agronomy

MDPI

Article

RDE-YOLOv7: An Improved Model Based on YOLOv7 for Better Performance in Detecting Dragon Fruits

Jialiang Zhou [1], Yueyue Zhang [1] and Jinpeng Wang [1,2,*]

[1] School of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China; zhoujialiang@njfu.edu.cn (J.Z.)
[2] Co-Innovation Center of Efficient Processing and Utilization of Forest Resources, Nanjing Forestry University, Nanjing 210037, China
* Correspondence: jpwang@njfu.edu.cn; Tel.: +86-025-85427765

Abstract: There is a great demand for dragon fruit in China and Southeast Asia. Manual picking of dragon fruit requires a lot of labor. It is imperative to study the dragon fruit-picking robot. The visual guidance system is an important part of a picking robot. To realize the automatic picking of dragon fruit, this paper proposes a detection method of dragon fruit based on RDE-YOLOv7 to identify and locate dragon fruit more accurately. RepGhost and decoupled head are introduced into YOLOv7 to better extract features and better predict results. In addition, multiple ECA blocks are introduced into various locations of the network to extract effective information from a large amount of information. The experimental results show that the RDE-YOLOv7 improves the precision, recall, and mean average precision by 5.0%, 2.1%, and 1.6%. The RDE-YOLOv7 also has high accuracy for fruit detection under different lighting conditions and different blur degrees. Using the RDE-YOLOv7, we build a dragon fruit picking system and conduct positioning and picking experiments. The spatial positioning error of the system is only 2.51 mm, 2.43 mm, and 1.84 mm. The picking experiments indicate that the RDE-YOLOv7 can accurately detect dragon fruits, theoretically supporting the development of dragon fruit-picking robots.

Keywords: dragon fruit detection; YOLOv7; RepGhost; decoupled head; ECA; picking robots

Citation: Zhou, J.; Zhang, Y.; Wang, J. RDE-YOLOv7: An Improved Model Based on YOLOv7 for Better Performance in Detecting Dragon Fruits. *Agronomy* **2023**, *13*, 1042. https://doi.org/10.3390/agronomy13041042

Academic Editors: Zhanyou Xu, Lizhi Wang and Reka Howard

Received: 15 March 2023
Revised: 28 March 2023
Accepted: 30 March 2023
Published: 31 March 2023

1. Introduction

China is a large agricultural country. Although it has a very large agricultural planting area, agricultural production has not yet achieved specialization and scale. The development of smart agriculture is an inevitable trend in the future [1]. China is not only one of the largest dragon fruit producers, but also one of the largest dragon fruit consumers. However, such a large area of dragon fruit is almost harvested manually, which leads to a large labor demand. Due to the hard branch, dragon fruit is usually harvested manually by cutting its branch connected to the root. Such a manual picking method not only requires special tools, but also may cause injury to workers. Therefore, it is imperative to study the automatic picking robot of dragon fruit.

Automatic fruit picking is an important part of intelligent agriculture, which can greatly reduce the labor intensity of workers. Fruit-picking robot is becoming one of the hottest topics in recent years [2–5]. The vision system is an important part of the fruit-picking robot, which determines the path and picking point of the picking hand. Only by accurately identifying and locating the fruit can we design the picking path and realize automatic picking. Early fruit recognition mostly uses traditional image processing methods. Most of these methods set some features artificially and then determine whether the image conforms to these set features, such as color features, shape features, and texture features. The accuracy of fruit recognition using these methods is generally low, and the generalization ability is weak, which makes it difficult to apply to picking robots. In recent years, with the development of deep learning, more and more fruit recognition

methods based on deep learning have been proposed [6–13]. Different fruit detection tasks may have different problems, such as small targets, occlusion, false detection, and complex background. Researchers should design corresponding solutions based on actual scenarios. Object detection models based on convolution neural networks (CNNs) are the most commonly used method. They are mainly divided into two types: one-stage and two-stage. Two-stage models first generate some candidate regions through Region Proposal Network (RPN), and then classify and locate targets through a convolution neural network. Its representative models include region-based convolutional neural network (RCNN) series, and spatial pyramid pooling Network (SPPNet). For example, Liu et al. [14] proposed an improved Mask RCNN to solve the problem that the characteristics of leaves and fruits are very similar in cucumber fruit detection. Hu et al. [15] used Faster RCNN to detect pecan fruit, and the mean average precision of this method reached 95.932%. Wan et al. [16] proposed a deep learning framework for multi-class fruits detection based on improved Faster R-CNN, this detection method got higher accuracy and speed compared with the original model. These above models generally have high accuracy and slow detection speed due to two stages in detection. One-stage models directly extract features through a convolution neural network to predict target classification and location. Its representative models include the you only look once (YOLO) series, single shot detection (SSD), and RetinaNet. Gai et al. [17] proposed an improved YOLOv4 to detect cherry fruits, and improved the detection speed and accuracy. Xu et al. [18] proposed an improved CBF module and replaced the bottleneck CSP module with the Specter module, which improved the detection accuracy and detection speed of the YOLOv5s model. Similarly, the CIoU Loss function was used in training YOLOv5s to decrease the missed detection rate and false detection rate [19]. Kang et al. [20] proposed A3N which is a geometry-aware network to perform end-to-end instance segmentation and grasping estimation using both color and geometry sensory data from an RGB-D camera. The instance segmentation accuracy of A3N achieved 0.873. Cardellicchio et al. [21] used YOLOv5 to effectively identify nodes, fruit, and flowers on a challenging dataset, and achieved high scores. These one-stage models may have had low accuracy in the past, but with the continuous improvement of scientific researchers, the one-stage models can now maintain high accuracy and high speed at the same time.

As for the detection of dragon fruits, some researchers have made contributions. For example, the backbone of YOLOv4 was replaced with Mobilenet-v3 to detect dragon fruits, thereby reducing the model size and improving speed. However, this method makes the average precision of the model decrease slightly [22]. Zhang et al. [23] proposed an improved YOLOv5s to accurately detect dragon fruits under different light conditions. These methods for detecting dragon fruits cannot apply to picking robots, because they only detect dragon fruits. In real scenes, the diverse postures of dragon fruit make it difficult to pick by cutting the branch. In this paper, we classify dragon fruit according to postures in the camera view. Based on the latest YOLOv7 model, this paper proposes RDE-YOLOv7 to improve the accuracy of dragon fruit detection, thus improving the performance of the dragon fruit-picking robot. We use the RDE-YOLOv7 to build a dragon fruit-picking system, and carry out spatial positioning verification and picking experiments.

The main contributions of this paper are as follows:

1. Dragon fruits are classified into two categories according to their postures in the camera view. Some existing methods in deep learning are applied to YOLOv7 to improve detection accuracy. The introduction of RepGhost and decoupled head are proved to be effective.
2. We compared the impact of three attention mechanisms on the performance of the model at different locations of the network, and finally added ECA blocks to the model to improve the detection accuracy.
3. We have built a dragon fruit picking system, and the validity of this method is proved by the spatial positioning verification experiment and the picking experiment.

