remote sensing

MDPI

Article

An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds

Chenxi Zhang, Feng Kang and Yaxiong Wang *

Key Lab of State Forestry and Grassland Administration on Forestry Equipment and Automation, School of Technology, Beijing Forestry University, Beijing 100083, China
* Correspondence: yaxiongwang87@bjfu.edu.cn

Abstract: Convolutional neural networks have recently experienced successful development in the field of computer vision. In precision agriculture, apple picking robots use computer vision methods to detect apples in orchards. However, existing object detection algorithms often face problems such as leaf shading, complex illumination environments, and small, dense recognition targets, resulting in low apple detection rates and inaccurate localization. In view of these problems, we designed an apple detection model based on lightweight YOLOv4—called Improved YOLOv4—from the perspective of industrial application. First, to improve the detection accuracy while reducing the amount of computation, the GhostNet feature extraction network with a Coordinate Attention module is implemented in YOLOv4, and depth-wise separable convolution is introduced to reconstruct the neck and YOLO head structures. Then, a Coordinate Attention module is added to the feature pyramid network (FPN) structure in order to enhance the feature extraction ability for medium and small targets. In the last 15% of epochs in training, the mosaic data augmentation strategy is turned off in order to further improve the detection performance. Finally, a long-range target screening strategy is proposed for standardized dense planting apple orchards with dwarf rootstock, removing apples in non-target rows and improving detection performance and recognition speed. On the constructed apple data set, compared with YOLOv4, the mAP of Improved YOLOv4 was increased by 3.45% (to 95.72%). The weight size of Improved YOLOv4 is only 37.9 MB, 15.53% of that of YOLOv4, and the detection speed is improved by 5.7 FPS. Two detection methods of similar size—YOLOX-s and EfficientNetB0-YOLOv3—were compared with Improved YOLOv4. Improved YOLOv4 outperformed these two algorithms by 1.82% and 2.33% mAP, respectively, on the total test set and performed optimally under all illumination conditions. The presented results indicate that Improved YOLOv4 has excellent detection accuracy and good robustness, and the proposed long-range target screening strategy has an important reference value for solving the problem of accurate and rapid identification of various fruits in standard orchards.

Keywords: precision agriculture; YOLOv4; attention mechanism

Citation: Zhang, C.; Kang, F.; Wang, Y. An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds. *Remote Sens.* **2022**, *14*, 4150. https://doi.org/10.3390/rs14174150

Academic Editors: Carlos Antonio Da Silva Junior and Luciano Shozo Shiratsuchi

Received: 12 July 2022
Accepted: 20 August 2022
Published: 24 August 2022

1. Introduction

The fresh fruit industry is labor-intensive, with apple-picking labor costs accounting for over 50% of total costs [1,2]. As the aging population expands, the working population of China is shrinking, and labor costs are rising. To address the problems of insufficient labor, high manual picking cost, and low efficiency [3], it is necessary to further study the key technology of apple picking robots.

In the past few decades, with the continuous progress of precision agriculture technology, fruit picking robots have been extensively developed in the field of agriculture. However, few efficient apple harvesting robots have been used in actual harvesting scenarios so far [4].

Although there are many object-detection models with excellent performance that can detect a wide range of different kinds of targets, some specific challenges still exist in the

field of agriculture. In particular, leaf shading and complex illumination environment are common in orchards, which requires object-detection algorithms with a stronger ability to identify targets in complex shading environments. In addition, the targets detected in agriculture are generally small and dense, which is challenging for some object-detection algorithms. Furthermore, the detection system must be lightweight, such that it can be implemented on the execution terminal for real-time detection. Therefore, it is of great research value and relevance for achieving automatic harvesting and yield estimation to study a vision system that is capable of accurately identifying fruit on trees in complex environments.

In this study, a new method for fruit detection is proposed. Our overall contributions can be summarized as follows:

- Based on YOLOv4, a lightweight GhostNet is used as the feature extraction network. Depth-wise convolutions are introduced in the neck and YOLO head structures. In this way, the number of parameters in the model can be significantly reduced.
- Coordinate attention modules are added in the backbone, as well as before the 52×52 and 26×26 scale inputs of the FPN in the neck structure. The apple target detection capability of the model is enhanced, especially for targets with small- and medium-sized pixels.
- A screening strategy for long-range targets is proposed. The distant apples are screened out as not part of the harvestable target in the non-maximum suppression (NMS) stage. Thus, the detection efficiency is improved.

The remainder of the paper is structured as follows: Section 2 presents related work in the field of fruit detection. Section 3 describes the methods used for collecting, expanding, and partitioning data sets, introduces GhostNet and attention mechanisms, and details the proposed improvements to YOLOv4. Section 4 presents the experimental configuration and results. Section 5 discusses the strengths and weaknesses of the algorithm proposed in this study. Finally, our conclusions are elaborated.

2. Related Work

2.1. Fruit Detection

In the early days, traditional methods were commonly used for image detection and segmentation. Lin et al. [5] have proposed a support vector machine-based citrus recognition algorithm, which can attenuate the effect of illumination changes based on the recognition effect; however, the shortcomings include that it takes 1.25 s to detect one image, such that the real-time performance is poor. Fan et al. [6] have combined local image features and color information to propose a pixel block segmentation method based on gray-scale centered RGB color space, which can effectively distinguish apple pixels from other pixels.

Since 2012, deep learning strategies have been widely used in the field of computer vision (CV) object detection due to their robustness. AlexNet [7], published by Alex et al. in 2012, has been used to apply convolutional neural networks (CNN) in the field of CV and has been used to detect objects, including various fruits, in images. ZF-net [8], published in 2014, explained the role of each layer of the CNN based on AlexNet. In 2015, VGG16 [9] was proposed as an improved version of AlexNet, in which the number of network layers was deepened. A kiwi fruit-picking robot using VGG16 achieved a recognition accuracy of 89.6% and a real picking rate of 51.0% in a field experiment [10]; however, it is demanding in terms of picking environment and, therefore, is not suitable for recognition in complex and sheltered environments.

With the continuous development and advancement of end-to-end object-detection algorithms, their great superiority in fruit and vegetable object detection has been highlighted. There are two main types of such algorithms: the first is two-stage object detection methods, represented by RCNN [11], Fast RCNN [12], Faster RCNN [13], and so on. The idea of these algorithms is to first obtain target-area suggestion boxes and then use the features of the regions to predict their classes and bounding boxes. Sun et al. [14] have proposed an improved Faster RCNN-based tomato detection method, which used ResNet50

as a feature extraction network and k-means clustering to adjust the bounding boxes, effectively improving the recognition accuracy; however, the detection speed was slow. Zhang et al. [15] have detected 86% of apple tree branches with R-CNN. Bargoti et al. [16] have provided an open-source fruit data set on which they trained their model based on Faster RCNN, ultimately achieving an F1 score of over 0.9. Grilli et al. [17] have studied an apple photogrammetry system. In their paper, two apple segmentation methods, k-Means and Mask RCNN, were compared, where Mask RCNN was found to be more accurate (with 95.18% precision).

The other type of method is the one-stage object-detection method, represented by SSD [18], the YOLO series [19–23], and so on. This method divides the image into regions and immediately determines the range of objects and the probability of classification for each object. Tian et al. [24] have put forward an improved YOLOv3 algorithm, using DenseNet as the feature extraction network. The low-resolution feature layer was enhanced to recognize apples in different growth stages. Lu et al. [25] have added a CBAM attention mechanism to the YOLOv4 detector in order to improve the detection accuracy by focusing only on the target canopy. By adding an adaptive layer and a larger scale feature map to the improved network structure, their experiments showed that the model achieved better performance when compared with the original model. Li et al. [26] have proposed a YOLOv4_tiny-based object detection algorithm for green peppers. The authors compared the results of their algorithm with SSD and Faster-RCNN, where the experimental results demonstrated that the AP value of the improved YOLOv4_tiny was 5.26% and 11.16% higher than those of SSD and Faster-RCNN, respectively, and the recognition speed was the fastest among the three. Different from the YOLO series, FCOS is an Anchor-Free one-stage target detector. Liu et al. [27] replaced the FPN in FCOS with a residual feature pyramid network (RFPN), which improves the detection accuracy of green fruits of different sizes. The detection and segmentation accuracy of green fruit reaches 81.2% and 85.3% on the apple dataset. Sun et al. [28] proposed an end-to-end RGB-D object-detection network, termed a noise-tolerant feature fusion network (NTFFN), to utilize the outdoor multi-modal data properly and improve the detection accuracy.

In general, two-stage methods have higher detection accuracy, while one-stage methods have faster inference speed [29]. One-stage object detection algorithms are typically chosen more often for picking robot vision systems in order to balance the accuracy and recognition speed. Compared with the YOLOv3 model, the YOLOv4 model has better accuracy while maintaining the same speed; however, the YOLOv4 model is a massive burden for low-performance devices due to the large size and the computational complexity.

2.2. Attention Mechanisms

In recent years, attention mechanisms have gradually become more and more important in the CV field, as they can redistribute the input weights and, thus, improve model performance. The SE attention mechanism [30] has been widely used, but this attention mechanism contains two fully connected networks, resulting in a cumbersome number of parameters. Wang et al. have proposed the Efficient Channel Attention Network (ECA-Net) [31], including a method to achieve local cross-channel interactions without dimensionality reduction by one-dimensional convolution, which allows for a significant performance improvement effect with a small number of parameters. The spatial structure of the target is important in vision tasks. Coordinate Attention [32], proposed by Hou et al., effectively solves the problem that the SE attention mechanism only considers internal channel information and ignores the importance of location information. The Coordinate Attention module can improve accuracy without increasing the number of parameters. Han et al. [33] have constructed a remote sensing image denoising network based on a deep learning approach, which enhances the ECA-Net by using multiple local jump connections to improve the denoising ability of the model. Kim et al. [34] have reduced the computational effort required to detect small targets and improved the detection rate by using the channel attention pyramid method. To split the overlapping apples from the monochrome

background, Jia et al. [35] transplanted the Gaussian non-local attention mechanism into Mask R-CNN. The AP box and AP mask metric values have reached 85.6% and 86.2% in a reasonable run time, respectively.

