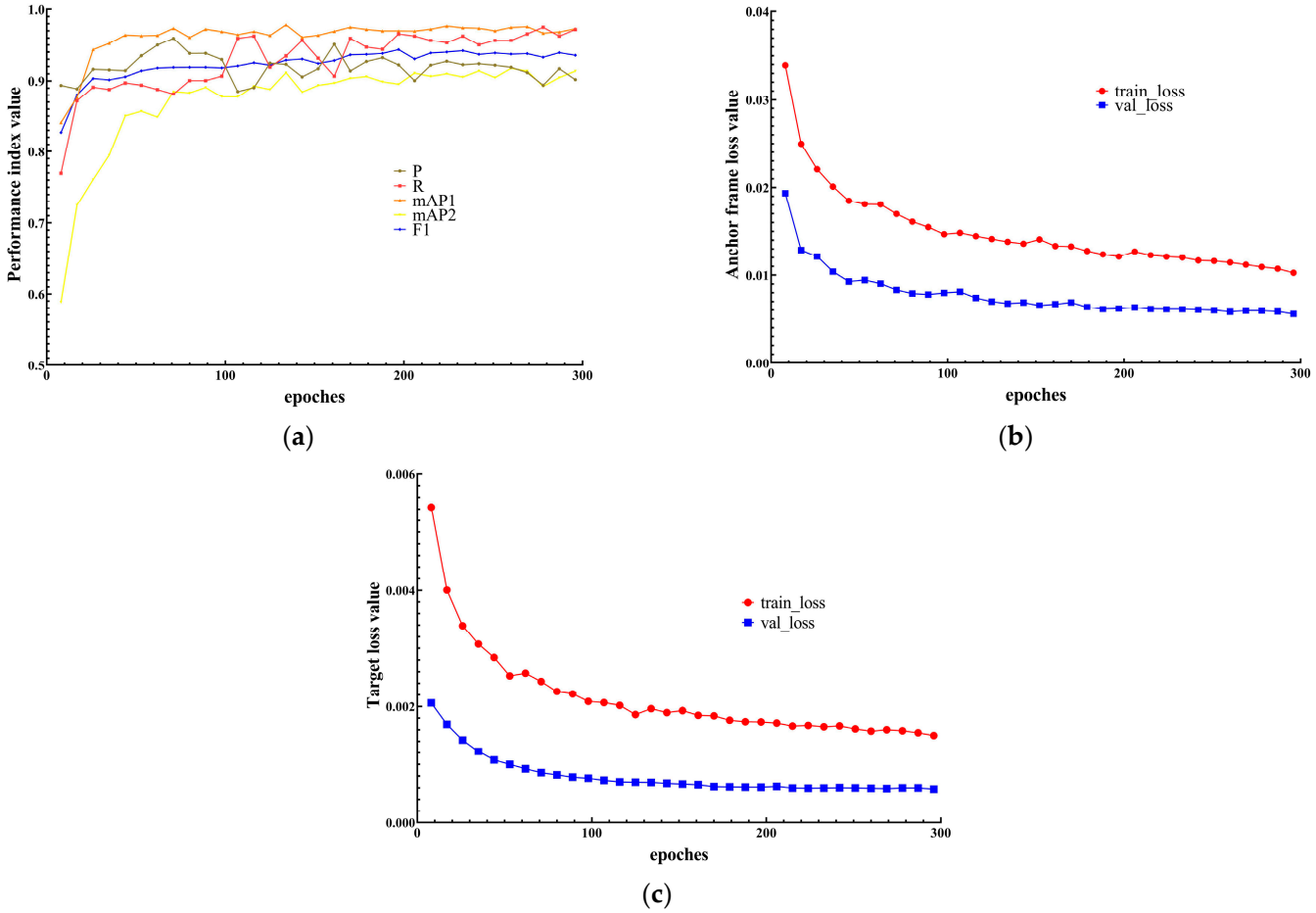

Figure 10. Performance index value and loss curve: (**a**) Performance index value, (**b**) Anchor frame loss curve, and (**c**) Target loss curve.