2. Materials and Methods

2.1. Data Acquisition

The dragon fruit pictures used in this article were taken from orchard Lile agricultural, Nanjing city. They are taken under different weather conditions (cloudy, sunny), daytimes (morning, noon, evening, and night), lighting conditions (strong light, weak light, and artificial light), and shooting distance. Due to the need of picking, we classify dragon fruit into two categories according to the postures of dragon fruit in the camera. First, when a dragon fruit grows on the left or right side of its branch, it is named the dragon fruit in the side (DF_S). Second, when a dragon fruit grows on the front side of its branch, it is named as the dragon fruit in the front (DF_F). The location of dragon fruit and their categories were labeled by using LabelImg software before the experiment. In the outdoors, the variability of light is the main factor affecting the detection performance of the model. Therefore, we group the original images according to strong light, weak light, and artificial light. The number of them is 876, 883, and 729, respectively. At last, we divide the image data into training sets, validation sets, and test sets according to the three light conditions. The training set is used to train the model, the validation set is used to determine whether the training result is the best, and the test set is used to test and compare the detection performance of each model. Table 1 shows the data sets used in this paper.

Table 1. Datasets constructed under different lighting conditions.

Light Conditions	Training Set	Validation Set	Test Set
Strong light	699	65	112
Weak light	715	71	97
Artificial light	582	60	87

2.2. Data Augmentation

The diversity of data ensures better robustness of the model. To improve the robustness and generalization ability of the model in the experiment, data augmentation of the training set is needed to improve the learning effect. HSV augmentation, image translation, image scale, and image flip (left and right) are used to increase the diversity of training data. In addition, mixup and mosaic data augmentation are also used in training. Their goal is to have more targets on the newly generated images, to make the data more diversified. The mosaic data augmentation method can also generate more small targets, thus enhancing the ability of the model to detect small targets. The image generated by the mosaic data augmentation method is quite different from the actual one, so it is not used in the later stage of training. Figure 1 shows the process of mixup and mosaic data augmentation.

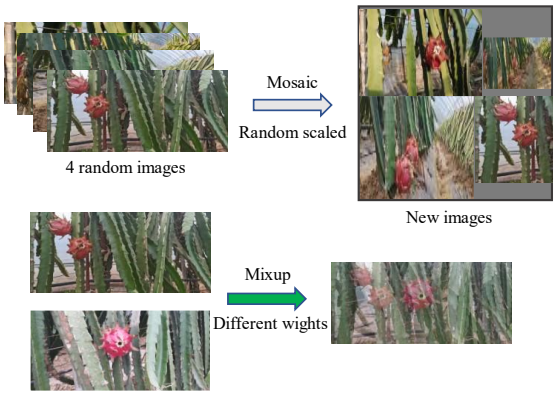

Figure 1. Process of mixup and mosaic data augmentation.

The hyper-parameters of the above data augmentation methods used in the experiment are shown in Table 2.

Table 2. The hyper-parameter of data augmentation.

Methods	Parameter
Hue	0.015 fraction
Saturation	0.7 fraction
Value	0.4 fraction
Translation	+/−20%
Scale	+/−10%
Flip	50% probability
Mosaic	0.8 probability
Mixup	0.15 probability

2.3. Methodology

2.3.1. YOLOv7

YOLO is an object detection model widely used in harvesting robots. It may have various problems when detecting different objects and different scenes [24]. Therefore, researchers need to improve the model according to the actual scene needs to adapt it to different scenarios. At present, the improvement direction of the model is mainly to improve the detection accuracy and detection speed. YOLOv7 is a relatively advanced object detector at present [25]. It has shown very good results on some public datasets. Figure 2 shows the network structure of YOLOv7. The structure of YOLOv7 is mainly divided into a backbone network and a head network. The original image enters the backbone network as input after some preprocessing. The backbone network is mainly used to extract features, and the head network is mainly used to further fuse the extracted features. The ELAN structure in the backbone network makes the deeper network effectively learn and converge by controlling the shortest and longest gradient path. The MP1 structure integrates two down-sampling methods: pooling and convolution, which allows the network to choose the better of the two down-sampling methods. In the head network, a structure named as ELAN-W is used to learn features after fusion at different scales, and this structure has two more gradient paths than the ELAN structure.

Figure 2. The structure of the YOLOv7 network. The Conv denotes the convolution layer, the BN denotes the batch normalization layer, and Silu and Sigmoid denote the activation function.

2.3.2. Attention Mechanism

In the convolutional neural network, the attention mechanism can assign different weights to different parts of the input feature map, to select the most important information from a large amount of information. Efficient channel attention (ECA) [26] is an efficient attention mechanism as shown in Figure 3. It is a local cross-channel interaction strategy without dimensionality reduction, which can be efficiently implemented via one-dimensional convolution. Furthermore, the convolution kernel size of one-dimensional convolution can be adjusted adaptively according to the number of channels. This attention mechanism can improve the detection accuracy of the model with few additional parameters.

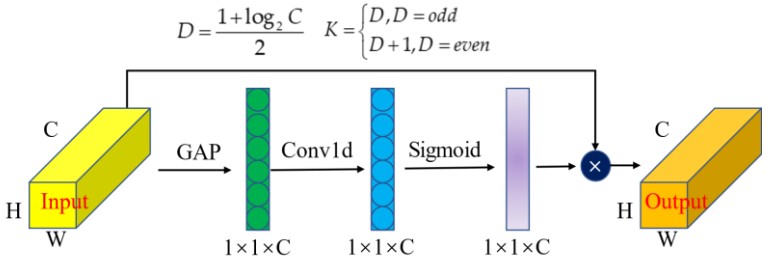

Figure 3. The structure of the ECA block.

In an efficient channel attention block, the input feature first gets the channel vector through the adaptive average pooling layer, then uses the one-dimensional convolution to interact with the information between channels, and finally uses the sigmoid activation function to obtain the weight value of each channel. The input feature adjusts the valof on the channel dimension according to the obtained weight.

2.3.3. RepGhost Module

The RepGhost module is a lightweight convolution module [27]. It utilizes a re-parameterization technique to realize feature reuse implicitly by replacing the inefficient concatenation operator in the Ghost module. Their research proves that Concat operation is more time-consuming than Add operation. Therefore, the RepGhost module replaces the Concat operation in the original Ghost module with the Add operation and uses the activation function after the Add operation to meet the rule of re-parameterization structure. Figure 4 shows the RepGhost structure.