In the computer vision field, attention mechanisms are either applied in conjunction with convolutional networks or used to replace certain components of convolutional networks while keeping their overall structure in place [36]. A Vision Transformer is a Transformer structure based on a complete self-attention without using a CNN [37]. The best Top-1 ACC on ImageNet was 88.55%, comparable with SOTA on ImageNet at that time. However, it only performs better after training on a large data set, and the number of parameters in the model is considered too large for edge computing.

2.3. Lightweight Models

The storage and computation of CNN models on mobile and embedded devices remain a considerable challenge due to storage space and computational power limitations. Lightweight networks are a research hotspot in the field of CV, with the aim of reducing the number of model parameters and computational complexity while maintaining high accuracy.

Current research on lightweight models can be divided into two main mainstream directions: Lightweight structure design and model compression. The former direction includes the MobileNet series [38–40], GhostNet [41], ShuffleNet series [42,43], and SqueezeNet [44], among others, whose aim is to design different convolutional methods and structures to make the CNN more lightweight; while the latter compresses the model using knowledge distillation [45], pruning [46], or similar techniques. Additionally, changing the form of convolution provides a good way to reduce the number of model parameters, usually through depth-wise separable convolution [47] and/or dilated convolution [48].

In the field of fruit and vegetable picking, lightweight detection algorithms are essential to reduce the hardware computing power requirements, enabling picking robots to be used in a wider range of environments. Wu et al. [49] have replaced the backbone network, CSPDarknet53, of the YOLOv4 model with EfficientNet. They added convolutional layers to the three outputs in order to reduce the model weight size and the computational complexity. The mean values of Recall, Precision, and F1 were 97.43%, 95.52%, and 96.54%, respectively, and the average detection time of the model was 0.338 s per frame. Zhang et al. [50] have introduced MobileNet v2 into YOLOv4 with an improved attentional feature fusion module. The number of model parameters was compressed to 16.76% of the original model, and the mAP on the PASCAL VOC data set was 81.67%.

Research on lightweight networks has engineering application value; however, there remain problems regarding the recognition of apples in complex environments, such as poor detection ability for dense small targets and differences in recognition effects under different illumination. The detection performance of the relevant models needs to be further enhanced. Therefore, in this study, in order to detect targets better and faster, we combined GhostNet with depth-wise separable convolution to construct a lightweight YOLOv4-based object-detection algorithm.

3. Materials and Methods

3.1. Introduction to the Data Set

3.1.1. Image Acquisition

The apple images used in this study were partially collected in October 2020 at the orchard base of Yaozhuang Village, Qixia City, Yantai City, Shandong Province. The apple variety was "Yanxia 3". The resolution of the camera (Raspberry Pi Camera V2) was 5 MP. A total of 1800 images were collected at three angles: forward light, side light, and back light. The operation of the picking robot was simulated during photography, and the images captured contained different colors, postures, sizes, light, backgrounds, and overlapping and obscured fruit. There were too few images containing less than three apples in the obtained data set. To avoid the effect of the number of apples on the results, eighteen

images of this type were sourced from the Fruits 360 data set [51]. Figure 1 shows a portion of the collected images.

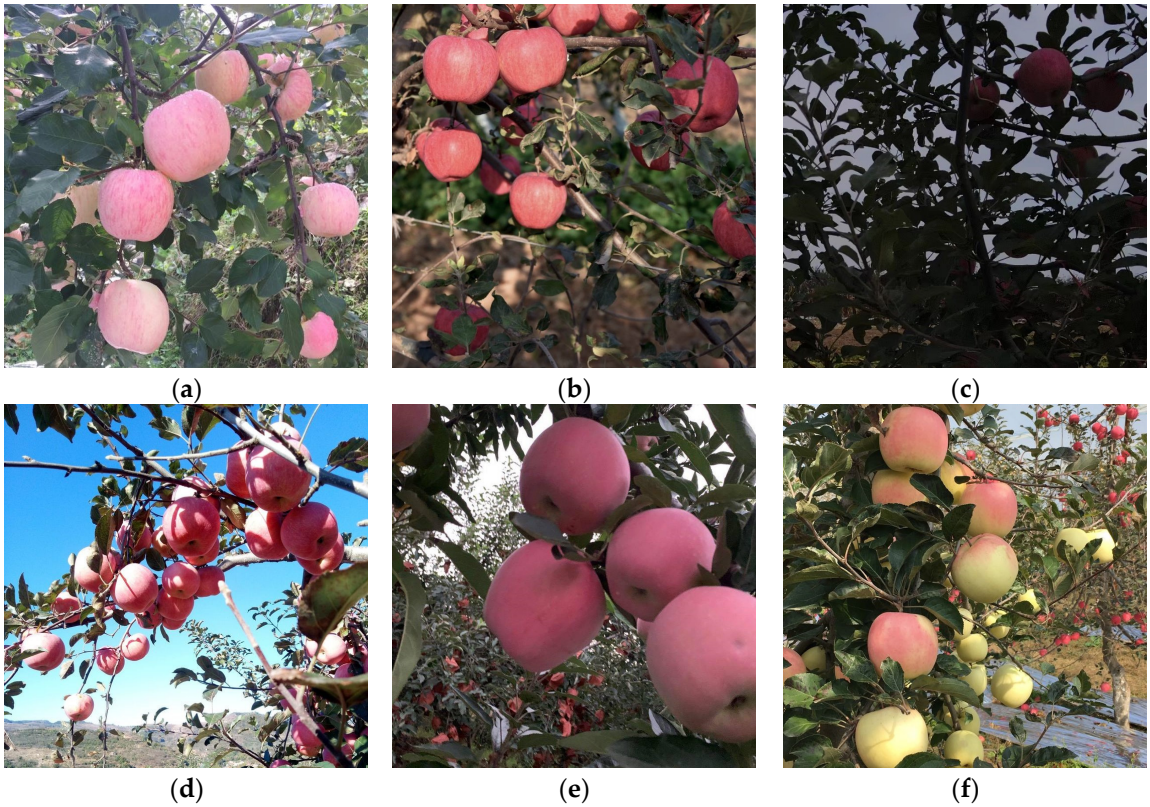


Figure 1. Images of apples in complex scenes: (**a**) Front light; (**b**) side light; (**c**) back light; (**d**) shaded; (**e**) unshaded; and (**f**) unripe.

3.1.2. Data Set Production

In order to improve the training efficiency of the apple target recognition model, the original 1800 images were compressed. The apples were divided into two categories according to different growth periods, one of which was ripe while the other was unripe (i.e., the fruits are not fully colored within one week after bag removal). Ripe_apple and unripe_apple were set to be labeled in two separate categories. In this study, 60 front light images, 60 side light images, and 60 back light images were randomly selected to construct the test set. The training set contained 1440 images, and the validation set had 180 images. Therefore, the ratio of the training set, validation set, and test set was 8:1:1. The details of the test set division are shown in Table 1, while Figure 2 demonstrates the distribution of apple labeling boxes in the training set at different scales. Small and medium targets in the images accounted for the majority, with small targets accounting for about 45.1% and medium targets accounting for about 54.4%.

Table 1. Details of the test set images.

Test Set	Front Light	Side Light	Back Light	Total
Number of images	60	60	60	180
Number of ripe apples	511	386	393	1290
Number of unripe apples	179	206	149	534

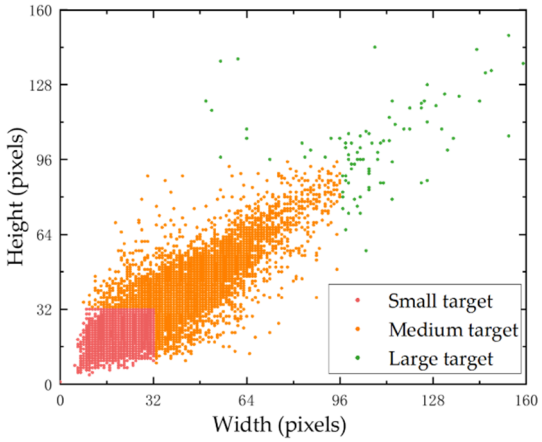


Figure 2. Scale distribution of apples in the training set.

3.1.3. Data Augmentation

In order to better extract apple features of different marker categories and avoid overfitting of the model in training, data augmentation was performed on a total of 1620 apple images in the training and validation sets. We used the getRotationMatrix2D, warpAffine, and flip functions of the OpenCV library to perform random rotation, panning, and mirroring of the original images, respectively. The angle range of the image was set to $\pm 5^{\circ}$ per rotation. The principle of image panning is that all targets should still be included after panning. The probabilities of horizontal flip, vertical flip, and diagonal flip were all 0.33. Each image was judged by a random seed to determine which data augmentation method to invoke, and two new images were randomly generated while the original images were retained, resulting in a total of 4860 images as the final training and validation sets for training the model. There were 3240 augmented images and 1620 original images. The training set, validation set, and test set did not overlap each other. The schematic diagram after enhancement is shown in Figure 3.

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

Figure 3. Augmented training data: (**a**) Original image; (**b**) random rotation + panning; and (**c**) random rotation + mirroring.

3.2. Framework of the Model

The YOLO series are very effective one-stage detection models that offer the advantages of both high real-time performance and fast detection speed [29]. However, the models are computationally complex and consume significant memory space, making them unsuitable for embedded devices and mobile deployments [52]. Therefore, we improved the original model structure of YOLOv4.

The improved network structure can be divided into three parts: Backbone, neck, and YOLO head. Figure 4 shows the improved network structure. In this study, for the purpose of making the model lightweight, after comparison with some other lightweight backbone networks, GhostNet, with better overall performance, was finally selected as the backbone

network to replace the complex CSPDarknet53 backbone network in YOLOv4. To further reduce the number of model parameters, depth-wise separable convolution was introduced to reconstruct the neck and YOLO head structures, while the SE Attention module in the GhostNet network was replaced with a Coordinate Attention module, obtaining a smaller number of parameters and better performance.

