horticulturae

Article

Lightweight Apple Detection in Complex Orchards Using YOLOV5-PRE

Lijuan Sun, Guangrui Hu, Chao Chen, Haoxuan Cai, Chuanlin Li, Shixia Zhang and Jun Chen *

College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China
* Correspondence: chenjun_jdxy@nwsuaf.edu.cn; Tel.: +86-29-8709-1867

Abstract: The detection of apple yield in complex orchards plays an important role in smart agriculture. Due to the large number of fruit trees in the orchard, improving the speed of apple detection has become one of the challenges of apple yield detection. Additional challenges in the detection of apples in complex orchard environments are vision obstruction by leaves, branches and other fruit, and uneven illumination. The YOLOv5 (You Only Look Once version 5) network structure has thus far been increasingly utilized for fruit recognition, but its detection accuracy and real-time detection speed can be improved. Thus, an upgraded lightweight apple detection method YOLOv5-PRE (YOLOv5 Prediction) is proposed for the rapid detection of apple yield in an orchard environment. The ShuffleNet and the GhostNet lightweight structures were introduced into the YOLOv5-PRE model to reduce the size of the model, and the CA (Coordinate Attention) and CBAM (Convolutional Block Attention Module) attention mechanisms were used to improve the detection accuracy of the algorithm. After applying this algorithm on PC with NVIDIA Quadro P620 GPU, and after comparing the results of the YOLOv5s (You Only Look Once version 5 small) and the YOLOv5-PRE models outputs, the following conclusions were obtained: the average precision of the YOLOv5-PRE model was 94.03%, which is 0.58% higher than YOLOv5s. As for the average detection time of a single image on GPU and CPU, it was 27.0 ms and 172.3 ms, respectively, which is 17.93% and 35.23% higher than YOLOV5s. Added to that, the YOLOv5-PRE model had a missed detection rate of 6.54% when being subject to back-light conditions, and a false detection rate of 4.31% when facing front-light conditions, which are 2.8% and 0.86% higher than YOLOv5s, respectively. Finally, the feature extraction process of the YOLOv5-PRE model was presented in the form of a feature map visualization, which enhances the interpretability of the model. Thus, the YOLOv5-PRE model is more suitable for transplanting into embedded devices and adapts well to different lighting conditions in the orchard, which provides an effective method and a theoretical basis for the rapid detection of apples in the process of rapid detection of apple yield.

Keywords: YOLOv5; apple yield; lightweight network; attention mechanism; image processing

check for updates

Citation: Sun, L.; Hu, G.; Chen, C.; Cai, H.; Li, C.; Zhang, S.; Chen, J. Lightweight Apple Detection in Complex Orchards Using YOLOV5-PRE. *Horticulturae* **2022**, *8*, 1169. https://doi.org/10.3390/horticulturae8121169

Academic Editor: Stefano Poni

Received: 12 November 2022
Accepted: 6 December 2022
Published: 8 December 2022

1. Introduction

Apple is the third largest fruit crop in the world. The plantation area of apples in China is about two million hectares, the output ranks first in the world, and both the production and consumption scales constitute more than 50% of the world numbers [1,2]. In order to improve the ventilation and light transmission of the tree crown and improve the fruit quality, the fruit trees in the new apple orchard are mainly dwarf and spindle trees [3]. The development of the fruit tree industry should be accompanied by the top-level design of combining agricultural big data and market analysis. Rapid detection of apple yield can support the decisions of orchard management in terms of labor demand, storage, transportation and marketing, which is of great significance in regulating the relationship between market supply and demand, and the economic benefits of fruit farmers. The unmanned rapid detection of apple yield also has broad application prospects in future unmanned farms. The visual recognition of apples is the most intuitive and effective means to achieve rapid yield detection. The growth status and the number of fruits of each fruit tree are different. Therefore, to achieve a higher yield detection accuracy, yield detection should be carried out for each fruit tree. The corresponding yield of each tree corresponds to the growth and development of the fruit tree. The yield map established by this yield prediction method can provide a reference for the next year's fruit tree cultivation plan. Due to the large number of fruit trees in the orchard, yield detection needs higher detection efficiency. The apple detection model with fast reasoning speed, high precision and small size can achieve a higher apple yield detection speed, and is more suitable for running in different embedded devices.

The visual recognition algorithm of fruit has always been the key subject of rapid detection of fruit yield research. Several local and international researchers have developed various detection algorithms for the recognition of different types of fruits, including tomatoes [4–7], grapes [8–12], lychee [13–15], apples [16–18], and so on. Early target fruit detection methods mainly relied on traditional image processing or machine learning methods to identify fruit according to its color, shape, texture, and other features [2]. For instance, Yu et al. [13] used color and texture features to train the random forest binary classification model to identify litchi fruit, and the proposed method achieved a recognition accuracy rate of 89.92% for the green litchi and 94.50% for red litchi. In addition, Syazwani et al. [19] demonstrated good fruit counting performance using ANN-GDX (Artificial Neural Network, Variable Learning Rate Backpropagation) classification to detect pineapple crown images, with an accuracy of 94.4%. As for Fu et al. [20], they worked on the combined HOG and LBP (Oriented Gradient and Local Binary Pattern) texture feature while using the SVM (Support Vector Machine) classification algorithm, and they reached an 89.63% average scale detection rate for bananas and an average detection time of 1.325 s, where the shortest detection time was 0.343 s. To sum up and refer to the different studies, the feature-based image processing method has slow detection speed, low accuracy, and poor adaptability to the extreme lighting environment in the orchard.

CNN (Convolutional Neural Networks), i.e., deep learning models, can automatically extract image target features, realize automatic target recognition, and significantly improve the adaptability of model recognition [21]. The deep learning target detection method can be divided into two different types: the two-stage target detection algorithm and the single-stage target detection algorithm.