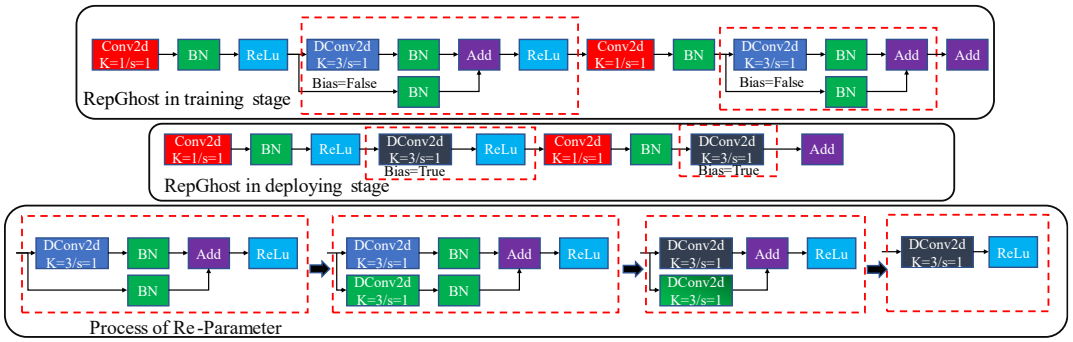

Figure 4. The structure of RepGhost and the process of re-parameter. During the training stage, there are BN branches. During deploying stage, there are only the convolution layer and activation layer. The process of the re-parameter is generating an equivalent convolution layer on the BN branch, fusing the convolution layer and BN layer, and fusing convolution layers on the two branches.

2.3.4. Decoupled Head

In the detection head of YOLOv7, classification, regression, and prediction are carried out simultaneously. Decoupled Head was proposed in the YOLOX model [28]. Compared

with the head of YOLOv7, the decoupled head uses features for classification, regression, and prediction, respectively. More convolution layers are added to the decoupled head for classification, regression, and prediction. Of course, the decoupled head will have more learning parameters to make the detection accuracy higher. Figure 5 shows the comparison of decoupled head and coupled head.

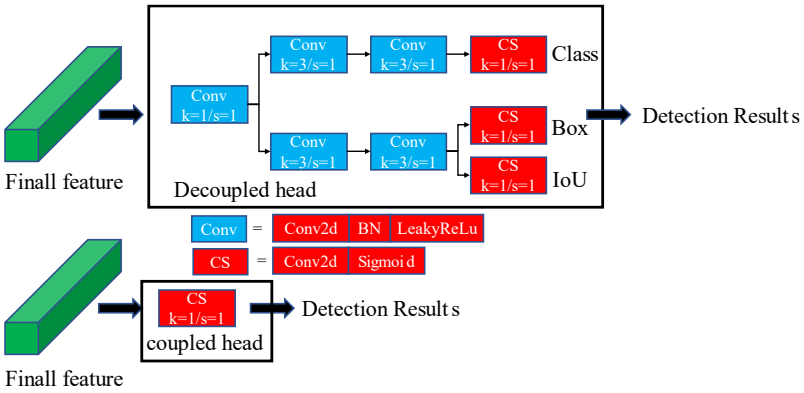

Figure 5. The structure of the decoupled head and coupled head. Compared to coupled head, the decoupled head has more convolution layers.

2.3.5. RDE-YOLOv7

Figure 6 depicts the RDE-YOLOv7 structure, highlighting the methods employed to improve the detection accuracy of the YOLOv7 model. Table 3 shows the output size of each layer.

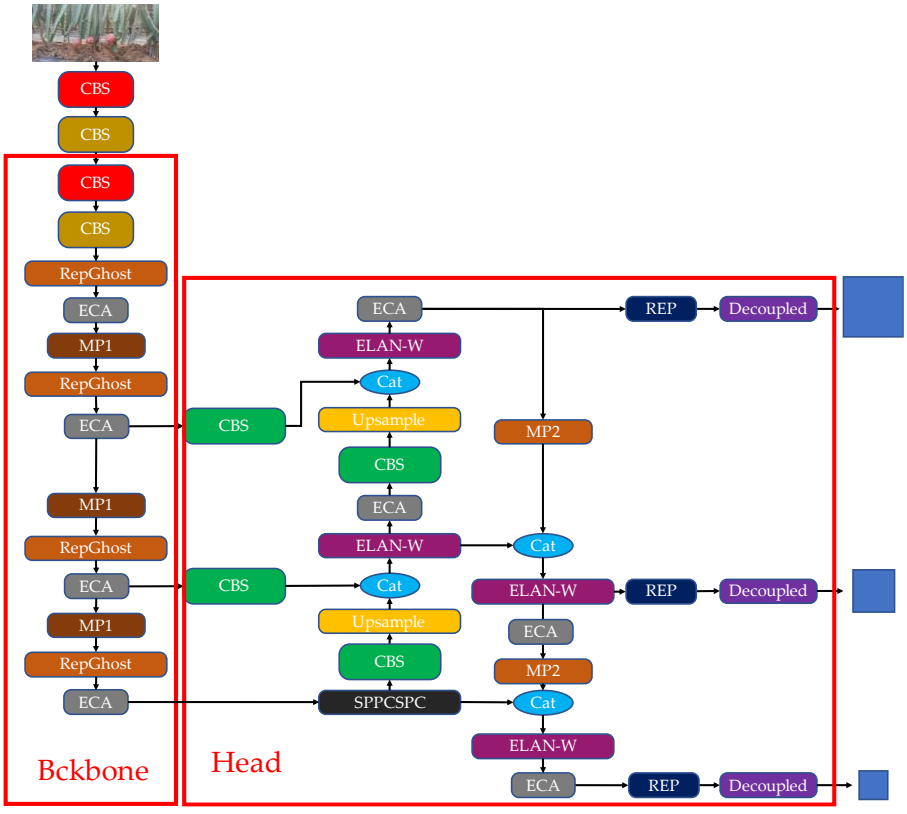

Figure 6. The structure of RDE-YOLOv7. RepGhost is used to replace the ELAN structure in the backbone. Decoupled head is used in the prediction layers. ECA is introduced into the backbone network and head network.

Table 3. The structure parameter of the RDE-YOLOv7 network.

Layer ID	Layer Name	ID of the Input Layer	Output Size
0	CBS	-	32 × 640 × 640
1	CBS	0	64 × 320 × 320
2	CBS	1	64 × 320 × 320
3	CBS	2	128 × 160 × 160
4	RepGhost	3	256 × 160 × 160
5	ECA	4	256 × 160 × 160
6	MP1	5	256 × 80 × 80
7	RepGhost	6	512 × 80 × 80
8	ECA	7	512 × 80 × 80
9	MP1	8	512 × 40 × 40
10	RepGhost	9	1024 × 40 × 40
11	ECA	10	1024 × 40 × 40
12	MP1	11	1024 × 20 × 20
13	RepGhost	12	1024 × 20 × 20
14	ECA	13	1024 × 20 × 20
15	SPPCSPC	14	512 × 20 × 20
16	CBS	15	256 × 20 × 20
17	Upsample	16	256 × 40 × 40
18	CBS	11	256 × 40 × 40
19	Concat	17, 18	512 × 40 × 40
20	ELAN-W	19	256 × 40 × 40
21	ECA	20	256 × 40 × 40
22	CBS	21	128 × 40 × 40
23	Upsample	22	128 × 80 × 80
24	CBS	8	128 × 80 × 80
25	Concat	23, 24	256 × 80 × 80
26	ELAN-W	25	128 × 80 × 80
27	ECA	26	128 × 80 × 80
28	MP2	27	256 × 40 × 40
29	Concat	20,28	512 × 40 × 40
30	ELAN-W	29	256 × 40 × 40
31	ECA	30	256 × 40 × 40
32	MP2	31	512 × 20 × 20
33	Concat	15, 32	1024 × 20 × 20
34	ELAN-W	33	512 × 20 × 20
35	ECA	34	512 × 20 × 20
36	REP	27	256 × 80 × 80
37	REP	31	512 × 40 × 40
38	REP	35	1024 × 20 × 20
39	Decoupled head	36, 37, 38	3 × (2 + 4 + 1) × 20 × 20 3 × (2 + 4 + 1) × 40 × 40 3 × (2 + 4 + 1) × 80 × 80

2.4. Evaluation Metrics

To evaluate the performance of the model, Precision (P), recall (R), and mean average precision (mAP) are used. These above metrics are calculated as follows:

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

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

$$AP = \int_0^1 P(R)dR \tag{3}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \quad (4)$$

where TP is the number of positive samples predicted to be positive class. FP is the number of negative samples predicted to be positive class. FN is the number of positive samples predicted to be negative class. AP is the area below the PR curve. mAP is the mean AP value for each category; n represents the number of categories in object detection. In this paper n = 2.