Nevertheless, the anchor frame loss value of the validation set and the target loss value of the validation set were still declining. The training set's anchor frame loss and target loss values tended to be smooth and stable after 200 iterations. At the same time, the decreasing range of the anchor frame loss value and target loss value of the training set gradually decreased and became stable, indicating that the model had completed the fitting after the 200th iteration.

3.7.4. Analysis of Training Data of Different Optimization Algorithms

The same training, validation, and test sets were used to train different optimization algorithms. After 300 iterations, various improved and optimized YOLOv7 algorithms obtained their best performance index parameters, as shown in Table 3.

Table 3. Effects of different optimization algorithms on the network model performance.

YOLOv7 Network	MobileOne	SPPFCSPC	Detection Head	*P*/%	*R*/%	mAP1/%	mAP2/%	*F*1/%
√	×	×	×	89.8	86.6	91.5	88.5	88.2
√	√	×	×	90.3	80.5	92.5	87.3	85.1
√	×	√	×	89.8	89.6	96.3	94.1	89.7
√	×	×	√	92.0	83.8	92.2	89.6	87.7
√	√	√	×	91.4	92.4	86.8	85.6	91.9
√	√	×	√	94.4	86.2	93.8	84.1	90.1
√	√	√	√	96.7	96.6	96.5	92.3	96.6

As shown in Table 3, adding MobileOne, SPPFCSPC, and a detection head to the original YOLOv7 network model would result in a varying gain. The accuracy of the MobileOne module, SPPFCSPC module, and detection head increased by 0.5, 0, and 2.2 percentage points, respectively, compared with the original network model when using the MobileOne module, SPPFCSPC module, and detection head separately. This result indicates that adding a detection head had the most significant impact on the accuracy of the network model. Compared with the original YOLOv7 network model, the YOLOv7+ detection head accuracy increased by 2.21 percentage points. However, improving network performance by using only one optimization algorithm was limited.

When the two optimization algorithms were adopted, the accuracy was improved significantly. As shown in Table 3, the MobileOne and SPPFCSPC module co-optimization accuracy increases by 1.6 percentage points. In comparison, co-optimizing the MobileOne module and detection head increased the accuracy by 4.6 percentage points.

Through experiments, we confirm that the co-optimization of the backbone network, neck network, and head network on YOLOv7 significantly increased the accuracy of the apple image recognition model. The accuracy of the improved YOLOv7+MobileOne+SPPFCSPC+ detection head algorithm also increased by 6.9%. The recall rate increased by 10%, mAP1 by 5%, and mAP2 by 3.8%. Figure 11 shows the training set anchor-frame loss curve, the training set target loss curve, the validation set anchor frame loss curve, and the validation set target loss curve of various optimization algorithms. In the training process, the proposed algorithm converged faster than other optimization algorithms in the validation set anchor frame loss and target loss, and the loss value was the lowest. The result showed that adding the MobileOne module, SPPFCSPC module, and detection head to the YOLOv7 network had a specific gain for target detection.