The two-stage target detection algorithm includes R-CNN (Regions with CNN features), SPP (Spatial Pyramid Pooling), Fast R-CNN (Fast Region-based Convolutional Network method), etc., and contains two network branches: RPN (Region Proposal Network) branch and the classification branch. The RPN proposes an ROI (Region of Interest) for the foreground class, while the classification branch classifies and estimates the bounding box of each ROI [22]. For instance, Wan et al. [23] proposed an improved Faster R-CNN framework for a multi-category fruit detection system. The average precision of the display category was 90.72%, and the detection time of each image was 58 ms. In addition, Chu et al. [24] developed an improved Mask R-CNN for apple detection, which integrated the suppression feature into the standard Mask R-CNN to suppress the non-apple features generated by the original network; the obtained results showed an F1-score of 0.905, along with a detection time of 0.25 s per frame. To sum up, the two-stage target detection method has a high detection accuracy, but its detection speed is slow when the model size is large, which is not conducive to implementing it into embedded devices.

As for the single-stage target detection algorithm, it includes the YOLO (You Only Look Once) series and the SSD (Single-Shot MultiBox Detector) model. Compared to the two-stage target detection method, the detection speed and accuracy are better balanced, and the overall performance of the model is improved. For example, Zhao et al. [25] detected apples, in complex environments and based on YOLOv3, with an accuracy rate of 97%, and the average detection time of a single image on the GPU device was 16.69 ms. As for Wang et al. [26], they performed channel pruning and fine-tuning on YOLOv5s; the obtained results yielded an apple detection accuracy equal to 95.8%, an F1-score of 91.5%, and an average detection time per image of 8 ms. In addition, Guan et al. [6] used YOLOv5 to detect a single frame of tomatoes, with an average recognition time of 104 ms; however, Zhou et al. [27] used YOLOv3 for strawberry ripeness classification with a mAP (Mean Average Precision) of 0.89, and the highest obtained classification AP (Average Precision) for fully ripe fruit was 0.94.

In addition, for the single-stage target detection algorithm, the introduction of the attention mechanism is beneficial to further improve the accuracy of detection. For instance, Li et al. [10] added the SE (Squeeze and Excitation) module to the YOLOv4-tiny network and proposed the YOLO-Grape algorithm. The obtained results, at the level of the category average precision mAP, showed an increase of 7.48% compared to the YOLOv4-tiny method. Ning et al. [28] added the CBAM (Convolutional Block Attention Module) method to the YOLOv4 (You Only Look Once version 4) algorithm, which led to an improvement in the accuracy of 2.21%. Adding an attention mechanism to the single-stage target detection network can effectively enhance the ability of the network to extract target features, thereby improving the detection accuracy of the model.

Dwarf and spindle- shaped tree leaves have low density. This reduces the number of apples hidden in the crown that cannot be detected. In addition, there are many apples that are blocked by leaves, apples, branches, etc., where the illumination is uneven, and the background is complex. The traditional image detection algorithm has had difficulty meeting the demand, so this paper adopts the depth learning algorithm. During fruit coloring, to obtain more evenly colored apples, some Chinese apple growers will use reflective film to cover the ground between tree rows, resulting in poor lighting conditions for image acquisition on the back-light of fruit trees. It is difficult for the naked eye to distinguish apples from the back-light side image obtained by the camera, which undoubtedly makes apple detection a great challenge for machine vision. In light of some of these situations, this study also improved the detection performance when the reflective film was covered between tree rows.

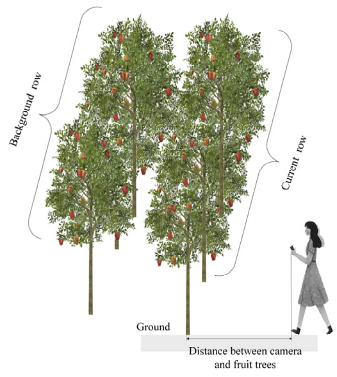
The above research does not focus on detection speed and model volume while improving detection accuracy. Therefore, this research proposes the YOLOv5-PRE (YOLOv5 Prediction) algorithm, which integrates the lightweight networks ShuffleNet V2 [29] and GhostNet [30], based on YOLOv5s, and introduces the CA (Coordinate Attention) [31] and CBAM [32] attention mechanism to improve the detection speed of the deep learning algorithm and reduce the model size.

To sum up, this paper is divided as follows: in Section 2, the data collection, processing, the methodology of work and the different algorithms will be shown. The results and the discussion will be proposed in Section 3 and finally, a conclusion summarizing the work and proposing some future ideas will be shown in Section 4.

2. Materials and Methods

2.1. Data Collection

The original dataset was collected in 2021 in Yujia Orchard, Fengxiang County, Baoji City, Shaanxi Province (N 34°35′ E 107°23′). The orchard environment is shown in Figure 1a. The fruit trees were dense and spindle-shaped, and there were several apple types, such as ‘Yuhua Fuji’, ‘Micui’, and ‘Yan Fu No. 6’. The acquisition method consisted of taking a picture from the mobile phone (SHARK PRS-A0) while being at a distance of about 2.5 m from the fruit tree, facing it, and using the wide-angle mode to include the entire fruit tree in the image. The process of taking an image is shown in Figure 1b. In clear weather, 1200 images were considered in this study (three apple varieties each accounted for one-third), all being taken in the orchard, where half consisted of the front-light images and the other half referred to the back-light images, with an image size of 1844 × 3280 pixels. The collected image contained apples that were occluded by the current tree row, its branches, and its leaves. Different levels of occlusion exist. All apples in the current tree row, except for complete occlusion, need to be detected. Background lines being detected was avoided by marking only the current line.