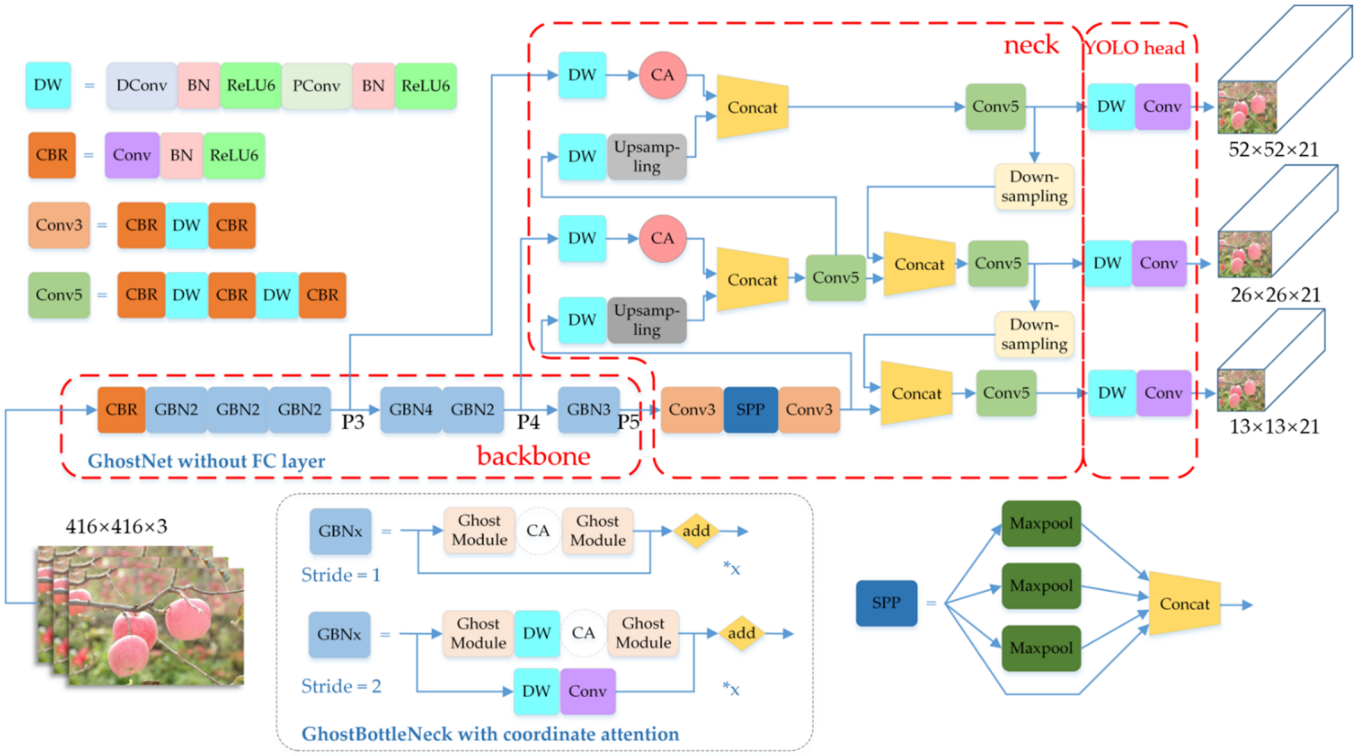


Figure 4. Improved YOLOv4 network. The Ghost Module in the backbone uses the ReLU activation function, while the rest of the network structures use the ReLU6 activation function. CA module means Coordinate Attention. The CA module in GhostBottleNeck is only present in the GBN structures at layers 4, 5, 10, 11, 12, and 14. The symbol "*x" indicates the GBNx is composed of x GBN structures.

From the improved backbone network CA-GhostNet, a total of 3 scales of feature maps—13 × 13 (feature layer P5), 26 × 26 (feature layer P4), and 52 × 52 (feature layer P3)—are output. P5 is suitable for large target detection, P4 is for medium target detection, and P3 is for small-scale target detection. Two Coordinate Attention modules are set before the two inputs of the FPN, which are the inputs of the 52 × 52 feature map and the 26 × 26 map, as well as after the convolution layers. The SPP module was used to enhance the receptive fields, and the three initial feature maps are fused in a feature-wise manner using the FPN structure.

The network features are fused, and 3D tensors at different scales are output through the YOLO head. These 3D tensors are responsible for predicting the target detection boxes at different scales. The YOLO head of each scale of the network outputs a 21-dimensional tensor, calculated as 3 × (4 + 1 + 2), representing the three different scales of anchor boxes at that scale ("3"), the four box position parameters (t_x, t_y, t_w, t_h; "4"), the confidence of the prediction result ("1"), and the number of labels in the data set ("2").

3.3. GhostNet Backbone Network

Deep convolutional neural networks may produce many similar feature maps, which are often treated as redundant information. Han [41] has argued that the powerful extraction function of convolutional networks is positively correlated with these redundant

feature maps and proposed the GhostNet network. Combining a small amount of traditional convolutional computations with light redundant feature generation reduces the number of model parameters while maintaining detection accuracy.

3.3.1. Ghost Convolution and Ghost Bottleneck Module

A comparison of conventional convolution and Ghost convolution is shown in Figure 5. The convolution operation of the Ghost module divides the traditional method into two parts. The first part of the Ghost convolution process uses traditional convolution to generate a small number of intrinsic feature maps, where the convolution operation can be customized with respect to the convolution kernel size. The second part performs linear operations on the feature maps generated in the first part, using light operations and constant transformations to add channels and expand the features. Finally, the feature map of this part and the intrinsic feature map of the first part are stitched together, by constant transformation, as the output feature map of the Ghost module.

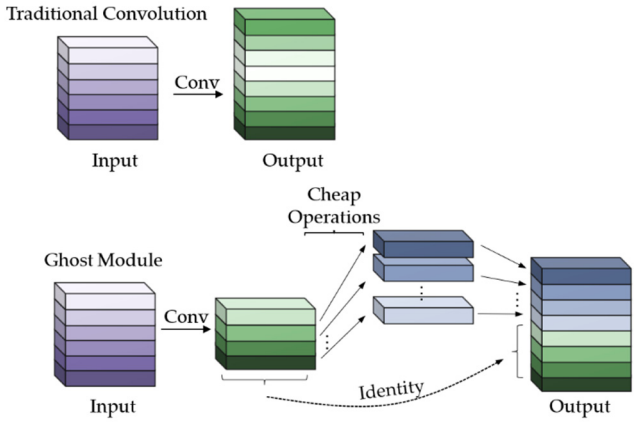


Figure 5. Comparison of Traditional convolution and Ghost convolution.

Assume that the input feature map size is $H \times W$, the number of input channels is C, the output feature map size is $H' \times W'$, the number of output channels is $m \times s$, and the convolution kernel size is $k \times k$. Then, traditional convolution is computed as:

$$(m \times s) \times H' \times W' \times C \times k \times k. \tag{1}$$

In order to keep the number of feature maps output by the Ghost module consistent with traditional convolution, the convolution kernel size, stride, and padding in the Ghost module need to be the same as in traditional convolution. Assuming that the convolution of the first part of the Ghost module generates m feature maps, and each feature map generates $s-1$ new feature maps, the second part obtains a total of $m \times (s-1)$ feature maps. The total calculated amount of the Ghost module, therefore, is:

$$m \times H' \times W' \times C \times k \times k + m \times (s-1) \times H' \times W' \times k \times k. \tag{2}$$

The theoretical parameter compression ratio of Ghost module to conventional convolution is:

$$\frac{(m \times s) \times H' \times W' \times C \times k \times k}{m \times H' \times W' \times C \times k \times k + m \times (s-1) \times H' \times W' \times k \times k} = \frac{C \times s}{C + s - 1}. \tag{3}$$

In general, $C \gg s$ [38], therefore:

$$\frac{C \times s}{C + s - 1} \approx s. \tag{4}$$

From Equation (4), it can be seen that using the Ghost module for the convolution operation can reduce the calculations in the convolution process. The Ghost Bottleneck is designed to store the Ghost module, as shown in Figure 6.

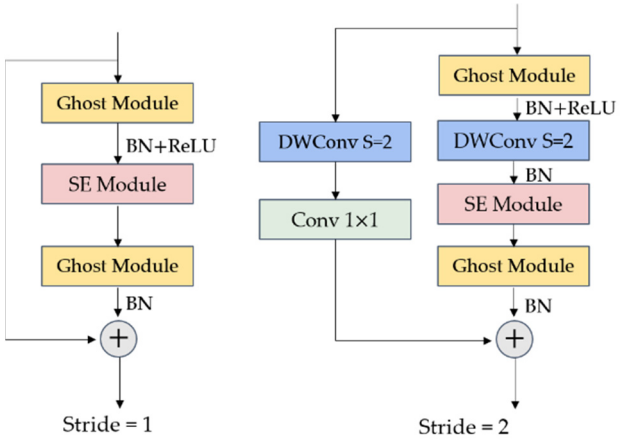


Figure 6. Bottleneck with SE module.

The Ghost Bottleneck is similar to the basic residual block in ResNet [53]. The Ghost Bottleneck consists mainly of two stacked Ghost modules. The Ghost Bottleneck with a stride of two inserts deeply separable convolutional layers to reduce the effect of feature geometry variation and decrease the parameter scale. It should be noted that the normal Ghost Bottleneck does not include an SE Attention module, which serves to enhance or reduce the feature map weights according to the importance of features.

3.3.2. GhostNet

The GhostNet network architecture was inspired by the MobileNetV3 architecture. GhostNet Bottleneck was used instead of the bottleneck structure in MobileNetV3, and the Hard-Swish activation function was changed to ReLU [41]. The network structure is shown in Figure 7.