2.5. Training Parameters and Experimental Environment

The training hyperparameters used in the experiment are shown in Table 4.

Table 4. The hyperparameters in the training process.

Hyperparameters	Value
Epoch	300
Initial learning rate	0.01
Batch size	8
momentum	0.937
Weight decay	0.0005
Loss function	BCE, CIoU

The GPU of the computer is an NVIDIA GeForce RTX3070Ti. The CPU is AMD Ryzen 7 5800X with an 8-core processor. The operating system was Windows 10 and PyTorch version 1.10, Python version 3.8, and CUDA version 11 were used.

3. Results

3.1. Experiments of Some Methods Applying to YOLOv7

To improve the detection performance of the YOLOv7 model, some methods that may influence the detection performance have been used. By comparing with the original YOLOv7 model, whether these methods have a positive effect on improving the detection performance of YOLOv7. They are trained and tested under the same condition. Figure 7 shows the variation curves of training loss and validation loss.

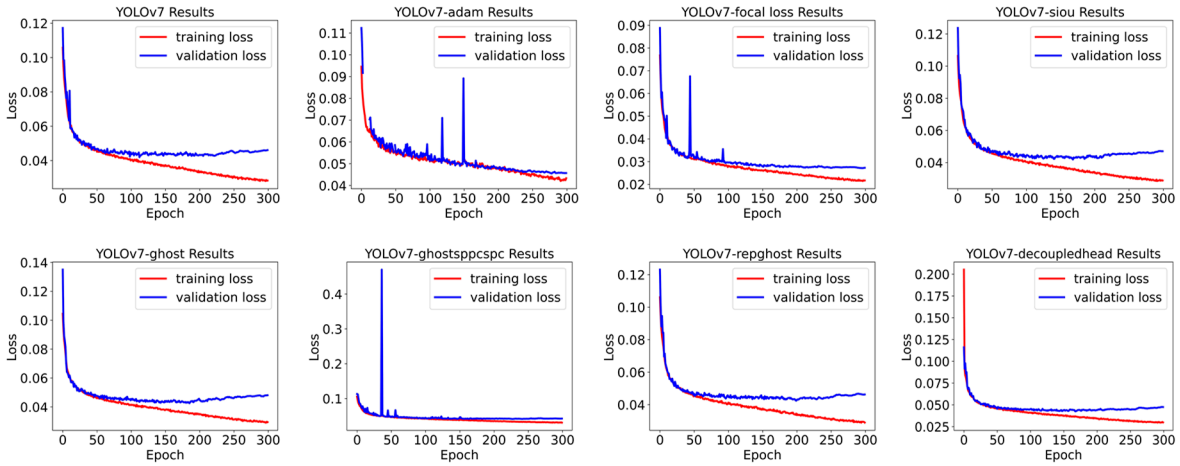

Figure 7. The training process of the different methods introduced into YOLOv7.

As shown in Figure 7, in 300 epoch training, although the loss on the training set still has a downward trend, the loss curve of all the above models on the validation set tends to be flat, indicating that the model training has converged. We tested these trained models, and the results are shown in Table 5.

Table 5. The hyperparameters in the training process.

Methods	*P*(%)	*R*(%)	*mAP*(%)
YOLOv7	85.7	88.7	91.8
YOLOv7-Adam	81.5	88.1	91.1
YOLOv7-SIoU	83.5	90.3	91.5
YOLOv7-Focal	84.0	87.7	91.6
YOLOv7-Ghost	86.1	88.3	91.4
YOLOv7-GhostSPPCSPC	85.8	89.5	92.1
YOLOv7-RepGhost	86.9	88.8	92.3
YOLOv7-Decoupled	88.4	87.9	92.3

YOLOv7-Adam means training with Adam optimizer, YOLOv7-SIoU means training with SIoU box regression loss function, YOLOv7-Focal means training with Focal loss function, YOLOv7-Ghost means replacing ELAN with Ghostbottleneck, YOLOv7-GhostSPPCSPC means replacing SPPCSPC with GhostSPPCSPC, YOLOv7-RepGhost means replacing ELAN with RepGhost, YOLOv7-Decoupled means using decoupled head.

We can get the following conclusions from Table 5. Compared with the original YOLOv7, the *P*, *R*, and *mAP* of YOLOv7-RepGhost improved by 1.2%, 0.1%, and 0.5%. The *P* and *mAP* of YOLOv7-Decoupled improved by 2.7% and 0.5%. The *P*, *R*, and *mAP* of YOLOv7-GhostSPPCSPC improved by 0.1%, 0.8%, and 0.3%. The introduction of these three methods has significantly improved the detection performance of the network. On the contrary, the introduction of other methods has no positive effects or little positive effects on the network, and we will no further use these methods.

3.2. Ablation Experiments

Through the experiment in Section 3.1, we found that introducing one of RepGhost, decoupled head, and GhostSPPCSPC can improve the detection performance of the model. Therefore, we conduct ablation experiments to explore whether the fusion of these methods can further enhance the performance of the YOLOv7 network. Taking the original YOLOv7 as the baseline, we introduced RepGhost, Decoupled, and GhostSPPCSPC into the network in turn. The experiment results are shown in Table 6.

Table 6. Ablation experiment of RepGhost, decoupled head, and GhostSPPCSPC.

RepGhost	Decoupled Head	GhostSPPCSPC	*P*(%)	*R*(%)	*mAP*(%)
×	×	×	85.7	88.7	91.8
√	×	×	86.9	88.8	92.3
√	√	×	89.8	90.6	92.8
√	√	√	86.1	91.2	92.6

The experiment results in Table 6 show that the introduction of RepGhost improves the *P*, *R*, and *mAP* by 1.2%, 0.1%, and 0.5%. The introduction of RepGhost and decoupled head improves the *P*, *R*, and *mAP* by 4.1%, 1.9%, and 0.8%. When GhostSPPCSPC is further introduced, the detection performance of the model declines. Therefore, we name the model which introduced RepGhost and decoupled head as RD-YOLOv7 and try to introduce attention mechanisms into the RD-YOLOv7 to further improve detection performance.