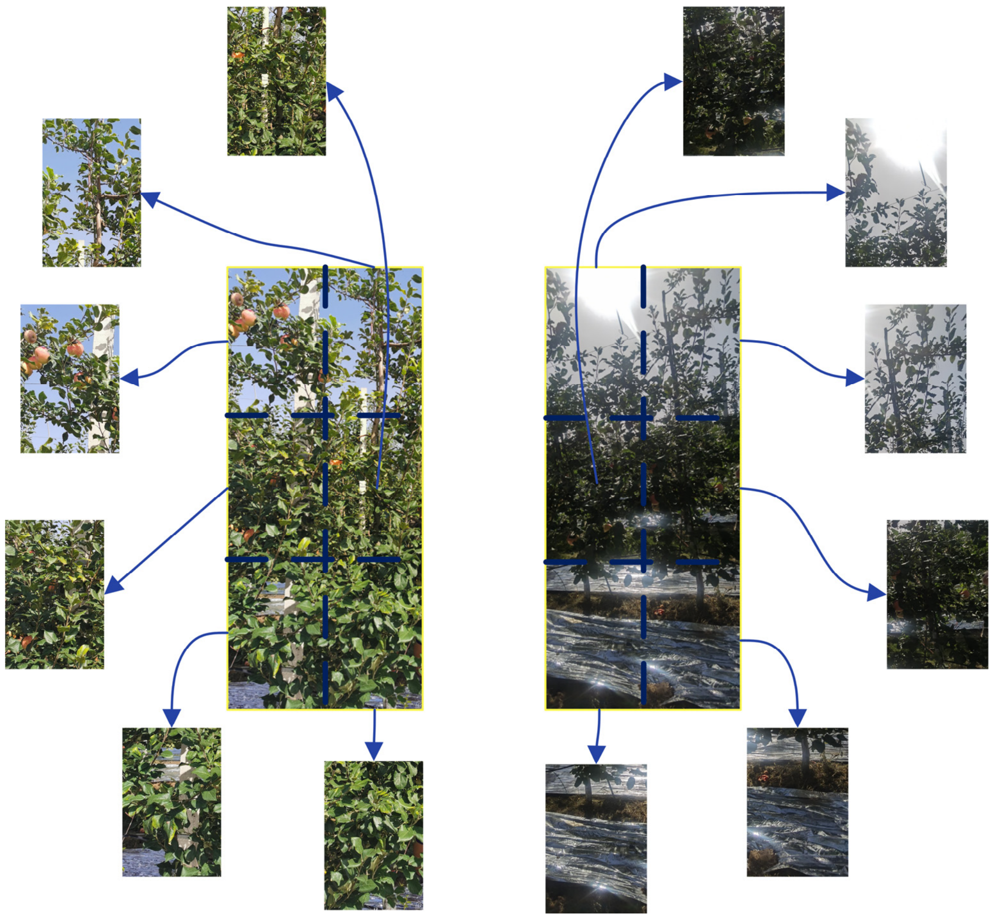
(**a**) (**b**)

Figure 1. Apple image acquisition. (**a**) Orchard environment. (**b**) Schematic diagram of image acquisition.

2.2. Data Preprocessing

In this study, both front-light (Figure 2a) and back-light images (Figure 2b) were taken throughout the day in clear weather. A selection has been performed among the acquired images to make a dataset, where the images include mainly three aspects:

- fruits that are occluded at different degrees and under different light intensities;
- motion blurring;
- fruits in background tree rows.



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

Figure 2. Sample dataset. (**a**) Data sample in front-light. (**b**) Data set sample in back-light.

One or more interference factors were included in each image to enhance the anti-interference ability of the YOLOv5-PRE algorithm model of this study.

The collected pictures (displayed in Figure 2) were cut into six equally sized pictures, and those without apples were screened out. A total of 2400 pictures were kept, where 1200 pictures were backlit and 1200 were smooth pictures. Then, the division of the images into the dataset and a testing set with a ratio of 9 to 1 was performed; thus, among every 10 pictures, 9 were saved in the dataset and the last one was maintained just for testing purposes. The last phase consisted of using Makesense AI to manually mark the detection target and select the minimum circumscribed rectangle of the apple to specify the target apple and reduce the background pixels.

2.3. YOLOv5 Algorithm

YOLOv5 is an improved network of YOLO series, based on YOLOv4, and its performance has been greatly improved. YOLOv5 is divided into four types: YOLOv5s, yolov5m, yolov5l, and yolov5x, according to the size of the network model. In this study, YOLOv5s model was used to achieve the aim of detecting target apples in complex orchard environments. The YOLOv5s network structure consists of an input, a backbone network, and a neck. The model structure is displayed in Figure 3. The different parts of the network structure are presented below:

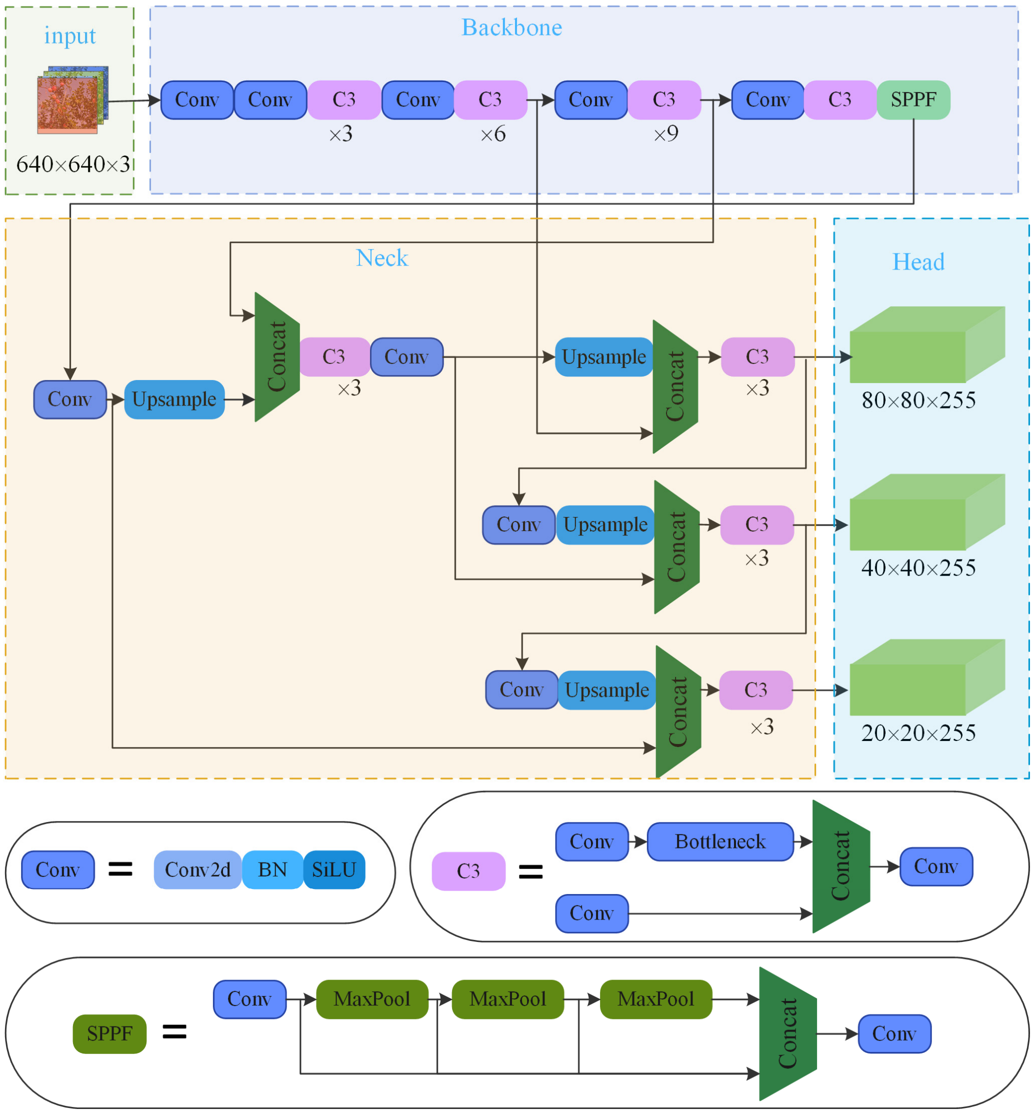


Figure 3. The architecture of the YOLOv5s method.

(1) The backbone network part mainly included Conv (Standard Convolutional Layer), C3 (CSP Bottleneck with 3 Convolutions) and SPPF (Fast Spatial Pyramid Pooling) structures. In more detail, the Conv structure sequentially performs conv_2d (Two-Dimensional Convolution), BN (Batch Normalization), and SiLU (Sigmoid-Weighted Linear Unit) activation function operations on the input feature map. The C3 module contains three Conv structures and multiple Bottleneck modules for the extraction of residual features. After the SPPF module passes the input through the Conv operation, it performs multiple MaxPool (Maximum Pooling) operations with a convolution kernel of 5×5, which is used to extract and to aggregate the image features, and standardize the size of the feature images;

(2) The neck network uses the feature pyramid structure of the FPN (Feature Pyramid Network) and the PAN (Path Aggregation Network) to generate feature pyramids to adapt to target detection at different scales;

(3) The main body of the detection head part consists of three detectors, with three feature maps of different scales that are used to detect targets of different sizes.

Due to the presence of several layers and the different parameters of YOLOv5, no attention preference is reserved in the feature extraction process. In addition, the training time is long, the processor performance requirements are high, and the model size is large. All of these reasons mean that it is considered not conducive for being transplanted into embedded devices.

2.4. Improved Apple Detection Model

To improve the detection accuracy and the real-time detection speed, this study used ShuffleNet and GhostNet networks to improve the backbone and the neck networks of YOLOv5 and introduced CA and CBAM attention mechanisms to realize the lightweight design of the network model; thus, a YOLOv5-PRE was proposed. The apple detection model and the network structure are shown in Figure 4.

YOLOv5-PRE removed all C3 modules in the backbone network, used lightweight SHUF (Shuffle) and GhostConv modules to replace the Conv modules except for the zeroth layer, and added CA and CBAM attention modules. In the bottleneck network, S_C (Shuffle_Conv) and C3Ghost were used to replace all Conv and C3 modules, respectively. The YOLOv5s-SG (YOLOv5s ShuffleNet GhostNet) model is a smaller version of the YOLOv5-PRE model that solely provided lightweight enhancements by eliminating the attention mechanisms CA and CBAM. YOLOv5-Shuffle only deleted all C3 modules in the trunk network, and used SHUF to replace Conv modules except for the zeroth layer, which is an intermediate product of the model change process. Below, the different stages of the apple detection model based on the lightweight improved YOLOv5s are listed:

(1) The SHUF module is a unit with a step size of 2 in ShuffleNet V2, which was used to replace some of the Conv modules, originally located in the backbone network. ShuffleNet V2, which is an efficient and lightweight network, is designed according to four principles. While keeping the same number of input and output channels, this method reduces the network branches and does not use them much. The convolution is applied without over-grouping and while reducing element-wise operations. As shown in Figure 4, the SHUF module uses channel splitting, which can significantly reduce the number of parameters. However, channel splitting blocks the information flow between channel groups and reduces the channel information expression ability. To this end, a Channel-Shuffle (Channel Shuffling Mechanism) is designed to enhance the information association between channels;

(2) The S_C module borrowed the original GhostConv module and replaced the Conv in GhostConv with the SHUF module. GhostConv divides the input features into two branches to perform the convolution operations with the Conv module, and uses Concat to splice the features and output them. The GhostConv module generates more feature maps through simple operations. Based on a set of intrinsic feature maps, a series of linear transformations are used to generate many pseudo-feature maps that can fully reveal the information behind the intrinsic features. Based on the above, the two branches of the S_C

module respectively process the input through the SHUF module and then splicing and outputting it. Through the channel splitting and shuffling mechanism in the SHUF module, the consumption of parameters and computation of resources is further reduced, compared to GhostConv method;