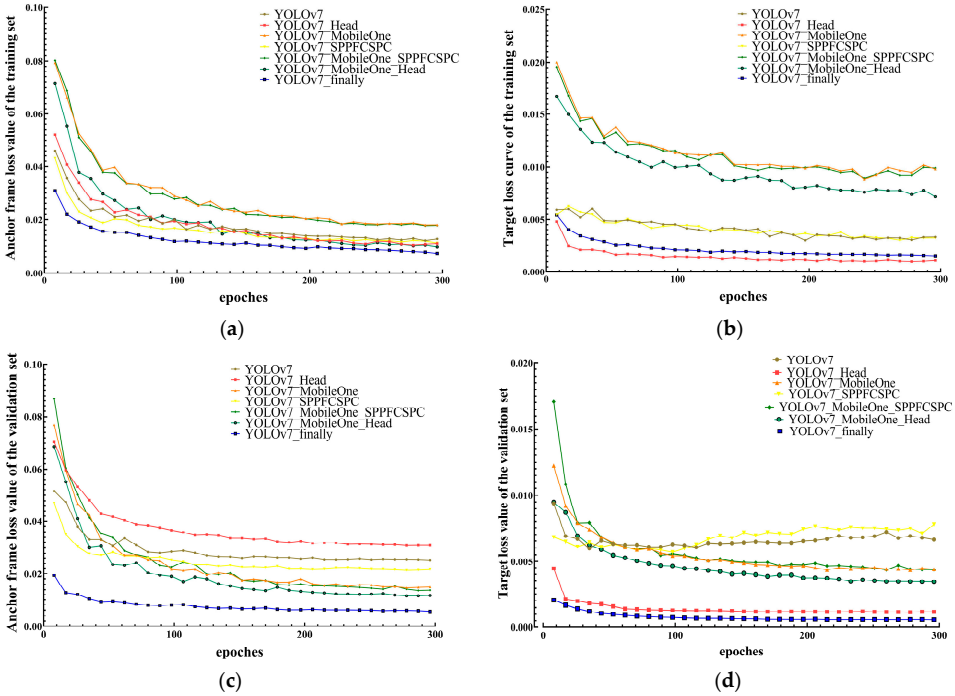

Figure 11. Loss curves of different combinations of optimization algorithms: (**a**) Anchor frame loss curve of the training set, (**b**) Target loss curve of the training set, (**c**) Anchor frame loss curve of the validation set, and (**d**) Target loss curve of the validation set.

4. Experimental Analysis

4.1. Evaluation of Test Results

The model obtained by the proposed algorithm was tested using the test set, and its *P*-*R* curve and *F*1 curve are shown in Figure 12. *F*1 is the harmonic mean of accuracy *P*, recall rate *R*, and its value changes with the change of confidence threshold. Figure 12 shows the change curve of the *F*1 value of the recognition model. When the confidence threshold was 0.66, the *F*1 value reached its peak of 96.65. The recall rate decreased with an increase in confidence threshold, and the missed targets increased. Therefore, the threshold was set at 0.5.

This study selected five conditions: smooth close-range light, close-range backlight, smooth light in dense scenes (the number of apples in the image was more than 30), the backlight in dense scenes (the number of apples in the image was more than 30), and large field-of-view scenes (complete view of the apple tree). The proposed and original YOLOv7 algorithms were used for a case test on the apple image.

The two algorithms were compared using false detection, missing detection, and confidence value. Figures 13 and 14 show the apple target recognition results based on YOLOv7 and the proposed algorithm, respectively.

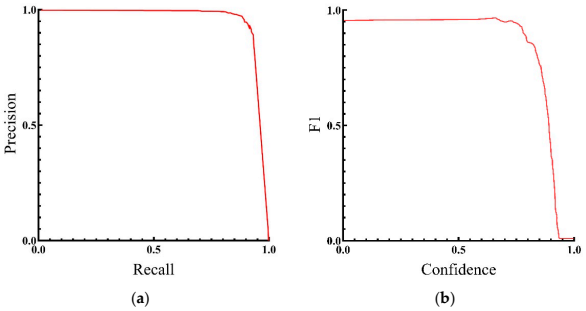

Figure 12. *P*-*R* curve and *F*1 curve under the test set: (**a**) *P*-*R* curve and (**b**) *F*1 curve.

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

(**e**)

Figure 13. Recognition results based on YOLOv7: (**a**) Close-range smooth light, (**b**) Close-range backlight, (**c**) Smooth light in dense scenes, (**d**) Backlight in dense scenes, and (**e**) Large field-of-view scenes.

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

(**e**)

Figure 14. Recognition results based on the algorithm: (**a**) Close-range smooth light, (**b**) Close-range backlight, (**c**) Smooth light in dense scenes, (**d**) Backlight in dense scenes, and (**e**) Large field-of-view scenes.