3.3. Experiments of Different Attention Mechanisms in Dragon Fruit Detection

Using RepGhost and decoupled head can improve the detection accuracy of the YOLOv7 model. However, how to extract and fuse features more accurately is still a potential problem. Thus, we try to add different attention mechanisms at different positions in RD-YOLOv7 to increase the weights of effective features. Convolutional block attention module (CBAM), coordinate attention (CA) and ECA are added to the RD-YOLOv7 model. The attention mechanism is a plug-and-play module. Its position in the network will also affect the detection performance. Therefore, we add these attention mechanisms to two

positions in the network which are after each RepGhost in the backbone network and after each ELAN-W structure in the head network. The experiment results are shown in Table 7.

Table 7. Experiment results of different attention mechanisms in different positions.

Base Model	Plug Position	Attention Mechanism	*P*(%)	*R*(%)	*mAP*(%)
RD-YOLOv7	RepGhost	CBAM	85.2	89.6	91.8
		CA	87.6	89.0	92.4
		ECA	90.1	88.9	92.7
	ELAN-W	CBAM	86.1	90.3	92.5
		CA	83.6	88.5	91.7
		ECA	88.9	91.2	93.0
	Both	CBAM	82.5	91.7	92.3
		CA	84.7	86.2	92.0
		ECA	90.7	90.8	93.4

As shown in Table 7, the introduction of CBAM and CA in RD-YOLOv7 will reduce the detection performance. After introducing ECA into the backbone network, the *P* of RD-YOLOv7 is increased by 0.3%. After introducing ECA into the head network, the *R* and *mAP* of RD-YOLOv7 are increased by 0.6% and 0.2%. When ECA is introduced in both the two positions at the same time, the performance of the model is significantly improved, where *P* is improved by 0.9%, *R* is improved by 0.2%, and *mAP* is improved by 0.6. The results mean that the ECA is the best attention mechanism in RD-YOLOv7, and the model can learn the parameter in the ECA by itself. In the training stages, the ECA block in different positions may play a different role in the model. By introducing ECA into both the backbone network and head network, the final model named RDE-YOLOv7 is proposed. The RDE-YOLOv7 improves the *P*, *R*, and *mAP* by 5.0%, 2.1%, and 1.6%.

3.4. Experiments under Different Light Conditions

In the natural environment, the drastic change of light is an important factor that affects the detection accuracy of the model. To verify that the RDE-YOLOv7 model has high detection accuracy under different light conditions. Use the images taken under different lighting conditions in Table 1 for experiments. The experiment results are shown in Table 7.

The results in Table 8 show that the RDE-YOLOv7 proposed in this study has high detection accuracy under different lighting conditions. In addition, the proposed model also has high detection accuracy in detecting the two categories of dragon fruit. The experimental results show that the model is robust enough to detect dragon fruit under different light conditions, which makes it possible for the picking robot to pick all day long. Figure 8 shows some visualization results of detecting dragon fruit under different light conditions by using the RDE-YOLOv7 model.

Table 8. The detection results of RDE-YOLOv7 under different light conditions.

Light Condition	Class	*P*(%)	*R*(%)
Strong light	DF_F	91.7	92.1
	DF_S	90.6	90.1
Weak light	DF_F	92.1	92.5
	DF_S	91.0	91.6
Artificial light	DF_F	89.5	89.2
	DF_S	89.3	89.3

Figure 8. The visualization results of detecting dragon fruit under different light conditions by using the RDE-YOLOv7 model.

3.5. Experiments under Different Blur Conditions

When performing the picking task in the orchard, the image captured by the camera may be blurred when the robot is moving. Therefore, to verify the fault tolerance of RDE-YOLOv7 on blurred images, we randomly select 100 images from the test set and use Gaussian blur to process these images, the kernel size of Gaussian blur is set to 51, 101, 151 for generating images with different blur degrees. The larger the Gaussian kernel, the more blurred the generated image. Then, 300 blurred images are used to test. Table 9 shows the experiment results of detecting the 300 blurred images.

Table 9. The detection results of RDE-YOLOv7 under different blur conditions.

Gaussian Kernel	*P*(%)	*R*(%)	*mAP*(%)
0	91.8	91.1	93.3
51	89.8	90.7	92.7
101	89.4	76.2	86.1
151	75.1	59.3	62.4

As shown in Table 9, when the Gaussian kernel with the size of 51 is used for blur processing, the detection precision, recall, and *mAP* of the RDE-YOLOv7 model are decreased by 2.0%, 0.3%, and 0.6% respectively. when the Gaussian kernel with the size of 101 is used for blur processing, the detection precision, recall, and *mAP* of the RDE-YOLOv7 model are decreased by 2.4%, 14.9%, and 7.2% respectively. when the Gaussian kernel with the size of 151 is used for blur processing, the detection precision, recall, and *mAP* of the RDE-YOLOv7 model are decreased by 16.7%, 31.8%, and 30.9% respectively. The experiment results show that the image with a low blurring degree has little impact on the RDE-YOLOv7 model and can be ignored, the image with a medium blurring degree has a great impact on the recall of the RDE-YOLOv7model, which lead to missing detection of some fruits. When the blurring degree of the image is high, the detection precision and recall of the RDE-YOLOv7 model will decrease significantly. At this time, there may be many fruits that are misidentified and missed. Figure 9 shows some detection results of images processed with different Gaussian kernels.

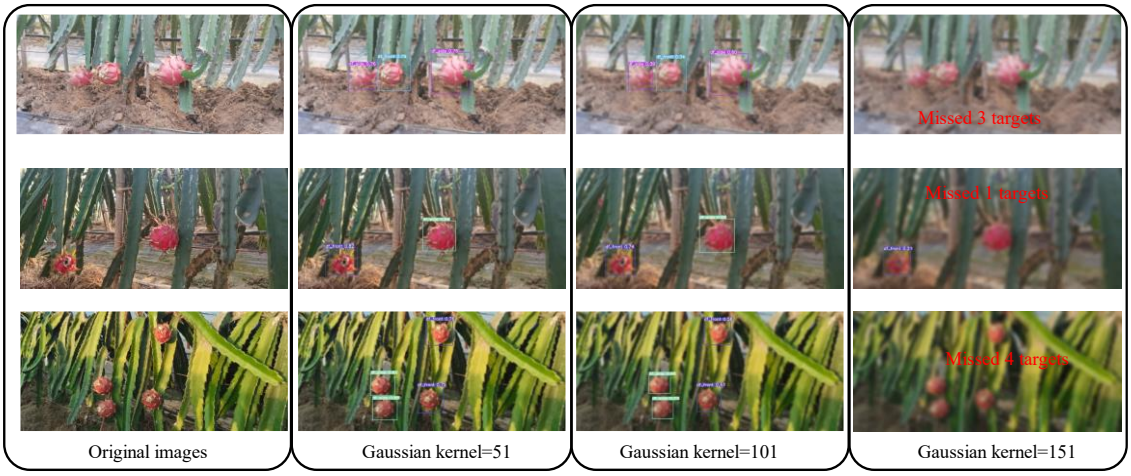

Figure 9. The detection results of images with different Gaussian kernels.