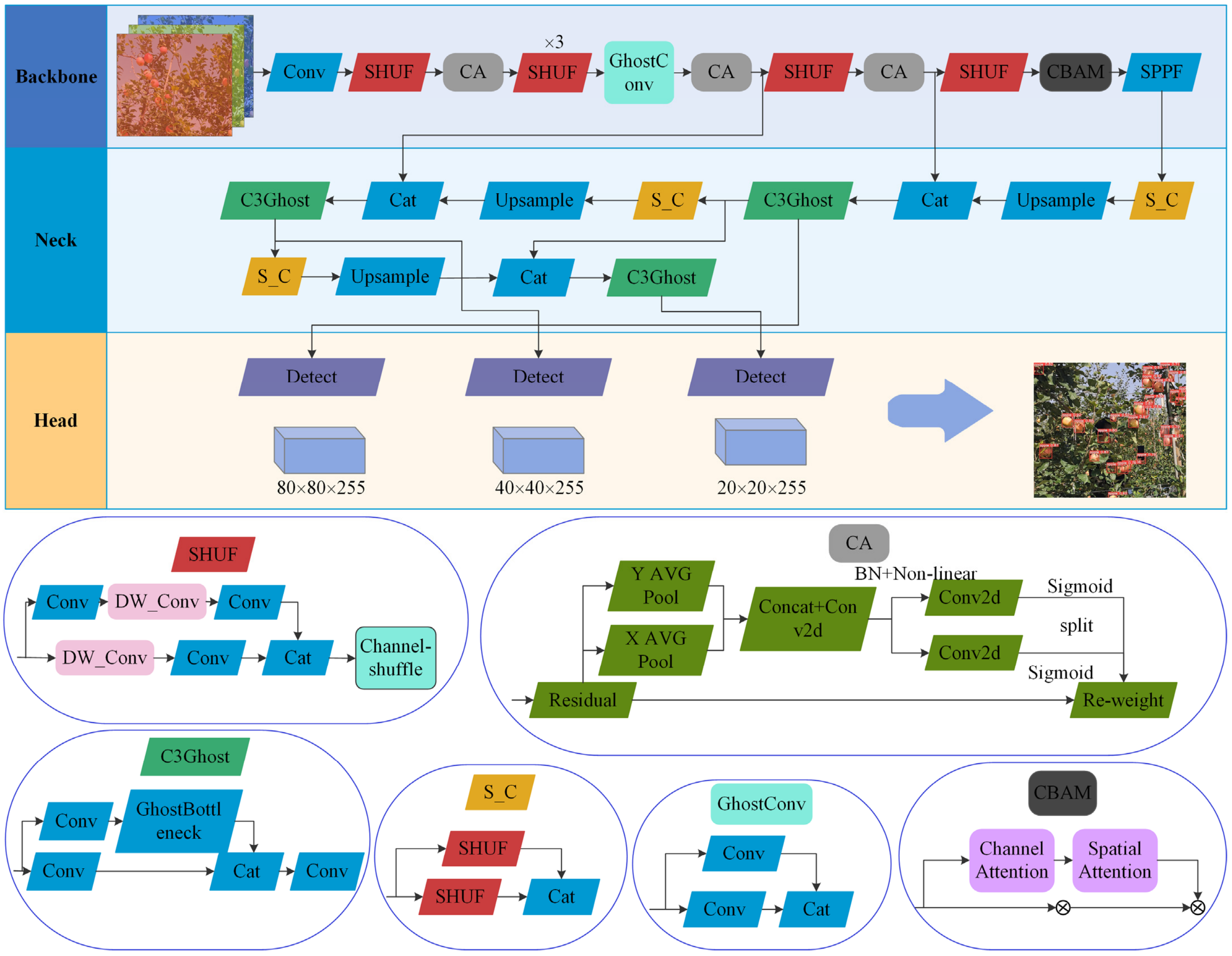


Figure 4. The architecture of the YOLOv5-PRE method.

(3) The C3Ghost module is a fusion structure of the C3 module and the Ghost Conv module. The input is sent via one branch's conventional convolution unit, while the input of the other branch is successively passed through procedures, such as depth-wise separable convolution, GhostConv convolution, and GhostConv convolution. The branch's outcomes are combined and output;

(4) CA first divides the input feature map into width and height directions for global average pooling, and obtains the feature map regarding these two directions. The feature maps in the width and height directions of the global receptive field are spliced together, and then the features are sent to the convolution module with a convolution kernel of 1×1; thus, the input features are obtained after batch normalization and Sigmoid activation function. The graph's attention weighs in the height and width directions. Finally, through the multiplication and the weighting calculation of the original feature map, the final feature map with attention weights in the width and height directions is obtained.

(5) The core idea of CBAM is to enhance the weight of the target features through the Channel_Attention and Spatial_Attention modules to achieve the effect of enhancing the detection accuracy.

The addition of the attention mechanism adopted two strategies; one is to place only one layer of attention module before the fast spatial pyramid pooling module of the backbone network; the other is to put it in each layer of the backbone network after the convolution module (SHUF, Group Conv).

3. Results and Discussion

3.1. Model Training

The workstation configuration parameters used in the test platform of this study are shown in Table 1. The training process used adaptive anchor boxes to detect targets of different sizes. The initial training parameters were with the learning rate set to 0.01, the momentum factor equal to 0.937, the number of iterations equal to 300, and the batch-size set to 32.

Table 1. Configuration parameters of experimental platform.

Parameters	Dispose
CPU	Intel (R) Xeon (R) Silver 4214 CPU @2.20 GHz 2.19 GHz
GPU	NVIDIA Quadro P620
Programing Language	Python 3.9
System	Windows 10
Software Environment	Cuda 10.2

3.2. Evaluation Indicators

This study used the AP (Average Precision), the P (Precision Rate), the R (Recall Rate), the inference time on GPU devices, the inference time on CPU devices, and the model volume as evaluation indicators.