As shown in Figures 14 and 15, both the original and proposed algorithms had no missed or false detection under close-range conditions. However, the YOLOv7 algorithm was lower than the proposed algorithm in terms of confidence, indicating that the proposed algorithm had a better recognition effect on apple targets. From Table 4, both the proposed and YOLOv7 algorithms had missed detection cases in dense scenes. However, the number of apple targets missed by the proposed algorithm was far lower than that of the YOLOv7 algorithm. The YOLOv7 algorithm could not detect the overlapping and blocking of apple targets in dense scenes, and the confidence was low. Thus, the proposed algorithm

could effectively identify fruit overlapped occlusion and leaf and branch occlusion under normal circumstances, and its confidence was much higher than that of YOLOv7. However, the proposed algorithm could not accurately identify apple targets with severe occlusion (fruit targets were occluded by more than 50%), with an average missed detection rate of 12.68%. More apples were identified in the large field of view, both in the soft light and reverse light, showing the significant advantages of the proposed algorithm. YOLOv7 only detected 24 apples, and the missing rate increased by 67.74%, whereas our algorithm detected 87 apples, indicating that the proposed algorithm had higher robustness under complex conditions.

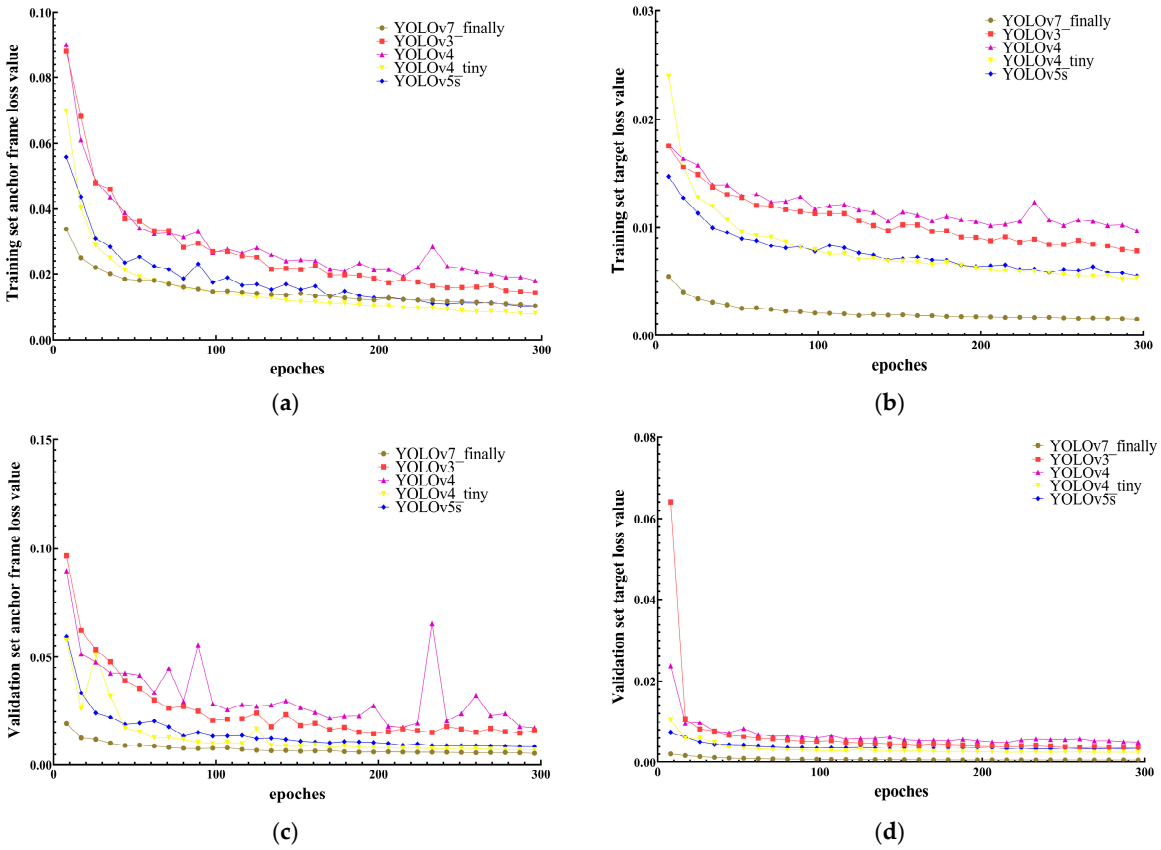

Figure 15. Loss curves of different algorithms: (**a**) Training set anchor frame loss curve, (**b**) Training set target loss curve, (**c**) Validation set anchor frame loss curve, and (**d**) Validation set target loss curve.