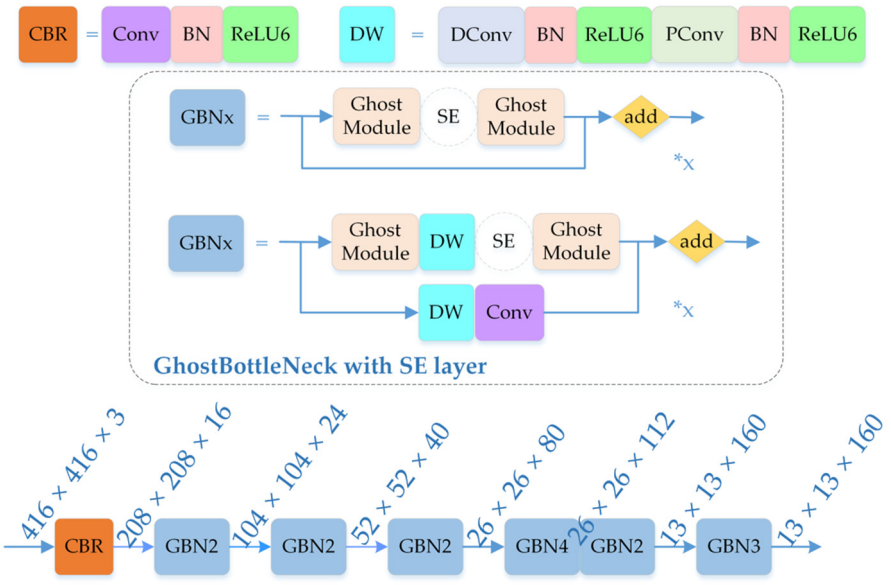


Figure 7. Diagram of GhostNet structure without fully connected layers. The ReLU activation function is used in Ghost Module. The symbol "*x" indicates the GBNx is composed of x GBN structures.

Depending on the size of the input feature map, the GhostNet Bottleneck can be divided into six stages. The last GhostNet Bottleneck in each of the first five stages has a step size of two, while the rest of the GhostNet Bottleneck has a step size of one. Three

effective feature layers at different scales (with pixel sizes of 52 × 52, 26 × 26, and 13 × 13, respectively) are obtained after preliminary feature extraction.

3.4. Coordinate Attention

Attention mechanisms are often used to focus a model on more important information, thus improving their efficiency and accuracy [54]. It has been widely used in CNNs to improve their performance [55]. Considering the weight of the model, the most widely used attention mechanism is still the SE Attention in SENet. As shown in Figure 8a, it calculates the channel attention through 2D global pooling. Unfortunately, the SE module only considers the encoding of inter-channel information and neglects the importance of location information, which is critical for many vision tasks that require target structures to be captured. Moreover, the fully connected layer in the SE module has a high number of parameters [32], which is not in line with the goal of lightweight design.

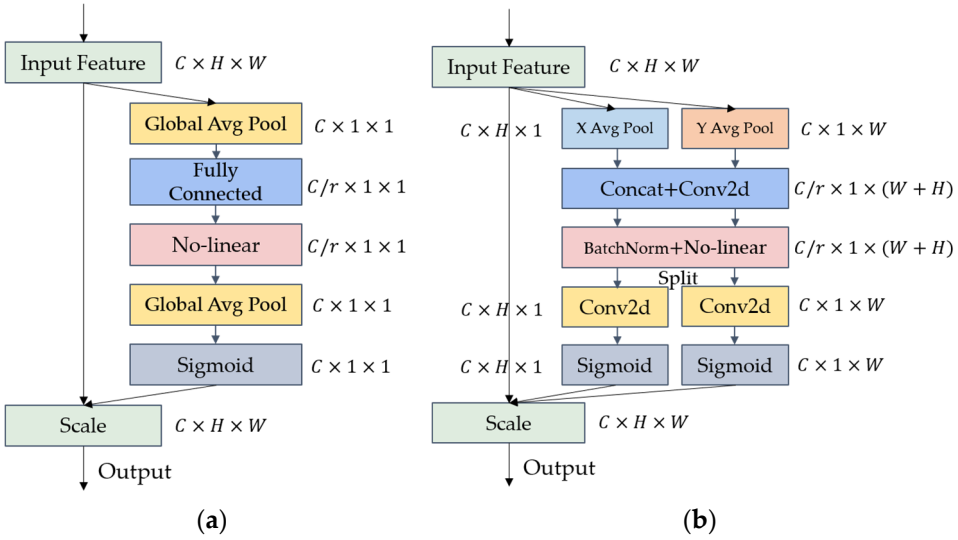


Figure 8. (**a**) SE Attention structure; and (**b**) Coordinate Attention structure.

Unlike Channel Attention, which transforms the input into a single feature vector through 2D global pooling, Coordinate Attention decomposes Channel Attention into two 1D feature encoding processes that aggregate features along different directions, as shown in Figure 8b. This has the advantage of capturing long-range dependencies along one spatial direction and retaining precise position information along the other. The generated feature maps are then encoded separately in order to form a pair of orientation-aware and position-sensitive feature maps, which can be complementarily applied to the input feature maps to enhance the representation of the target of interest [32].

A Coordinate Attention module can be seen as a computational unit used to enhance the feature representation capability. It can take any intermediate tensor $X = [x_1, x_2, \ldots, x_C] \in \mathbb{R}^{C \times H \times W}$ as input and output a tensor $Y = [y_1, y_2, \ldots, y_C]$ of the same size with enhanced representation capabilities.

As shown in Figure 2, in the actual scenario of apple detection, small and medium targets account for the majority. In the task of improving small and medium target recognition, we used the attention mechanism to inform the model where it needs to pay more attention. For this purpose, we added the Coordinate Attention module before the 52 × 52 and 26 × 26 scale inputs of the FPN in the neck structure, as shown in Figure 4. The red CA module denotes the new Coordinate Attention module. The 52 × 52 scale is mainly responsible for feature extraction of small targets, while the 26 × 26 scale is mainly responsible for feature extraction of medium targets.

3.5. Mosaic Closing Early Strategy

YOLOv4 utilizes the Mosaic data augmentation method. The main idea is to crop four images randomly and then stitch them onto a single image as training data. This has the advantage of enriching the background of the images and disguising the batch size by stitching the four images together, such that model training does not require stronger GPU performance [22].

Although Mosaic data enhancement enriches the background of the image, it leads to distortion of the input image, which is not conducive to the model recognizing real scenarios. Inspired by YOLOX [23], we turned off the Mosaic data augmentation method in the last 15% of epochs in training in order to restore the image realism.

3.6. Long-Range Target Screening

Dense apple planting with dwarf rootstock is a key model for modern orchards in China. Trees are typically arranged in rows and are narrow in shape, with a crown diameter of about 1.5–2 m and a row spacing of about 3–3.5 m [56], as shown in Figure 9. When the picking robot operates, the vision sensor inevitably picks up apples from distant rows, which adds many unnecessary processing objects, affecting the processing speed of the recognition model and increasing the probability of inspection errors. Therefore, we added a screening mechanism to the NMS section, filtering away the distant apples through consideration of the pixel size.

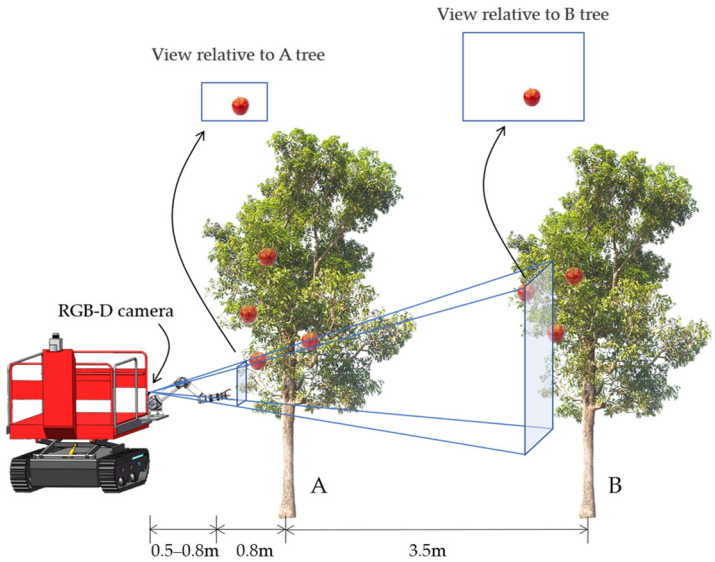


Figure 9. Diagram of apple picking robot operating environment under complex environment.

The model of the proposed picking robot arm was a Phantom X-arm6, with a maximum arm span of 0.8 m. The vision sensor was a Microsoft Azure Kinect DK. The operating range of this depth camera is 0.50–3.86 m, and the rated field of view (FOV) is 90° × 75°. To simplify the fruit tree model, the crown diameter and row spacing of the fruit tree were 1.6 m and 3.5 m, respectively. The diameter of an apple is about 80 mm. The model shown in Figure 9 was built.

Assuming that the depth camera is fixed on the base of the robot arm, the operating range of the depth camera and the operating range of the robot arm determine the shooting distance of the depth camera from the apple to be 0.5–0.8 m. Equation 5 calculates the area in terms of the actual area of the image. The actual area of the image acquired was calculated to be between 1.00 × 0.77 m^2 and 1.60 × 1.23 m^2 for tree A and between 8.00 × 6.14 m^2 and 8.60 × 6.60 m^2 for tree B:

$$\begin{cases} W = 2 \cdot \tan \frac{\theta_W}{2} \cdot d \\ H = 2 \cdot \tan \frac{\theta_H}{2} \cdot d \end{cases}, \quad (5)$$

where W refers to the width of the actual area of the image, H refers to the height of the actual area of the image, θ_W refers to the horizontal field of view of the camera, θ_H refers to the vertical field of view of the camera, and d refers to the distance of the camera relative to the fruit tree.

In order to determine the size boundaries of the pixels of the fruit on the two fruit trees in the image, the actual areas of 1.60×1.23 m^2 and 8.00×6.14 m^2 of fruit tree A and fruit tree B, respectively, were taken for the study; the former was taken as a larger area, such that the proportion of the apples in the image was as small as possible, while the latter was taken as a smaller area, such that the proportion of the apples in the image was as large as possible. The image input to the prediction model was resized into a 416×416 square. In order to avoid image distortion, our processing method involved filling with a gray bar when scaling (e.g., an image of 160×123 pixels will become 160×160 after filling with gray, then scaled to 416×416 pixels afterward). After resizing with the above method, the bounding box length of an apple with a diameter of 0.08 m is 1/20 of the actual area of 1.60×1.23 m^2 and 1/100 of the actual area of 8.00×6.14 m^2. Thus, in the image of size 416×416 pixels, the former edge length occupies a minimum of 20.8 pixels width, while the latter edge length occupies a maximum of 4.16 pixels width. Therefore, 12 pixels (as the mean value) were taken as the screening threshold. In NMS, candidate targets with a length or width of fewer than 12 pixels were screened out.