The *P* value represents the proportion of all predicted positive targets that are actually positive, and it is defined as follows:

$$P = \frac{T_P}{T_P + F_P} \tag{1}$$

The recall rate is the proportion of predicted positive samples among all targets that are actually positive samples. This value can be defined as follow:

$$R = \frac{T_P}{T_P + F_N} \tag{2}$$

In Equations (1) and (2), $\mathrm{T_P}$ is the total number of target boxes for correctly identifying apples, whereas $\mathrm{F_P}$ is the total number of target boxes for wrongly identifying apples, and $\mathrm{F_N}$ is the total number of missed detections for apples.

In Equation (3), the average precision represents the area enclosed by the PR curve drawn by the precision rate and the recall rate, and the coordinate axis. This area shows the precision value of a single type of target:

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

3.3. Network Structure Ablation Experiment

To explore the effectiveness of this study upon the improvement of YOLOv5s and its impact on the detection rate, an ablation experiment was mounted on the YOLOv5-PRE network structure. The different backbone network structure models are shown in Table 2. The neck network and the detection head of YOLOv5s-Shuffle are similar to the YOLOv5s algorithm. The neck network of YOLOv5-1, YOLOv5-2, YOLOv5-3, YOLOv5-4, YOLOv5-5, and YOLOv5-6 used the GhostConv module and C3 ghost module to replace Conv module and C3 module, respectively, and the detection head is similar to that of

YOLOv5s. YOLOv5-PRE and YOLOv5s-Shuffle-Ghost used the S_C module and C3ghost module, respectively, to replace the Conv module and C3 module in the neck network, and the detection head is similar to YOLOv5s.

Table 2. Algorithm Backbone structure.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
YOLOv5s-Shuffle	Conv	SHUF	SHUF	GhostConv	SHUF	SHUF	SPPF	-	-	-	-
YOLOv5-1	Conv	SHUF	SHUF	GhostConv	SHUF	SHUF	SPPF	-	-	-	-
YOLOv5-2	Conv	SHUF	CA	SHUF	GhostConv	SHUF	SHUF	SPPF	-	-	-
YOLOv5-3	Conv	SHUF	CA	SHUF	GhostConv	CA	SHUF	CA	SHUF	SPPF	-
YOLOv5-4	Conv	SHUF	CA	SHUF	GhostConv	CA	SHUF	CA	SHUF	CA	SPPF
YOLOv5-5	Conv	SHUF	CA	SHUF	GhostConv	CA	SHUF	CA	SHUF	CBAM	SPPF
YOLOv5-6	Conv	SHUF	CA	SHUF	GhostConv	CA	SHUF	CBAM	SHUF	CBAM	SPPF
YOLOv5-PRE	Conv	SHUF	CA	SHUF	GhostConv	CA	SHUF	CA	SHUF	CBAM	SPPF
YOLOv5s-SG	Conv	SHUF	SHUF	GhostConv	SHUF	SHUF	SPPF	-	-	-	-

Training is shown in Table 3, where the YOLOv5s-Shuffle algorithm had a large volume and low average precision. Due to the improvement of the neck network by YOLOv5-1 on the basis of YOLOv5s-Shuffle, the algorithm volume was greatly reduced, resulting in a decrease in the average precision. Based on YOLOv5-1, YOLOv5-2 added a layer of CA module after the SHUF module of the backbone network. The algorithm size was only increased by 9 KB, and the detection accuracy was improved by 0.02%. As for YOLOv5-3, it added a three-layer CA module to the backbone network of YOLOv5-1, and the average precision increased by 0.31%. To further improve the detection accuracy, a layer of CA module was added before the SPPF of the backbone network to obtain YOLOv5-4; however, the detection accuracy of the model decreased. Thus, a layer of CBAM attention module was added based on the YOLOv5-3 algorithm. In addition, the detection accuracy of the algorithm YOLOv5-5 had been further improved. Replacing one of the CA attention modules of YOLOv5-5 with a CBAM module yielded YOLOv5-6, which greatly reduced the detection accuracy of the YOLOv5-5 algorithm. Therefore, based on YOLOv5-5, the GhostConv module in the neck network of YOLOv5-5 was replaced by the S_C module that was proposed in this study, and the YOLOv5-PRE model was obtained. Based on YOLOv5-5, the algorithm volume was further reduced, while the average AP of the algorithm was greatly improved. All the attention modules in YOLOv5-PRE algorithm were removed, and the algorithm YOLOv5s-SG was obtained. The model volume of YOLOv5s-SG is the smallest among all the algorithms in this study.

Table 3. Performance of ablation experiment algorithm.

	YOLOv5s-Shuffle	**YOLOv5-1**	**YOLOv5-2**	**YOLOv5-3**	**YOLOv5-4**	**YOLOv5-5**	**YOLOv5-6**	**YOLOv5s-SG**	**YOLOv5-PRE**
Model Volume/KB	11,531	4955	4964	5040	5443	5060	5061	4695	4800
AP	93.09%	92.54%	92.56%	92.85%	91.16%	93.46%	91.92	93.34%	94.03%

3.4. Comparative Test

During the training process, YOLOv5-Shuffle, YOLOv5s-SG, YOLOv5-PRE, YOLOv5s-Ghost models, and YOLOv5s, converged when considering 300 epochs. The average precision curves of the five models are shown in Figure 5, where they converged quickly. YOLOv5s has a faster growth rate than YOLOv5-PRE AP, but its training curve oscillates more in the first 50 rounds. In the last 100 rounds, the average precision of the YOLOv5-PRE algorithm is slightly higher than that of YOLOv5s, and the final training accuracy of the YOLOv5-PRE algorithm is highest, and the accuracy of the other algorithms is lower than that of YOLOv5s.