Table 4. Fruit detection in different scenarios.

Algorithm Model.	Scene Conditions	Actual Fruit Count	Correct Number to Check Out	Missing Count	Missed Detection Rate/%
YOLOv7	Close-range smooth light	6	6	0	0
	Close-range backlight	3	3	0	0
	Smooth light in a dense scene	58	29	29	50.00
	Backlight in dense scenes	51	21	30	58.82
	Large field-of-view scenes	93	24	69	74.19
Proposed algorithm	Close-range smooth light	6	6	0	0
	Close-range backlight	3	3	0	0
	Smooth light in a dense scene	58	47	11	23.40
	Backlight in dense scenes	51	50	1	1.96
	Large field-of-view scenes	93	87	6	6.45

4.2. Comparison of Recognition Results of Different Algorithms

The improved algorithm and four classical general algorithms, namely YOLOv3, YOLOv4, YOLOV4-Tiny, and YOLOv5s, were introduced for comparative analysis. While training these four algorithms, we used images of the same size, same training set, and validation set to test five different algorithms through the same test set. Table 5 shows the performance parameters of the five algorithms on the test set, and the change curve of their loss value is shown in Figure 15.

Table 5. The recognition effect of different algorithms on the test set.

Performance Indicators	YOLOv7-Finally	YOLOv3	YOLOv4	YOLOv4-Tiny	YOLOv5s
P/%	96.7	93.2	82.7	87.6	90.2
R/%	96.6	91.0	81.6	88.4	92.8
mAP1/%	96.5	95.8	84.4	91.5	90.9
mAP2/%	92.3	86.4	70.5	89.7	83.6
*F*1/%	96.7	92.1	82.2	88.0	91.5

Regarding accuracy, the proposed algorithm reached 96.7%; the recall rate *R* was 96.6%. The mAP1, mAP2, and *F*1 were 96.5%, 92.3%, and 96.7%, respectively, higher than the other four algorithms. The values of mAP1 and mAP2 were 5.6 and 8.7 percentage points, respectively, higher than that of the YOLOv5 model. As shown in Figure 15, the loss value of the convergence of the proposed algorithm was the lowest. Moreover, the proposed algorithm was superior to the other four algorithms regarding the training set anchor frame loss, the training set target loss, the validation set anchor frame loss, and the validation set target loss. After comparing the algorithms proposed in previous studies and their evaluation indices with the algorithm proposed in this paper, an algorithm from a previous study [21] was selected for comparison with the results of our algorithm. The *P*, *R*, and *F*1 values of our algorithm were 5.7, 4, and 4.9 percentage points higher than those of the algorithm proposed in that study [21]. In summary, the proposed algorithm had the best all-around performance.

5. Conclusions

This study proposes a multi-apple target recognition algorithm for high fruit density, overlapping fruit branches, and leaves in densely planted apple orchards with short stock. Based on the YOLOv7 model, we introduced the MobileOne module to realize the backbone network through parametric fusion and changed the image fusion mode from serial channel to parallel channel. Finally, we constructed a new recognition algorithm and added an auxiliary detection head to improve the model's accuracy for recognizing apple targets.

Moreover, we used the same training and validation sets to conduct multi-scale training on different improved algorithms and different model algorithms. Then, we used the same test set to calculate the accuracy *P*, recall rate *R*, the average precision mean mAP1 (mAP@0.5), average precision mean mAP2 (mAP@0.50–0.95), *F*1 value, and other five index parameters of different algorithms. Comparing different improved algorithms showed that the algorithm model had the best performance under the test set, and its accuracy rate and *F*1 value were 96.7% and 96.6%, respectively.