4. Results

4.1. Experimental Environment

To ensure the fairness of the experiment, the same initial training parameters were set for each group of experiments. The resolution of all input images was adjusted to 416×416 pixels. The batch size was set to 16. Due to the computing power limitation, the batch size for YOLOv4 model training was set to 8. All models were trained for 200 epochs. The learning rate was set to 1×10^{-3} for the first 50 epochs and 1×10^{-4} for the last 150 epochs. Cosine annealing was used to reduce the learning rate. The mosaic data augmentation method was enabled. The confidence threshold was set to 0.5, and the IoU threshold was set to 0.3. The experimental environment is detailed in Table 2.

Table 2. Experimental environment.

Environment	Versions or Model Number
CPU	AMD Ryzen 5 3600X, 3.80 GHz
GPU	NVIDIA RTX 3060Ti, 8 GB memory
OS	Windows 10
CUDA	CUDA 11.3
PyTorch	v1.10
Python	v3.8

4.2. Model Evaluation Metrics

In this study, objective evaluation metrics, including Precision, Recall, *F*1, *AP*, *mAP*, and others, were used to evaluate the performance of the trained apple object detection models. These were calculated as follows:

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

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

$$F1 = \frac{2PR}{P + R}, \tag{8}$$

$$AP = \int_0^1 P(R)dR \times 100\%, \tag{9}$$

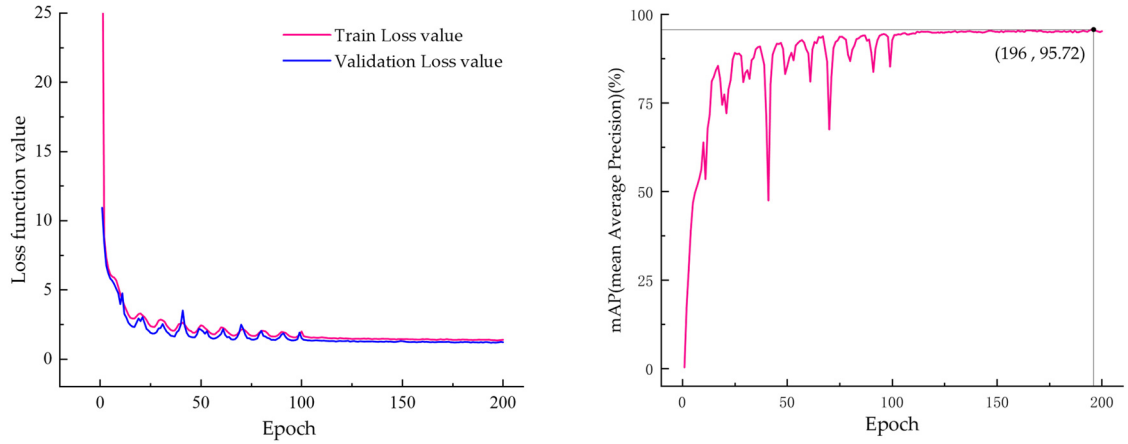
$$mAP = \frac{\sum_{n=1}^{2} AP(n)}{2} \times 100\%, \tag{10}$$

where P (precision) denotes the proportion of correct detections in all prediction boxes; R (recall) indicates the proportion of correctly detected label boxes among all label boxes; TP is the number of prediction boxes that correctly match the label box; FP is the number of prediction boxes with incorrect prediction; FN is the number of labeled boxes with missed detection; $F1$ is the harmonic mean of precision and recall, which measures the actual average of these two metrics; AP denotes the average precision value for each category of apples; and mAP denotes the average precision value for the two categories of apples, which comprehensively reflects the detection performance of the model. $F1$, P, and R are all macro-averaged metrics. In addition to the above metrics, the number of network parameters and the size of the model were also introduced as evaluation criteria. The average speed required to infer the same image 300 times was used to calculate the FPS of the model. In this study, the main focus was on the mAP value as a performance metric. Therefore, the weight file with the highest mAP for each model was taken as the final result.

4.3. Experimental Results

4.3.1. Ablation Experiments

To verify the effect of each of the improvements on the model performance, ablation experiments were conducted. Figure 10 shows the trend of the loss value and mAP of the model during training.



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

Figure 10. Network training results: (**a**) Training and validation loss curves; and (**b**) mAP in the test set.

Table 3 provides the results of the ablation experiments on YOLOv4. In this study, we use "Model X" to refer to the model after adding one of the improvements, where "X" corresponds to the serial number in Table 3; therefore, Model 1 refers to the baseline model YOLOv4. We denote Model 7 as "Improved YOLOv4", which was the final version of the network after we added the improvements. A total of six improvements were made on the basis of Model 1, described as follows.

Table 3. Results of the ablation experiments on the test set.

No.	Model	mAP (%) (*, *)	Weight Size (MB) (*)	Parameters (M) (*)	Speed (FPS) (*, *)
1	YOLOv4 baseline	92.27	244.00	64.40	39.50
2	+GhostNet	93.83 (+1.56, +1.56)	150.00 (61.48%)	39.70 (61.65%)	42.10 (+2.60, +2.60)
3	+Depth-wise separable convolution	94.00 (+1.73, +0.17)	42.50 (17.42%)	11.40 (17.70%)	44.50 (+5.00, +2.40)
4	+Coordinate Attention	94.48 (+2.21, +0.48)	37.80 (15.49%)	10.20 (15.84%)	45.30 (+5.80, +0.80)
5	+Improving S&M target detection	94.77 (+2.50, +0.29)	37.90 (15.53%)	10.20 (15.84%)	45.20 (+5.70, −0.10)
6	+Closing last 15% phase of Mosaic	94.97 (+2.70, +0.20)	37.90 (15.53%)	10.20 (15.84%)	45.10 (+5.60, −0.10)
7	+Long-range target screening	95.72 (+3.45, +0.75)	37.90 (15.53%)	10.20 (15.84%)	45.20 (+5.70, +0.10)

Note: The symbols (*, *) indicate the changes in value compared with Model 1 and the previous model, respectively. The symbol (*) indicates the percentage value compared with Model 1. S&M means small and medium size.

1→2. The backbone network was replaced. GhostNet, a lightweight feature extraction network, was used as the backbone network of the model. The results indicated that the weight size and the number of parameters of the model were 61.48% and 61.65% of Model 1, respectively. At the same time, instead of degrading the performance of the modified model, the mAP improved by 1.56% (to 93.83%). The inference speed of the model was also improved by 2.6 FPS over Model 1. Considering such an effective gain, we believe that GhostNet is more efficient than the original CSPDarknet53 in YOLOv4 for the considered task.

2→3. To further reduce the number of network parameters, depth-wise separable convolutions were introduced in the neck and YOLO head of the model. The weight size and number of parameters of the model were 17.42% and 17.70% of Model 1, respectively; therefore, the size and computation requirements of the model were greatly reduced. The mAP was further improved to 94.00% (1.73% and 0.17% higher than Models 1 and 2, respectively). The detection speed was increased by 2.4 FPS over Model 2 and by 5.0 FPS over Model 1. These results indicate that the introduction of depth-wise separable convolution can reduce memory consumption, reduce the number of model parameters, and improve the speed of model recognition without affecting the accuracy.

3→4. We replaced the SE Attention module in the GhostNet backbone network with a Coordinate Attention module to form CA-GhostNet. In this way, the number of model parameters was reduced by about 2%. The mAP performance was improved by 0.48% over Model 3 and by 2.21% over Model 1, and the detection speed improved by 0.8 FPS over Model 3 and 5.8 FPS over Model 1. For comparison, we tested another lightweight attention mechanism, ECA-net. Although the number of model parameters decreased by 2.32%, the mAP of the model decreased by 0.32% (to 93.68%). Therefore, we ultimately chose to use Coordinate Attention.

4→5. We added the Coordinate Attention module before the 52 × 52 pixels and 26 × 26 pixels inputs of the FPN. The number of parameters required for a Coordinate Attention module is 2×10^{-3} M. With the addition of two of these modules, the weight size and the number of parameters remained essentially unchanged. The mAP was increased by 0.29% over Model 4 and by 2.50% over Model 1, reaching 94.77%, with a slight decrease in detection speed (by 0.1 FPS). This was due to the additional inference steps caused by the addition of the new module, resulting in a decreased inference speed. The specific performance tests for this improvement are detailed in Section 4.3.3.

5→6. We turned off the Mosaic data enhancement method in the last 15% epochs of the training stage in order to restore image realism. This improved the mAP performance of the model by 0.20% and decreased the model inference speed by 0.1 FPS. The improvement to the data augmentation strategy is an adjustment to the pre-processing stage and, so, does not affect the inference speed of the model. Therefore, this can be considered a random error within reasonable limits.

6→7. To screen out non-target rows of apples from the view field of the visual sensor, we added a screening mechanism to the NMS part of the model. We screened out candidate

targets with apples in the image that are less than 12 pixels in length or width. The increase in mAP was significant, increasing by 0.75% to 95.72%. This process increased the model inference speed by 0.1 FPS as a portion of candidate boxes were first screened out before NMS, thus reducing the number of frames requiring NMS processing.

4.3.2. Comparison Experiment of the Same Type of Object Detection Algorithm

To verify the effectiveness of Improved YOLOv4, we selected two classical object detection algorithms, YOLOX-s [23] and E-YOLOv3 [21], with weight sizes similar to that of Improved YOLOv4.E-YOLOv3 replaces the backbone of YOLOv3 with EfficientNetB0, thus approaching the size of Improved YOLOv4. YOLOX-s is 3.6 MB smaller than Improved YOLOv4, while E-YOLOv3 is 2.7 MB larger than Improved YOLOv4. In the following comparison experiments, both YOLOX-s and E-YOLOv3 use their original unmodified versions compared with Improved YOLOv4.