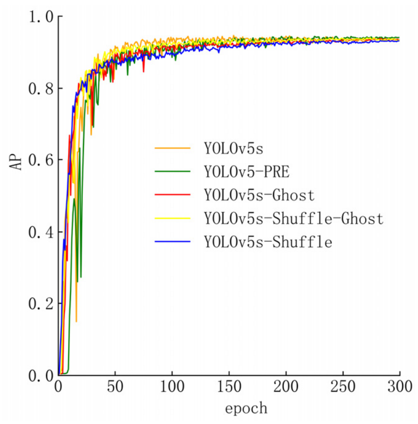


Figure 5. AP curve of YOLOv5-PRE and YOLOv5s models.

To verify the effect of the model size and the average precision of YOLOv5-PRE, the training results of the research algorithms YOLOv5-Shuffle, YOLOv5s-SG, YOLOv5-PRE, YOLOv5s and YOLOv5s-Ghost, under the same training set and test set, are shown in Table 4.

Table 4. Comparison of algorithm performance.

Method	Model Volume (kB)	Reasoning Time GPU (ms)	Reasoning Time CPU (ms)	P	R	AP
YOLOv5s	14,001	32.9	266.0	88.00%	89.63%	93.45%
YOLOv5s-Ghost	7555	31.2	266.2	87.81%	89.08%	93.41%
	−46.04%	−5.10%	0.05%	−0.19%	−0.55%	−0.04%
YOLOv5s-Shuffle	11,531	30.2	166.4	90.47	86.00	93.09%
	−17.64%	−8.20%	−37.45%	+2.47%	−3.63%	−0.36%
YOLOv5s-SG	4695	26.9	153.3	87.81%	88.96%	93.34%
	−66.47%	−18.24%	−42.37%	−0.19%	−0.67%	−0.11%
YOLOv5-PRE	4800	27.0	172.3	88.68	89.08	94.03%
	−65.72%	−17.93%	−35.23%	+0.68%	−0.55%	+0.58%

YOLOv5s-Ghost reduced the algorithm volume by 46.04%, while the accuracy was only reduced by 0.04%; however, the algorithm inference speed was less improved on both GPU and CPU devices.

Data in Table 4 shows that the YOLOv5-Shuffle algorithm has a smaller volume and a faster execution speed than the YOLOv5s model; however, compared to the performance of the YOLOv5-Ghost algorithm, the algorithm's inference speed and accuracy are poor.

The YOLOv5s-SG algorithm obtained after lightning was 66.47%, which is smaller than the YOLOv5 model, and the inference speed was increased by 18.24% and 42.37% under GPU and CPU devices, respectively. As for the AP, it was reduced by 0.11%. Compared to YOLOv5 and YOLOv5-Ghost, the YOLOv5s-SG algorithm has a relatively larger reduction in the inference speed and the model size.

The YOLOv5-PRE algorithm with attention mechanism has improved the model accuracy obtained by YOLOv5s-SG, and the inference time under GPU and CPU devices have increased by 0.31% and 7.14%, respectively. Compared to the YOLOv5s algorithm, the volume of the YOLOv5-PRE algorithm is reduced by 65.72%, the AP has increased by 0.58%, and the precision rate has increased by 0.68%, but the recall rate has decreased. The detection speed has increased by 17.93% under the GPU device, and by 35.23% under the CPU device.

3.5. Comparison of Detection Effects

YOLOv5-PRE and YOLOv5s were used for the final weight model, after 300 rounds of training, to detect the front-light and back-light environment where the input image size was set to 600 × 600, the IoU (Intersection over Union) parameter was set to 0.2, the confidence level parameter was set to 0.4, the number of target statistics was placed in the upper left corner of the image; the comparison of detection effects is shown in Figure 6.

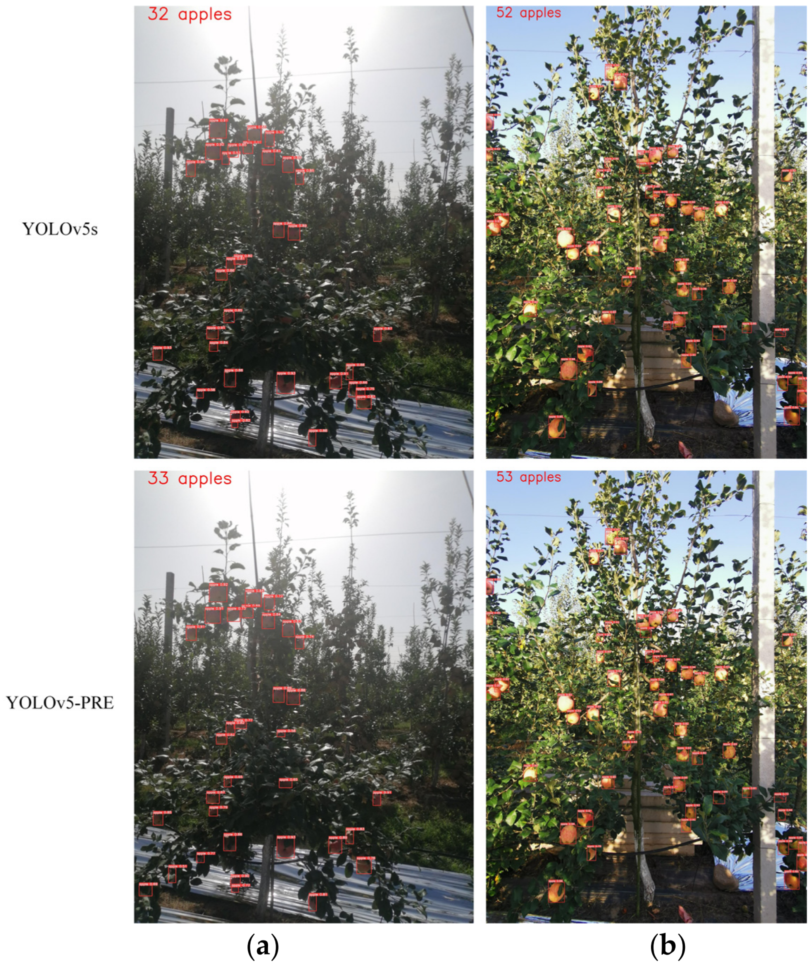


Figure 6. Comparison of recognition effect between YOLOv5-PRE and YOLOv5s. (**a**) Example of detection of fruit trees under front-light. (**b**) Example of detection of fruit trees under back-light.