Compared with other YOLO algorithms, the improved YOLOv7 model increased its accuracy and *F*1 value by 6.9% and 8.4%, respectively. Moreover, its accuracy was 3.5, 14, 9.1, and 6.5 percentage points higher than that of YOLOv3, YOLOv4, YoloV4-Tiny, and YOLOv5, respectively. The YOLOv7's *F*1 value was 4.6, 14.5, 8.7, and 5.2 percentage points higher than those of the YOLOv3, YOLOv4, Yolov4-Tiny, and YOLOv5, respectively. To sum up, the proposed algorithm was more suitable for identifying apple targets in dense scenarios. Finally, the dataset will be further expanded and made available to other researchers, who will focus on improving the detection accuracy of apple fruits and laying the foundation for automated picking.

Author Contributions: Writing—Original Draft, H.Y.; Data Curation and Cartography, Y.L.; Writing—Review and Editing, S.W.; Supervision, H.Q.; Funding Acquisition, J.W. (Jinxing Wang); Project Administration, J.Q.; Model Training, N.L. and J.W. (Jie Wu); Methodology, Y.Y. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Shandong Province Key Research and Development Plan (major scientific and technological innovation project) Project, Agricultural Manipulator Motion Planning and Intelligent Drive Technology Research and Development (2022CXGC020701); Supported by the earmarked fund for CARS (CARS-27); Shandong Province Science and Technology Small and Medium-sized Enterprise Innovation Ability Promotion Project, New Orchard Pomegranate Intelligent Picking Manipulator Research and Development and Application (2022TSGC2253); 2022 Science and Technology Think Tank Youth Talent Plan Project, Analysis of The Current Situation, Problems and Countermeasures of Modern Orchard Technology in Shandong Province (20220615ZZ07110137).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The datasets generated during and analyzed during the current study are provided in the article here.

Acknowledgments: The author thanks the tutor, all the staff in the team for their guidance and help, and the Shandong provincial government for the financial support of the project. Finally, I am grateful to all those who devote much time to reading this thesis and giving me much advice, which will benefit me in my later study.

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

References

1. Otani, T.; Itoh, A.; Mizukami, H.; Murakami, M.; Yoshida, S.; Terae, K.; Tanaka, T.; Masaya, K.; Aotake, S.; Funabashi, M.; et al. Agricultural Robot under Solar Panels for Sowing, Pruning, and Harvesting in a Synecoculture Environment. *Agriculture* **2022**, *13*, 18. [CrossRef]
2. Vrochidou, E.; Tsakalidou, V.N.; Kalathas, I.; Gkrimpizis, T.; Pachidis, T.; Kaburlasos, V.G. An Overview of End Effectors in Agricultural Robotic Harvesting Systems. *Agriculture* **2022**, *12*, 1240. [CrossRef]
3. Fan, P.; Lang, G.; Guo, P.; Liu, Z.; Yang, F.; Yan, B.; Lei, X. Multi-Feature Patch-Based Segmentation Technique in the Gray-Centered RGB Color Space for Improved Apple Target Recognition. *Agriculture* **2021**, *11*, 273. [CrossRef]
4. Fan, P.; Lang, G.; Yan, B.; Lei, X.; Guo, P.; Liu, Z.; Yang, F. A Method of Segmenting Apples Based on Gray-Centered RGB Color Space. *Remote Sens.* **2021**, *13*, 1211. [CrossRef]
5. Fan, P.; Yan, B.; Wang, M.; Lei, X.; Liu, Z.; Yang, F. Three-finger grasp planning and experimental analysis of picking patterns for robotic apple harvesting. *Comput. Electron. Agric.* **2021**, *188*, 106353. [CrossRef]
6. 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*, 105687. [CrossRef]
7. Duan, L.; Yang, F.; Yan, B.; Shi, S.; Qin, J. Research progress of apple production intelligent chassis and weeding and harvesting equipment technology. *Smart Agric.* **2022**, *4*, 24–41.
8. Wang, N.; Joost, W.; Zhang, F. Towards sustainable intensification of apple production in China-Yield gaps and nutrient use efficiency in apple farming systems. *J. Integr. Agric.* **2016**, *15*, 716–725. [CrossRef]
9. Bulanon, D.; Kataoka, T. Fruit detection system and an end effector for robotic harvesting of Fuji apples. *Agric. Eng. Int. CIGR E-J.* **2010**, *12*, 203–210.
10. Gongal, A.; Amatya, S.; Karkee, M.; Zhang, Q.; Lewis, K. Sensors and systems for fruit detection and localization: A review. *Comput. Electron. Agric.* **2015**, *116*, 8–19. [CrossRef]
11. Lv, J.; Zhao, D.; Ji, W. Fast tracing recognition method of target fruit for apple harvesting robot. *Trans. Chin. Soc. Agric. Mach.* **2014**, *45*, 65–72.
12. Mai, C.; Zheng, L.; Xiao, C.; Li, M. Comparison of apple recognition methods under natural light. *J. China Agric. Univ.* **2016**, *21*, 43–50.
13. Si, Y.; Qiao, J.; Liu, G.; Gao, R.; He, B. Recognition and location of fruits for appleharvesting robot. *Trans. Chin. Soc. Agric. Mach.* **2010**, *41*, 148–153.
14. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
15. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
16. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [CrossRef]