3.6. Experiments for the Best Confidence Threshold in Real Detection

For a picking robot, the error detection rate should be greatly reduced, because the error detection will make the manipulator drive the end-effector to some error positions, which may damage the fruit or even the manipulator. It is better to miss-picking than to error-picking. So, *P* is more important than *R*. In real detection, we need to set the confidence threshold and intersection over the union (IoU) threshold. The IoU threshold is used for non-maximum suppression (NMS), and we set it to 0.5. The confidence threshold is used to determine whether there is an object in the prediction box. When the confidence of a prediction box is bigger than the set threshold, it is considered that there is an object in the box. If the confidence threshold is set too small, some negative samples may be considered dragon fruits, which leads to error picking. If the confidence threshold is set too big, some dragon fruits will be considered negative samples, which leads to miss-picking. In the above experiments, the importance of *P* and *R* was not considered when calculating the metrics. The principle we follow is to ensure a certain *R* and try to improve the *P*. To select the best confidence threshold in real detection, we set different confidence thresholds, use the RDE-YOLOv7 to detect the images in the test set, and calculate the *P* and *R*. The test results are shown in Table 10.

Table 10. The experiment results of RDE-YOLOv7 with different confidence thresholds.

Confidence Threshold	*P*(%)	Increase(%)	*R*(%)	Decrease(%)
0.1	76.2	-	95.8	-
0.2	80.7	4.5	93.6	2.2
0.3	83.6	2.9	92.4	1.2
0.4	85.6	2.0	91.7	0.7
0.5	88.7	3.1	91.2	0.5
0.6	91.3	2.6	89.8	1.4
0.7	91.8	0.5	86.7	3.1
0.8	92.4	0.6	76.4	10.3
0.9	94.5	2.1	53.4	23.0

The results in Table 10 show that the greater the confidence setting, the higher the accuracy and the lower the recall rate. When the set confidence threshold is below 0.6, the *p* value increases rapidly and the *R*-value decreases slowly, which is because some negative samples and dragon fruit with low detection probability are gradually suppressed. When the set confidence threshold is bigger than 0.6, the *p* value increases slowly and the *R*-value decreases rapidly, which is because a small number of dragon fruits with incorrect classification and most dragon fruits with correct detection are gradually suppressed. Finally, we set the confidence threshold to 0.6 in real detection.

3.7. Experiments of Spatial Positioning

For a picking robot, accurate target positioning is necessary. To verify the high positioning accuracy of RDE-YOLOv7 proposed in this paper, we have conducted positioning verification Experiments. The equipment used in the experiment includes a manipulator, stereo camera, and edge computer. Their product models are S6H4D_Plus, Zed Mini, and Jetson AGX Orin. After camera calibration and hand-eye calibration, the experimental steps are as follows: (1) Detect the image collected by the camera to obtain the two-dimensional coordinates of the image; (2) calculate the point cloud chart and conduct coordinate transformation to obtain the three-dimensional coordinates of the target in the manipulator coordinate system; (3) send a motion command to the manipulator to make it move to the calculated coordinate position; (4) record the coordinates of the manipulator on the teaching pendant and the calculated coordinates. The coordinates displayed by the teaching pendant are real coordinates, and the calculated coordinates are positioning coordinates. Repeat the experiment several times according to these steps. Tables 11 and 12 show the spatial positioning experiment results of RDE-YOLOv7 and YOLOv7.

Table 11. The spatial positioning experiment results of RDE-YOLOv7.

No	Real Coordinates (mm)			Positioning Coordinates(mm)			Absolute Error (mm)		
	X_r	Y_r	Z_r	X_p	Y_p	Z_p	X_e	Y_e	Z_e
1	41.54	156.92	382.14	42.36	155.18	383.06	0.82	1.74	0.92
2	52.68	245.64	351.48	54.78	246.62	350.40	2.12	0.98	1.08
3	482.36	287.48	375.36	478.89	289.00	374.05	3.47	1.52	1.31
4	471.62	78.52	368.71	473.54	81.14	367.87	1.92	2.62	0.84
5	465.12	396.47	385.34	467.93	399.31	384.59	2.81	2.84	0.75
6	62.34	194.83	378.43	64.98	192.89	376.49	2.64	1.94	1.94
7	457.82	167.49	357.69	460.29	165.14	355.22	2.47	2.35	2.47
8	468.97	241.39	348.15	472.24	237.98	345.29	3.27	3.41	2.86
9	68.26	96.18	362.62	69.90	95.34	363.16	1.64	0.84	0.54
10	57.69	128.37	378.94	59.42	126.61	377.27	1.73	1.76	1.67
11	52.48	329.67	341.26	55.76	326.78	343.09	3.28	2.89	1.83
12	47.62	156.98	352.76	50.56	153.31	353.70	2.94	3.67	0.94
13	437.68	357.81	349.28	440.44	353.49	352.55	2.76	4.32	3.27
14	462.84	285.74	368.81	459.77	282.99	372.30	3.07	2.75	3.49
15	46.98	86.45	384.74	48.55	84.11	387.22	1.57	2.34	2.48
16	452.69	187.43	357.69	451.35	189.40	355.31	1.34	1.97	2.38
17	467.59	245.34	379.38	471.26	243.48	381.22	3.67	1.86	1.84
18	487.32	387.41	364.59	484.03	384.15	363.83	3.29	3.26	0.76
19	67.76	156.74	357.95	70.38	154.15	359.64	2.62	2.59	1.69
20	59.71	109.48	375.64	62.42	106.51	379.39	2.71	2.97	3.75

The X in the header represents the left and right coordinates in the camera view, Y represents the up and down coordinates in the camera view, and Z represents the depth coordinates.

The spatial positioning experiment results of RDE-YOLOv7 in Table 11 show that the maximum absolute errors of XYZ in three directions are 3.67 mm, 4.32 mm, and 3.75 mm respectively, and the minimum values are 0.82 mm, 0.84 mm, and 0.54 mm respectively, and the average values are 2.51 mm, 2.43 mm, and 1.84 mm respectively. The spatial positioning experiment results of YOLOv7 in Table 12 show that the maximum absolute errors of XYZ in three directions are 3.82 mm, 4.24 mm, and 3.54 mm respectively, the minimum values are 1.34 mm, 1.65 mm, and 0.87 mm, and the average values are 2.60 mm, 2.81 mm, and 2.06 mm respectively. The average positioning error of RDE-YOLOv7 is smaller than that of YOLOv7, which to some extent indicates that RDE-YOLOv7 has higher positioning accuracy. In addition, the small error in X and Y directions indicates that the RDE-YOLOv7 model is accurate in positioning. The spatial positioning experiment results show that the error value is relatively stable and there is no abnormal situation, and the error value is within 5 mm, which is tolerable by the picking system.

Table 12. The spatial positioning experiment results of YOLOv7.