Comparison Experiments in the Total Test Set

The other two object detection algorithms were trained with the same data set described in this study. Figure 11 shows the P–R plots of the three models. It can be intuitively seen that the area under the curve of the Improved YOLOv4 model was larger than that of the other two classical object detection models for both labels, indicating that the Improved YOLOv4 model has higher average precision and better overall performance. The test results for the three methods are given in Table 4.

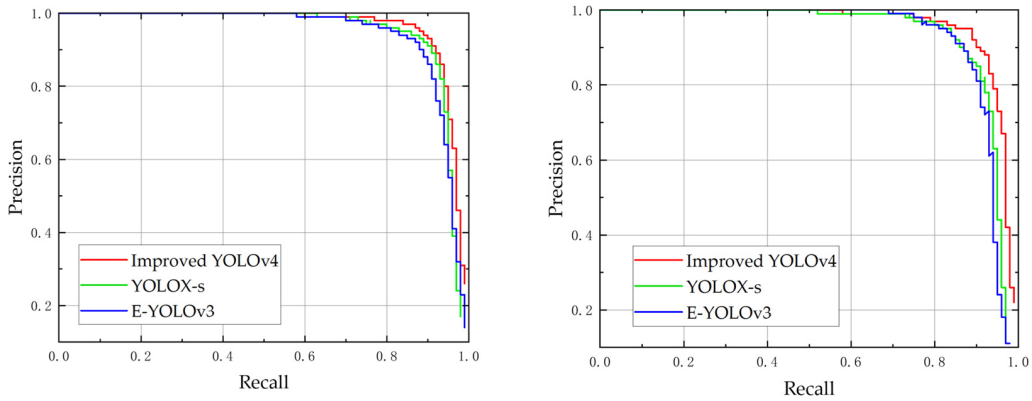


Figure 11. P–R curve for comparison: (**a**) Plot of ripe_apple; and (**b**) plot of unripe_apple.

Table 4. Comparison test results of three models.

Model	Weight Size (MB)	mAP (%)	AP1 (%)	AP2 (%)	F1	Precision (%)	Recall (%)	Speed (FPS)
YOLOX-s	34.3	93.90	94.46	93.34	0.90	93.13	86.65	52.3
E-YOLO v3	40.6	93.39	94.13	92.65	0.89	94.08	84.43	49.2
Improved YOLOv4	37.9	95.72	95.91	95.54	0.91	95.32	86.54	45.2

In terms of detection accuracy, Improved YOLOv4 has the highest mAP, which was 1.82% higher than YOLOX-s and 2.33% higher than E-YOLOv3. The precision was 95.32%, and the F1 score was 0.91. AP1 reached 95.91%, and AP2 reached 95.54%, higher than those of the other methods; namely, the AP1 and AP2 scores were 1.45% and 2.2% higher than those of YOLOX-s, and 1.78% and 2.89% higher than those of E-YOLOv3, respectively. The recall was 86.54%, which was slightly lower than that of YOLOX-s. The recall is related to the setting of the IoU value, as well as the precision. However, the F1 metric, which is a combined measure of precision and recall, had a higher value. This indicates that Improved YOLOv4 performed better overall. In particular, the AP for unripe apples was significantly

better than other algorithms. There were large differences between the detection results of the other two algorithms for ripe and unripe apples; the reason for this relative difficulty in recognition is that the color of the unripe apples is similar to that of the background leaves, and the apples are small and grow densely. In contrast, Improved YOLOv4 presented strong detection performance, enhancing the recognition of small and medium targets.

Comparative Experiments with Different Illumination Conditions

Next, we examined the robustness of the models under different illumination conditions. The test set included three conditions—front light, side light, and back light—with 60 images of apples in each condition. To avoid the influence of the number of apples on the experimental results, the number of apples in each category of pictures was ensured to be distributed similarly.

After completing the tests under different illumination conditions, the performance indicators shown in Table 5 were obtained. It can be seen that the performance of Improved YOLOv4 was optimal, regardless of the illumination conditions. The highest mAP value (95.99%) was obtained by Improved YOLOv4 under the side light condition, which was higher than the mAP value on the entire test set. Under the back light condition, Improved YOLOv4 achieved the lowest mAP value of 95.39%. Approximately the same results occurred with YOLOX-s and E-YOLOv3. This indicates that the models can obtain higher recognition accuracy under the side light condition, as apples have clear texture and uniform surface light intensity under side lighting, while the back light condition causes some interference with the detection.

Table 5. Comparison of the three models in different illumination data sets.

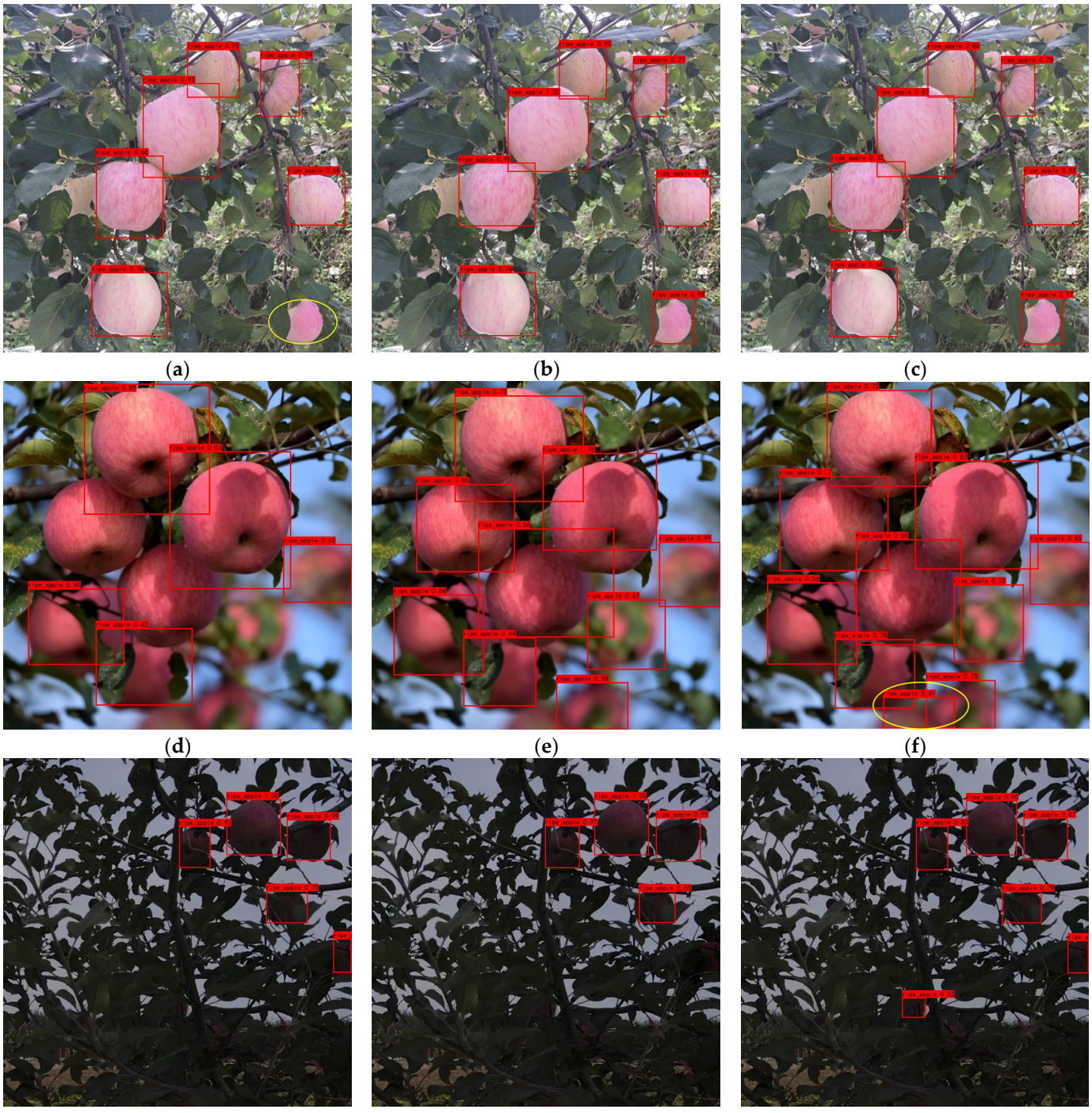
Illumination Condition	Model	mAP (%) (*)	F1	Precision (%)	Recall (%)
Front light	YOLOX-s	93.90 (−1.82)	0.90	93.13	86.65
	E-YOLO v3	93.39 (−2.33)	0.89	94.07	84.43
	Improved YOLOv4	95.72	0.91	95.32	86.54
Side light	YOLOX-s	94.33 (−1.66)	0.89	91.90	86.36
	E-YOLO v3	94.02 (−1.97)	0.90	94.43	84.88
	Improved YOLOv4	95.99	0.91	94.78	86.74
Back light	YOLOX-s	93.21 (−2.18)	0.88	92.17	84.12
	E-YOLO v3	92.27 (−3.12)	0.88	93.26	83.54
	Improved YOLOv4	95.39	0.90	94.73	85.47

Note: The symbol (*) indicates the change in value compared with Improved YOLOv4.

It is worth noting that Improved YOLOv4 had a greater advantage over the other two algorithms under the back light condition. The mAP value of Improved YOLOv4 was 2.18% and 3.12% higher than that of YOLOX-s and E-YOLOv3, respectively, under this condition. It is conceivable that an apple picking robot implemented with Improved YOLOv4 may perform better in back-lit and low illumination environments. Therefore, it can well-adapt to the actual operating conditions in complex environments.