17. Cardellicchio, A.; Solimani, F.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. *Comput. Electron. Agric.* **2023**, *207*, 107757. [CrossRef]
18. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516. [CrossRef]
19. Altaheri, H.; Alsulaiman, M.; Muhammad, G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access* **2019**, *7*, 117115–117133. [CrossRef]
20. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [CrossRef]
21. Zhao, H.; Qiao, Y.; Wang, H.; Yue, Y. Apple fruit recognition in complex orchard environment based on improved YOLOv3. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 127–135.
22. Yang, F.; Lei, X.; Liu, Z.; Fan, P.; Yan, B. Fast Recognition Method for Multiple Apple Targets in Dense Scenes Based on CenterNet. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 265–273.
23. Zheng, T.; Jiang, M.; Feng, M. Vision based target recognition and location for picking robot: A review. *Chin. J. Sci. Instrum.* **2021**, *42*, 28–51.
24. Wu, W.; Yang, T.; Li, R.; Chen, C.; Zhou, K.; Sun, M.; Li, C.; Zhu, X.; Guo, W. Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales. *J. Integr. Agric.* **2020**, *19*, 1998–2008. [CrossRef]
25. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once:unified, realtime object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 91, pp. 779–788.
26. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition IEEE, Honolulu, HI, USA, 21–26 June 2017; pp. 6517–6525.
27. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. In *Computer Vision and Pattern Recongintion*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 276, pp. 126–134.
28. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *Comput. Vis. Pattern Recognit.* **2020**, *10*, 34–51.
29. Mekhalfi, M.L.; Nicolò, C.; Bazi, Y.; Al Rahhal, M.M.; Alsharif, N.A.; Al Maghayreh, E. Contrasting YOLOv5, Transformer, and EfficientDet Detectors for Crop Circle Detection in Desert. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
30. Zeng, T.; Li, S.; Song, Q.; Zhong, F.; Wei, X. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. *Comput. Electron. Agric.* **2023**, *205*, 107625. [CrossRef]
31. Shi, R.; Li, T.; Yasushi, Y. An attribution-based pruning method for real-time mango detection with YOLO network. *Comput. Electron. Agric.* **2020**, *169*, 105214. [CrossRef]
32. Ying, S.; Huang, S.; Chang, S.; Yang, Z.; Feng, Z.; Guo, N. Convolutional and Transformer Based Deep Neural Network for Automatic Modulation Classification. *China Commun.* **2023**, *20*, 135–147. [CrossRef]
33. Zhang, Q.; Ma, W.; Wang, Y.; Zhang, Y.; Shi, Z.; Li, Y. Backdoor Attacks on Image Classification Models in Deep Neural Networks. *Chin. J. Electron.* **2022**, *31*, 199–212. [CrossRef]
34. Dai, G.; Fan, J.; Tian, Z.; Wang, C. PPLC-Net:Neural network-based plant disease identification model supported by weather data augmentation and multi-level attention mechanism. *J. King Saud Univ.—Comput. Inf. Sci.* **2023**, *35*, 101555. [CrossRef]
35. Wei, J.; Wang, Q.; Song, X.; Zhao, Z. The Status and Challenges of Image Data Augmentation Algorithms. *J. Phys. Conf. Ser.* **2023**, *2456*, 012041. [CrossRef]
36. Wang, C.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**. [CrossRef]
37. Zhou, J.; Zhang, Y.; Wang, J. A Dragon Fruit Picking Detection Method Based on YOLOv7 and PSP-Ellipse. *Sensors* **2023**, *23*, 3803. [CrossRef]
38. Roy, A.M.; Bhaduri, J. Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4. *Comput. Electron. Agric.* **2022**, *193*, 106694. [CrossRef]
39. Piao, Y.; Jiang, Y.; Zhang, M.; Wang, J.; Lu, H. PANet: Patch-Aware Network for Light Field Salient Object Detection. *IEEE Trans. Cybern.* **2021**, *53*, 379–391. [CrossRef]
40. Hong, F.; Tay, D.; Wei, L.; Ang, A. Intelligent Pick-and-Place System Using MobileNet. *Electronics* **2023**, *12*, 621. [CrossRef]
41. Li, X.; Ye, H.; Qiu, S. Cloud Contaminated Multispectral Remote Sensing Image Enhancement Algorithm Based on MobileNet. *Remote Sens.* **2022**, *14*, 4815. [CrossRef]
42. Sheng, G.; Sun, S.; Liu, C.; Yang, Y. Food recognition via an efficient neural network with transformer grouping. *Int. J. Intell. Syst.* **2022**, *37*, 11465–11481. [CrossRef]
43. Wang, Q.; Cheng, M.; Huang, S.; Cai, Z.; Zhang, J.; Yuan, H. A deep learning approach incorporating YOLO v5 and attention mechanisms for field real-time detection of the invasive weed Solanum rostratum Dunal seedlings. *Comput. Electron. Agric.* **2022**, *199*, 107194. [CrossRef]
44. Wei, D.; Chen, J.; Luo, T.; Long, T.; Wang, H. Classification of crop pests based on multi-scale feature fusion. *Comput. Electron. Agric.* **2022**, *194*, 106736. [CrossRef]
45. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Cai, W.; Yang, N.; Wang, B. Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification. *Expert Syst. Appl.* **2023**, *223*, 119858. [CrossRef]