No	Real Coordinates (mm)			Positioning Coordinates (mm)			Absolute Error (mm)		
	X_r	Y_r	Z_r	X_p	Y_p	Z_p	X_e	Y_e	Z_e
1	52.71	241.32	344.72	54.24	243.81	342.50	1.53	2.49	2.22
2	58.49	167.93	381.23	56.08	165.95	382.94	2.41	1.98	1.71
3	471.71	247.75	358.12	474.73	251.09	356.84	3.02	3.34	1.28
4	62.92	104.65	372.84	64.84	106.30	370.70	1.92	1.65	2.14
5	458.36	284.32	353.31	462.18	280.73	354.23	3.82	3.59	0.92
6	459.24	312.57	343.67	456.07	315.41	345.10	3.17	2.84	1.43
7	65.31	194.78	353.89	62.64	191.86	351.55	2.67	2.92	2.34
8	461.34	258.88	382.39	463.83	261.36	379.28	2.49	2.48	3.11
9	454.87	348.36	368.78	452.64	345.70	367.12	2.23	2.66	1.66
10	473.87	276.94	374.05	475.78	275.15	375.17	1.91	1.79	1.12
11	61.39	127.71	359.67	60.05	131.52	361.51	1.34	3.81	1.84
12	56.73	211.82	363.82	59.76	209.72	360.95	3.03	2.10	2.87
13	449.21	174.64	351.29	451.97	171.23	354.83	2.76	3.41	3.54
14	67.82	119.54	383.91	71.56	122.73	382.26	3.74	3.19	1.65
15	59.72	109.87	377.18	62.65	107.39	374.56	2.93	2.48	2.62
16	69.18	328.44	372.72	66.50	326.52	371.85	2.68	1.92	0.87
17	468.82	374.17	349.07	470.40	377.96	350.99	1.58	3.79	1.92
18	485.18	229.40	357.77	482.54	225.16	361.20	2.64	4.24	3.43
19	71.23	166.93	371.84	67.99	170.15	369.30	3.24	3.22	2.54
20	56.98	248.89	368.38	59.84	246.51	366.46	2.86	2.38	1.92

The X in the header represents the left and right coordinates in the camera view, Y represents the up and down coordinates in the camera view, and Z represents the depth coordinates.

3.8. Dragon Fruits Picking Experiments

To directly prove the effectiveness of the RDE-YOLOv7 model proposed in this paper, we have carried out some simulated picking experiments in the laboratory. The picking system is mainly composed manipulator, Zed Mini, Jetson AGX Orin, control box, and end-effector. Figure 10 shows the picking system.

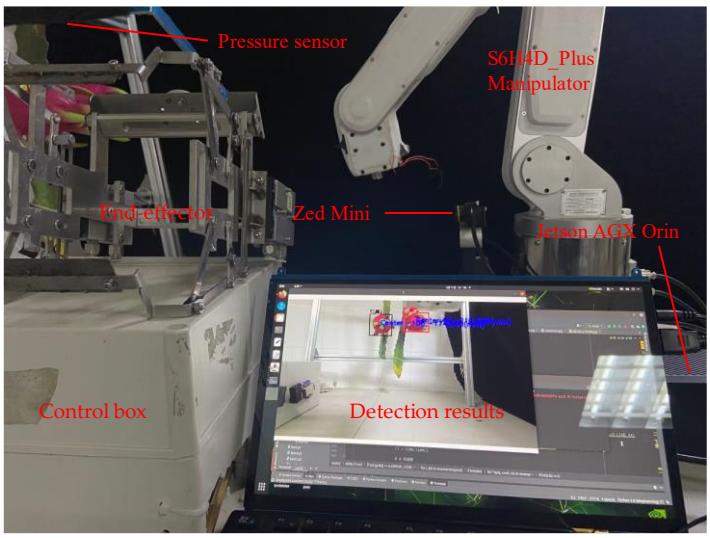

Figure 10. The dragon fruit picking system.

The manipulator is controlled to drive the end-effector to the target position. Zed Mini is a stereo camera, and it is used for image acquisition and 3D positioning. Jetson AGX Orin is used to inference the image and compute the point cloud chart. The control box is used to control some controlling elements and make the end-effector cut the branch. In this experiment, because DF_S cannot be picked directly according to the coordinates, only DF_F with successful recognition is picked.

The experiment steps are as follows: (1) start the program; (2) detect and spatially locate the DF_F; (3) send motion command to the manipulator; (4) pressure sensor contacts branch to trigger cut action. Place the dragon fruit at different positions in the camera view, and repeat the experiment many times. Figure 11 shows the picking process.

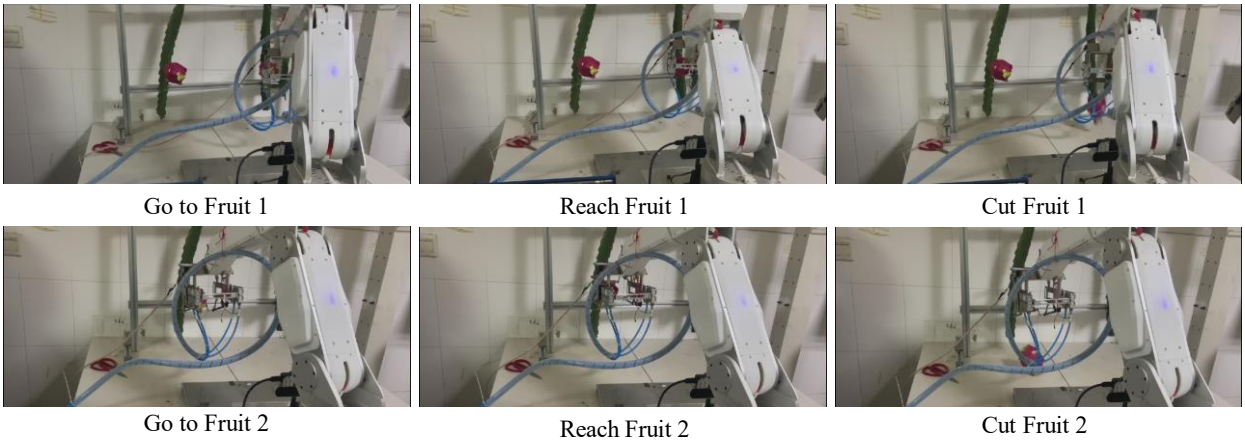

Figure 11. The process of picking dragon fruits.

In picking experiments, the average time required to infer an image using RDE-YOLOV7 is 27.1 ms, which meets the real-time requirements. In addition, it takes about 15 s for the manipulator to pick a dragon fruit and back to the original position.

4. Conclusions

This study proposes a dragon fruit detection method named RDE-YOLOv7, which can more accurately detect dragon fruits in complex scenes. RepGhost and decoupled head are introduced into YOLOv7 to replace the ELAN and coupled head for better detection performance, and they are proven to improve *P*, *R*, and *mAP* by 4.1%, 1.9%, and 1%. ECA is proven to be the best attention mechanism among CBAM, CA, and ECA. It is added into the backbone network and head network to pay more attention to the targets which further improve *P*, *R*, and *mAP* by 0.9%, 0.2%, and 0.6%. RDE-YOLOv7 is constructed by introducing RepGhost, decoupled head, and ECA block into YOLOv7.