Figure 12 shows the comparison of the recognition effects of the three models under different illumination conditions. In the front light condition, E-YOLOv3 failed to recognize the apple in the yellow-marked box in Figure 12a, with little difference between the remaining two algorithms. Under the side light condition, E-YOLOv3 did not perform well, with two more obvious apples not recognized, while YOLOX-s and Improved YOLOv4 were able to recognize apples that were out of focus in the camera. As shown in the yellow marked box in Figure 12f, Improved YOLOv4 was able to recognize more apples, proving that Improved YOLOv4 is more robust. Under the back light condition, Improved YOLOv4 also recognized more apples. In general, Improved YOLOv4 presented good recognition results, regardless of the illumination conditions. Moreover, it can be seen that the recognition results of Improved YOLOv4 are closer to the minimum enclosing rectangle of the

apple. This indicates that Improved YOLOv4 can produce more accurate apple locations than the other two algorithms.



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

(**d**) (**e**) (**f**)

(**g**) EfficientNetB0-YOLO v3 (**h**) YOLOX-s (**i**) Improved YOLOv4

Figure 12. Comparison of recognition effects of three models under different illumination conditions: (**a**–**c**) the recognition effect in the front light condition; (**d**–**f**) the recognition effect in the side light condition; and (**g**–**i**) the recognition effect in the back light condition.

4.3.3. Validation for the Improvement of Small and Medium Target Recognition Capability

The experiments detailed in this section were conducted to validate the enhanced effect of our model on the small target detection ability. Using the Pycocotools tool, the

target detection ability was evaluated at different scales. The data presented in Table 6 were derived at IoU = 0.50:0.95. The subscripts are defined as follows: S means small target (area $\leq 32^2$), M means medium target (32^2 < area $\leq 96^2$), L means large target (area > 96^2), and area denotes the number of pixels [57].

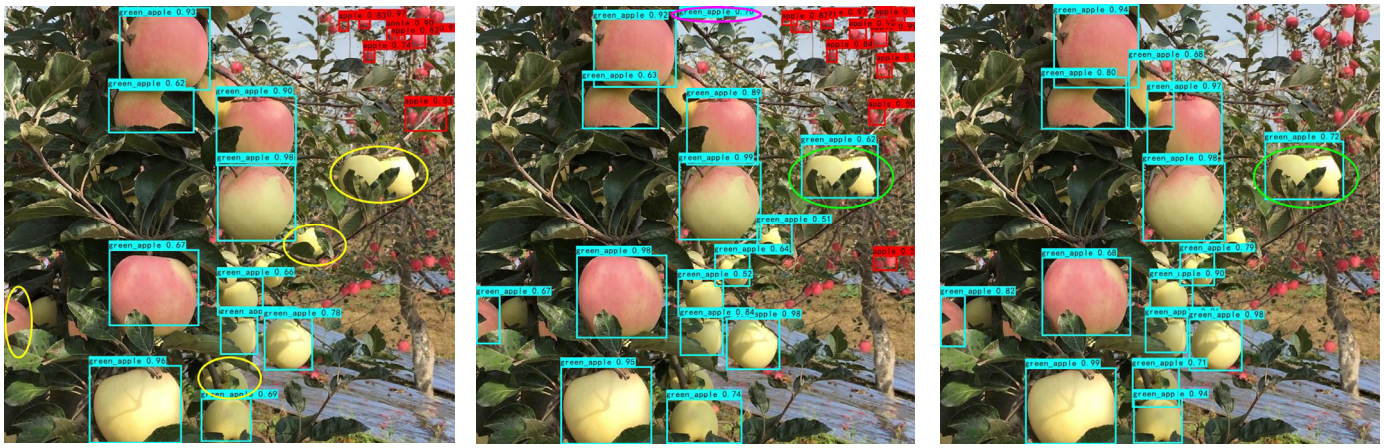
Table 6. Validation results for the improvement of small and medium target recognition capability.

Model	AP_S (%)	AP_M (%)	AP_L (%)	AR_S (%)	AR_M (%)	AR_L (%)
Model 1	0.170	0.540	0.681	0.335	0.618	0.730
Model 4	0.162	0.552	0.718	0.347	0.631	0.763
Model 5	0.181	0.560	0.713	0.356	0.642	0.758

As shown in Table 6, Model 5 effectively enhanced the performance for both medium targets and small targets, presenting an 11.7% and 2.6% improvement in AP and AR for small targets and a 1.5% and 1.7% improvement in AP and AR for medium targets, respectively, compared with the pre-improvement period. For small targets, AP and AR were improved by 6.5% and 6.3%, respectively, over Model 1. Meanwhile, for medium targets, AP and AR are improved by 3.7% and 3.9%, respectively, over Model 1. We observed a slight decrease in recognition for large targets, which may be a result of the added attention mechanism focusing more on smaller pixel feature extraction.

4.3.4. Validation Experiments on Target Screening Strategy

The recognition effect of the model with the addition of the long-range target screening module is shown in Figure 13. The original image contained a tree in a non-target row. The apples in the target row had not been colored for long enough and were all unripe apples distributed on the left side of the picture. The apples on the other fruit tree were in good ripening condition, distributed on the right side of the picture. The recognition effect of Model 1 is shown in Figure 13a, where the obvious unsuccessful apples are marked. Compared with Model 6 and Improved YOLOv4, the recognition success rate of Model 1 was not high, and there were missed detections. Figure 13c clearly shows that the apples in the non-target rows had been screened out. The purple marker in Figure 13b denotes the misidentified targets, which were also successfully screened out by the screening mechanism. From the recognition results, it can be seen that Model 7 has an excellent performance in identifying and distinguishing some overlapping dense targets.



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

Figure 13. Experimental validation results of target screening: (**a**) The recognition effect of YOLO v4; (**b**) the recognition effect of Model 6; and (**c**) the recognition effect of Model 7 with the addition of the long-range target screening module.

However, both Figure 13b,c did not distinguish the two unripe apples marked in green. This was probably due to the shape features disappearing because of the lack of color features of the unripe apples in the bright light environment, which resulted in failing to distinguish the two apples.

5. Discussion

5.1. Improved YOLOv4

In this study, a recognition method based on YOLOv4 was proposed to address the problems of low accuracy, real-time performance, and poor robustness of traditional methods for fruit recognition in complex environments such as orchards.

The YOLO series are universal object detection models that have presented an excellent performance on a COCO data set with 80 object classes [57]. In our scenario, only two kinds of targets in the environment needed to be identified. Therefore, we consider that the apple detection model does not need to learn too many shape and color features. In this context, the performance of traditional object detection algorithms is relatively redundant. The original CSPDarknet53 backbone network of YOLOv4 was replaced by GhostNet, a lightweight feature extraction network, and the traditional convolution was replaced by depth-wise separable convolution with fewer parameters. These two improvements increased the mAP by 1.73% while reducing the weight size and number of parameters of the model by 82.58% and 82.30%, respectively. The speed of inference increased by 5 FPS as a result of the lower number of parameters. It can be demonstrated that, for tasks with a few kinds of targets, it is more efficient to use the lightweight YOLOv4. Models with a larger number of parameters may be overfitted in simple object-detection tasks.

The actual environment of an orchard is complex, usually with fruit growing in dense clusters, and it is common for targets to be shaded by other fruit, leaves, and branches. Different illumination effects cause the fruit to appear in the images with unnatural color features. These are challenges for the task of recognition. The DIOU_nms of YOLOv4 can effectively regress out overlapping targets. On this basis, we used a Coordinate Attention module in the backbone network, as well as before the small and medium target feature extraction network, in order to reduce the number of parameters and improve the performance. After the experimental validation detailed in Section 4.3.3, Model 5 was improved in terms of small and medium target recognition.

In addition, inspired by YOLOX, the Mosaic data augmentation strategy was turned off early, which improved the mAP on the test set by 0.20%, indicating that the diversity of target distribution in the training set should not be pursued throughout the training phase of the model, as proper restoration of natural scenes is beneficial for training.

We look at the relationship between target and model detection performance from a new perspective. Existing target detection algorithms are generic and can detect a variety of objects. For us, we only need to be able to detect one kind of target: apples. Therefore, classic algorithms can appear to be over-performing. They may detect not only the target apples but also apples that are far away, which cannot be picked by the robotic arm. These apples with small pixel areas should be positive samples. However, as we generally do not label the small pixel apples in the distance, they are, in fact, misclassified as "FP" and are negative samples, thus affecting the precision. We calculated pixel value thresholds for the apples that can and cannot be picked in the pictures. According to the calculation in Section 3.6, the distant targets displayed in the pictures have a side length of fewer than 4.16 pixels, as reflected in the actual image with smaller pixels; meanwhile, the side length of each apple in the image that can be picked is typically greater than 20.8 pixels. Therefore, the screening threshold was set as 12 pixels. This threshold was applied in the long-range target screening strategy we proposed, resulting in a significant increase in mAP. We believe that this strategy can be applied to other fruit and vegetable objectives in the picking task.

5.2. Algorithm Comparison

In general, two-stage object-detection methods have higher detection accuracy, while one-stage object detection methods have faster inference speed [29]. With the development of one-stage object detection algorithms, they have also reached high levels of detection accuracy. Considering the need to ensure real-time detection and device friendliness, we did not consider two-stage target detection algorithms for comparison, nor the traditional SSD algorithm. The comparison algorithms were selected from the current, more advanced YOLOX family, which performs better than YOLOv5. Additionally, a more classic YOLOv3 model was also chosen. On the whole data set, YOLOv4 improved according to our method and presented the highest mAP, 1.82% higher than that of YOLOX-s and 2.33% higher than that of E-YOLOv3. In the comparison experiments under different illumination conditions, the performance of Improved YOLOv4 was optimal in all lighting conditions. Compared with YOLOX-s and E-YOLO v3, it is obvious that the bounding boxes generated by Improved YOLOv4 were closer to the actual minimum enclosing rectangles, as shown in Figure 12, which is more beneficial for returning accurate coordinate values to the picking robot.

In terms of detection speed, Improved YOLOv4 was lower than the other two algorithms. The detection speed was lower than YOLOX-s as this model is less computationally intensive than Improved YOLOv4 and uses the Anchor-free approach, which requires a shorter inference time. The possible reason for the lower detection speed than E-YOLOv3 is that the enhanced feature extraction network of the algorithm utilizes a unidirectional top-down fusion FPN feature pyramid structure, while Improved YOLOv4 uses a bidirectional fusion structure, which leads to a slower inference process. However, the detection accuracy of Improved YOLOv4 was higher compared with the other two algorithms. At present, an advanced picking robot can pick at a rate of 5.5 s/fruit [10]; therefore, Improved YOLOv4 can meet the real-time requirements of the actual picking operation.