46. Yang, Y.; Sun, S.; Huang, J.; Huang, T.; Liu, K. Large-Scale Aircraft Pose Estimation System Based on Depth Cameras. *Appl. Sci.* **2023**, *13*, 3736. [CrossRef]
47. Ding, J.; Cao, H.; Ding, X.; An, C. High Accuracy Real-Time Insulator String Defect Detection Method Based on Improved YOLOv5. *Front. Energy Res.* **2022**, *10*, 898. [CrossRef]
48. Gao, J.; Yang, T. Face detection algorithm based on improved TinyYOLOv3 and attention mechanism. *Comput. Commun.* **2021**, *181*, 329–337. [CrossRef]
49. Qi, J.; Zhang, J.; Meng, Q. Auxiliary Equipment Detection in Marine Engine Rooms Based on Deep Learning Model. *J. Mar. Sci. Eng.* **2021**, *9*, 1006. [CrossRef]
50. Amarasingam, N.; Gonzalez, F.; Salgadoe, A.S.A.; Sandino, J.; Powell, K. Detection of White Leaf Disease in Sugarcane Crops Using UAV-Derived RGB Imagery with Existing Deep Learning Models. *Remote Sens.* **2022**, *14*, 6137. [CrossRef]
51. Li, J.; Chen, L.; Shen, J.; Xiao, X.; Liu, X.; Sun, X.; Wang, X. Improved Neural Network with Spatial Pyramid Pooling and Online Datasets Preprocessing for Underwater Target Detection Based on Side Scan Sonar Imagery. *Remote Sens.* **2023**, *15*, 440. [CrossRef]