The following results are obtained by analyzing the experimental results. Compared with the original YOLOv7, RDE-YOLOv7 improves *P*, *R*, and *mAP* by 5.0%, 2.1%, and 1.6%. RDE-YOLOv7 also has good performance in detecting dragon fruits under different light conditions, which is of great significance to the outdoor all-day work of the picking robot. In addition, RDE-YOLOv7 can detect dragon fruits in slightly blurred images.

A dragon fruit-picking system is constructed in this paper. The results of Spatial positioning experiments show that the positioning accuracy of the vision system formed by RDE-YOLOv7 is very high. The coordinate positioning errors in the space of the vision system are only 2.51 mm, 2.43 mm, and 1.84 mm, which can lead the manipulator to drive the end-effector to reach the accurate position. The picking experiments have proved that the RDE-YOLOv7 proposed in this paper can be used to pick dragon fruit.

A high-precision object detection model can significantly improve fruit-picking robots' picking efficiency and accuracy. In future research, the fruit will not only be detected from the two-dimensional image, but also the point cloud image calculated by the stereo camera may be able to identify and locate the fruit more accurately.

Author Contributions: Conceptualization, Y.Z.; data curation, J.Z. and Y.Z.; formal analysis, Y.Z.; funding acquisition, J.W.; methodology, J.Z.; project administration, J.W.; software, J.Z.; validation, J.Z. and Y.Z.; visualization, J.Z.; writing—original draft, J.Z.; writing—review & editing, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Jiangsu Province Agricultural Science and Technology Independent Innovation Project (CX(22)3099), Key R&D Program of Jiangsu Modern Agricultural Machinery Equipment and Technology Promotion Project (NJ2021-18) and the Key R & D plan of Jiangsu Province (BE2021016-2).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Acknowledgments: In addition, we would like to thank Lile Agriculture Science and Technology Ltd., Nanjing City, Jiangsu Province, for the dragon fruit data provided and Zhou Lei for proofreading of the draft.

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

References

1. Reddy Maddikunta, P.K.; Hakak, S.; Alazab, M.; Bhattacharya, S.; Gadekallu, T.R.; Khan, W.Z.; Pham, Q.-V. Unmanned Aerial Vehicles in Smart Agriculture: Applications, Requirements, and Challenges. *IEEE Sens. J.* **2021**, *21*, 17608–17619. [CrossRef]
2. Wang, Z.; Xun, Y.; Wang, Y.; Yang, Q. Review of Smart Robots for Fruit and Vegetable Picking in Agriculture. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 33–54. [CrossRef]
3. Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. A Review of Key Techniques of Vision-Based Control for Harvesting Robot. *Comput. Electron. Agric.* **2016**, *127*, 311–323. [CrossRef]
4. Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic Method of Fruit Object Extraction under Complex Agricultural Background for Vision System of Fruit Picking Robot. *Optik* **2014**, *125*, 5684–5689. [CrossRef]
5. Zhou, H.; Wang, X.; Au, W.; Kang, H.; Chen, C. Intelligent Robots for Fruit Harvesting: Recent Developments and Future Challenges. *Precision Agric.* **2022**, *23*, 1856–1907. [CrossRef]
6. 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]
7. Ukwuoma, C.C.; Zhiguang, Q.; Bin Heyat, M.B.; Ali, L.; Almaspoor, Z.; Monday, H.N. Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques. *Math. Probl. Eng.* **2022**, *2022*, 9210947. [CrossRef]
8. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep Learning—Method Overview and Review of Use for Fruit Detection and Yield Estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234. [CrossRef]
9. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection during Different Growth Stages in Orchards Using the Improved YOLO-V3 Model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
10. Zhang, X.; Huo, L.; Liu, Y.; Zhuang, Z.; Yang, Y.; Gou, B. Research on 3D Phenotypic Reconstruction and Micro-Defect Detection of Green Plum Based on Multi-View Images. *Forests* **2023**, *14*, 218. [CrossRef]
11. Ding, F.; Zhuang, Z.; Liu, Y.; Jiang, D.; Yan, X.; Wang, Z. Detecting Defects on Solid Wood Panels Based on an Improved SSD Algorithm. *Sensors* **2020**, *20*, 5315. [CrossRef] [PubMed]
12. Wang, J.; Li, Z.; Chen, Q.; Ding, K.; Zhu, T.; Ni, C. Detection and Classification of Defective Hard Candies Based on Image Processing and Convolutional Neural Networks. *Electronics* **2022**, *10*, 2017. [CrossRef]
13. 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]
14. Liu, X.; Zhao, D.; Jia, W.; Ji, W.; Ruan, C.; Sun, Y. Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation. *IEEE Access* **2019**, *7*, 139635–139642. [CrossRef]
15. Hu, C.; Shi, Z.; Wei, H.; Hu, X.; Xie, Y.; Li, P. Automatic Detection of Pecan Fruits Based on Faster RCNN with FPN in Orchard. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 189–196. [CrossRef]
16. Wan, S.; Goudos, S. Faster R-CNN for Multi-Class Fruit Detection Using a Robotic Vision System. *Comput. Netw.* **2020**, *168*, 107036. [CrossRef]
17. Gai, R.; Chen, N.; Yuan, H. A Detection Algorithm for Cherry Fruits Based on the Improved YOLO-v4 Model. *Neural Comput. Applic* **2021**. [CrossRef]
18. Xu, Z.; Huang, X.; Huang, Y.; Sun, H.; Wan, F. A Real-Time Zanthoxylum Target Detection Method for an Intelligent Picking Robot under a Complex Background, Based on an Improved YOLOv5s Architecture. *Sensors* **2022**, *22*, 682. [CrossRef]
19. Xue, J.; Cheng, F.; Li, Y.; Song, Y.; Mao, T. Detection of Farmland Obstacles Based on an Improved YOLOv5s Algorithm by Using CIoU and Anchor Box Scale Clustering. *Sensors* **2022**, *22*, 1790. [CrossRef]
20. Kang, H.; Wang, X.; Chen, C. Geometry-Aware Fruit Grasping Estimation for Robotic Harvesting in Apple Orchards. *Comput. Electron. Agric.* **2022**, *193*, 106716. [CrossRef]
21. 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]

22. Wang, J.; Gao, K.; Jiang, H.; Zhou, H. Method for Detecting Dragon Fruit Based on Improved Lightweight Convolutional Neural Network. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 218–225. [CrossRef]
23. Zhang, B.; Wang, R.; Zhang, H.; Yin, C.; Xia, Y.; Fu, M.; Fu, W. Dragon Fruit Detection in Natural Orchard Environment by Integrating Lightweight Network and Attention Mechanism. *Front. Plant Sci.* **2022**, *13*, 1040923. [CrossRef]
24. Wu, D.; Jiang, S.; Zhao, E.; Liu, Y.; Zhu, H.; Wang, W.; Wang, R. Detection of Camellia Oleifera Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation. *Appl. Sci.* **2022**, *12*, 11318. [CrossRef]
25. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
26. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2020**, arXiv:1910.03151.
27. Chen, C.; Guo, Z.; Zeng, H.; Xiong, P.; Dong, J. RepGhost: A Hardware-Efficient Ghost Module via Re-Parameterization. *arXiv* **2022**, arXiv:2211.06088.
28. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.