From the comparison results, it is clear that there is a contradiction between the complexity of the network structure and the model detection performance. In light of this, we had to compromise between model detection performance and detection speed. The effectiveness of the model in detecting difficult targets was improved, within a certain range, while losing the original inference speed. Therefore, in future research, we will focus on improving the algorithm, and using more efficient strategies to reduce the FPS loss.

6. Conclusions

This study was dedicated to improving the detection performance of traditional target-detection models in complex, real orchard environments while streamlining the number of model parameters. In this regard, we carried out the following two main areas of work. In terms of lightweight modification of YOLOv4, a lightweight feature-extraction network, GhostNet, was used. The neck and YOLO head structures were reconstructed by introducing deep separable convolution. In terms of detection performance improvement, a Coordinate Attention module was introduced into the GhostNet network, as well as added before the 52×52 and 26×26 scale inputs of the FPN structure (responsible for small and medium target feature extraction, respectively), which enhanced the extraction capability for medium and small target features. The Mosaic data enhancement strategy was turned off in the last 15% of epochs in training in order to restore the realism of the input images. A long-range target screening strategy was proposed for standardized dense planting apple orchards with dwarf rootstock in order to remove apples from non-target rows, thus improving the detection performance and recognition speed.

Overall, the lightweight Improved YOLOv4 proposed in this study presented a 3.45% improvement in mAP compared with the YOLOv4 network, while the weight size was only 15.53% of that of the baseline model, and the number of parameters was only 15.84% of the baseline model. The model size and computation were about 1/6 that of YOLOv4. Furthermore, the detection speed of Improved YOLOv4 was 45.2 FPS, 14.43% higher than that of YOLOv4. Thus, Improved YOLOv4 is much more efficient than YOLOv4.

To further verify the effectiveness and feasibility of the proposed Improved YOLOv4 model, two sets of comparison experiments were designed, on the total test set and different illumination test sets, by selecting two similarly sized algorithms: YOLOX-s and E-YOLOv3. In the total test set, Improved YOLOv4 has the highest mAP, 1.82% higher than that of YOLOX-s and 2.33% higher than that of E-YOLOv3. The performance of Improved YOLOv4 was optimal under all three illumination conditions. In the front light condition, the mAP of Improved YOLOv4 was 1.82% and 2.33% higher than the other two algorithms, respectively. In the side light condition, the mAP of Improved YOLOv4 was 1.66% and 1.97% higher than these algorithms, respectively. Finally, in the back light condition, the mAP of Improved YOLOv4 was 2.18% and 3.12% higher than that of the other two algorithms, respectively. Therefore, Improved YOLOv4 was more improved over the other two algorithms under the back light condition, and all three algorithms had their respective best detection performance under the side light condition.

The proposed deep learning-based object detection technique was generally robust to illumination, and changes in illumination had less of an effect on the detection efficiency. This indicates that Improved YOLOv4 not only has a substantially reduced number of parameters but also is suitable for working in standardized dense planting apple orchards with dwarf rootstock. It provides a fresh approach for achieving highly effective fruit picking by robots. In the future, we intend to focus on constructing apple picking prototypes and conducting picking experiments to determine the directions for optimizing such robots.

Author Contributions: Conceptualization, F.K., Y.W. and C.Z.; methodology, F.K., Y.W. and C.Z.; investigation, F.K. and Y.W.; resources, F.K. and Y.W.; data curation, C.Z.; writing—original draft preparation, C.Z.; writing—review and editing, F.K., Y.W. and C.Z.; supervision, F.K. and Y.W.; funding acquisition, F.K. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NingXia key research and development program (Grant No. 2019BBF02009).

Data Availability Statement: The raw data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Acknowledgments: Thanks to the Key Lab of State Forestry and Grassland Administration on Forestry Equipment and Automation for providing the equipment for model training and data analysis.

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

References

1. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
2. Li, T.; Feng, Q.; Qiu, Q.; Xie, F.; Zhao, C. Occluded Apple Fruit Detection and Localization with a Frustum-Based Point-Cloud-Processing Approach for Robotic Harvesting. *Remote Sens.* **2022**, *14*, 482. [CrossRef]
3. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Lian, J.; Zheng, Y. Detection and Segmentation of Overlapped Fruits Based on Optimized Mask R-CNN Application in Apple Harvesting Robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [CrossRef]
4. Kuznetsova, A.; Maleva, T.; Soloviev, V. Using YOLOv3 Algorithm with Pre- and Post-Processing for Apple Detection in Fruit-Harvesting Robot. *Agronomy* **2020**, *10*, 1016. [CrossRef]
5. Lin, G.; Tang, Y.; Zou, X.; Li, J.; Xiong, J. In-Field Citrus Detection and Localisation Based on RGB-D Image Analysis. *Biosyst. Eng.* **2019**, *186*, 34–44. [CrossRef]
6. 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]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
8. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 818–833.
9. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.

10. Williams, H.A.M.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic Kiwifruit Harvesting Using Machine Vision, Convolutional Neural Networks, and Robotic Arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [CrossRef]
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *arXiv* **2014**, arXiv:1311.2524.
12. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef]
14. Sun, J.; He, X.; Ge, X.; Wu, X.; Shen, J.; Song, Y. Detection of Key Organs in Tomato Based on Deep Migration Learning in a Complex Background. *Agriculture* **2018**, *8*, 196. [CrossRef]
15. Zhang, J.; He, L.; Karkee, M.; Zhang, Q.; Zhang, X.; Gao, Z. Branch Detection for Apple Trees Trained in Fruiting Wall Architecture Using Depth Features and Regions-Convolutional Neural Network (R-CNN). *Comput. Electron. Agric.* **2018**, *155*, 386–393. [CrossRef]
16. Bargoti, S.; Underwood, J. Deep Fruit Detection in Orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3626–3633.
17. Grilli, E.; Battisti, R.; Remondino, F. An Advanced Photogrammetric Solution to Measure Apples. *Remote Sens.* **2021**, *13*, 3960. [CrossRef]
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
20. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
21. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
22. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
23. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
24. 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]
25. Lu, S.; Chen, W.; Zhang, X.; Karkee, M. Canopy-Attention-YOLOv4-Based Immature/Mature Apple Fruit Detection on Dense-Foliage Tree Architectures for Early Crop Load Estimation. *Comput. Electron. Agric.* **2022**, *193*, 106696. [CrossRef]
26. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and Accurate Green Pepper Detection in Complex Backgrounds via an Improved Yolov4-Tiny Model. *Comput. Electron. Agric.* **2021**, *191*, 106503. [CrossRef]
27. Liu, M.; Jia, W.; Wang, Z.; Niu, Y.; Yang, X.; Ruan, C. An Accurate Detection and Segmentation Model of Obscured Green Fruits. *Comput. Electron. Agric.* **2022**, *197*, 106984. [CrossRef]
28. Sun, Q.; Chai, X.; Zeng, Z.; Zhou, G.; Sun, T. Noise-Tolerant RGB-D Feature Fusion Network for Outdoor Fruit Detection. *Comput. Electron. Agric.* **2022**, *198*, 107034. [CrossRef]
29. Li, W.; Feng, X.S.; Zha, K.; Li, S.; Zhu, H.S. Summary of Target Detection Algorithms. *J. Phys. Conf. Ser.* **2021**, *1757*, 012003. [CrossRef]
30. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
31. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 19 June 2020; pp. 11531–11539.
32. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, arXiv:2103.02907.
33. Han, L.; Zhao, Y.; Lv, H.; Zhang, Y.; Liu, H.; Bi, G. Remote Sensing Image Denoising Based on Deep and Shallow Feature Fusion and Attention Mechanism. *Remote Sens.* **2022**, *14*, 1243. [CrossRef]
34. Kim, M.; Jeong, J.; Kim, S. ECAP-YOLO: Efficient Channel Attention Pyramid YOLO for Small Object Detection in Aerial Image. *Remote Sens.* **2021**, *13*, 4851. [CrossRef]
35. Jia, W.; Zhang, Z.; Shao, W.; Ji, Z.; Hou, S. RS-Net: Robust Segmentation of Green Overlapped Apples. *Precis. Agric.* **2022**, *23*, 492–513. [CrossRef]
36. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1. [CrossRef]
37. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
38. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
39. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv* **2019**, arXiv:1801.04381.

40. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244.
41. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
42. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083.
43. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *arXiv* **2018**, arXiv:1807.11164.
44. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. *arXiv* **2016**, arXiv:1602.07360.
45. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
46. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. *arXiv* **2017**, arXiv:1608.08710.
47. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.02357.
48. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
49. Wu, L.; Ma, J.; Zhao, Y.; Liu, H. Apple Detection in Complex Scene Using the Improved YOLOv4 Model. *Agronomy* **2021**, *11*, 476. [CrossRef]
50. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
51. Fruits 360. Available online: https://www.kaggle.com/datasets/moltean/fruits (accessed on 20 June 2022).
52. Zhao, S.; Zhang, S.; Lu, J.; Wang, H.; Feng, Y.; Shi, C.; Li, D.; Zhao, R. A Lightweight Dead Fish Detection Method Based on Deformable Convolution and YOLOV4. *Comput. Electron. Agric.* **2022**, *198*, 107098. [CrossRef]
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
54. Chaudhari, S.; Mithal, V.; Polatkan, G.; Ramanath, R. An Attentive Survey of Attention Models. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–32. [CrossRef]
55. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention Mechanisms in Computer Vision: A Survey. Comp. *Vis. Media* **2022**, *8*, 331–368. [CrossRef]
56. Hao, J.; Li, X.; Yan, X.; Fen, J.; Suo, X. Apple Dwarf Rootstock Dense Planting and Rootstock Combination. *Hebei Fruits* **2021**, *3*, 27–28+30. [CrossRef]
57. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.