As presented in Table 5, YOLOv5s had a total of 118 targets in the front-light environment, 4 missed detections and 6 false detections. As for the back-light environment target, they were 99, the number of missed detections was 10, and the number of false detections was 2.

Table 5. Comparison of missed and false detection between YOLOv5-PRE and YOLOv5s.

	Front-Light			Back-Light		
Method	**Undetected**	**False Check**	**Detected Number/Actual Number**	**Undetected**	**False Check**	**Detected Number/Actual Number**
YOLOv5-PRE	4	5	117/116	7	2	102/107
YOLOv5s	4	6	118/116	10	2	99/107

YOLOv5-PRE had a total of 117 targets in the forward light environment, 4 missed detections and 5 false positives. As for the back-light environment target, they were 102, the number of missed detections was 7, and the number of false positive detections was 2. Because the YOLOv5-PRE model has higher detection accuracy and higher probability values for the same target, compared with YOLOv5s, YOLOv5-PRE has fewer missed and false detections in a back-light environment. Thus, YOLOv5-PRE algorithm adapts well to different lighting conditions, and can detect fruits that are occluded by other fruits, branches, and leaves.

3.6. Feature Map Visualization of the Model

Since the deep learning target detection model is difficult to decompose into a single intuitive component, it is challenging to explain the link between the algorithm performance and the algorithm structure, only based on the input and output and model topology. To deeply analyze the segmentation process and the working principle of the YOLOv5-PRE algorithm, this study extracted the feature maps of the backbone network and the neck network, converted them into heat maps, superimposed the original images, and displayed them visually to intuitively display the YOLOv5-PRE algorithm. The feature map of the backbone network is shown in Figure 7.

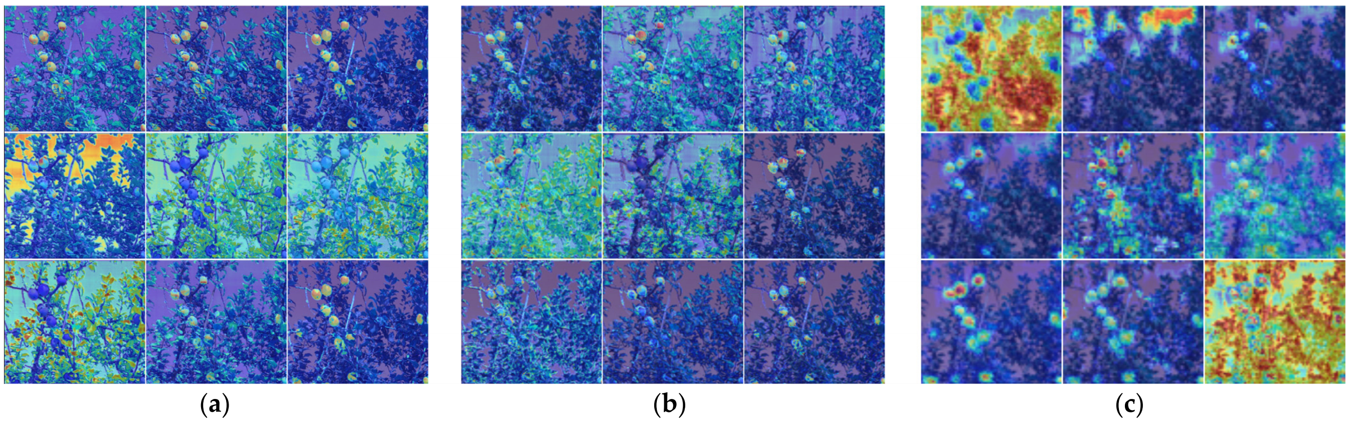


Figure 7. Backbone feature extraction network feature map. (**a**) CA output 80 × 80 characteristic figure. (**b**) CA output 40 × 40 characteristic figure. (**c**) SPPF output 20 × 20 characteristic figure.

Referring to Figure 7a,b, the shallow effect in the backbone network (the fifth and seventh layers of CA modules) carried out detailed feature extraction related to apples, leaves, tree trunks, and sky. The focus of the detection targets lying in contour, color, texture, and image segmentation was very clear. The feature map delivered by the SPPF module extracted the features of different categories of objects in the picture more accurately, but due to the small size and the reduced number of pixels in the picture, the extracted features were more blurred than those of the shallow network. It can be seen from Figure 7c that, when extracting the overall features of apples, this part of the output feature extraction tended to pay attention to the unobstructed and darker parts of the target. Apart from the extraction of the overall features of the apple, the extraction of the contour features of the apple was also heavily weighted. Through contour feature extraction, the calculation amount of the feature extraction of the entire image can be reduced while ensuring the accuracy of the target detection.

The backbone network sent the extracted features to the neck network for further feature extraction and fusion, and the obtained feature map heatmap is shown in Figure 8.

Referring to Figure 8a, although the 18th layer of the network still considered the texture and the color of objects, such as apples, leaves, and sky, the region of interest had been narrowed down and was now focused on a particular location. Starting from the network layer 21 output (Figure 8b), the network was more inclined to focus on the object's contours. The output of layer 24 of the network (Figure 8c) focused more on object outlines and background detection. Since the apple's shape differs significantly from that of other background objects, paying close attention to it makes background filtering and removal easier.

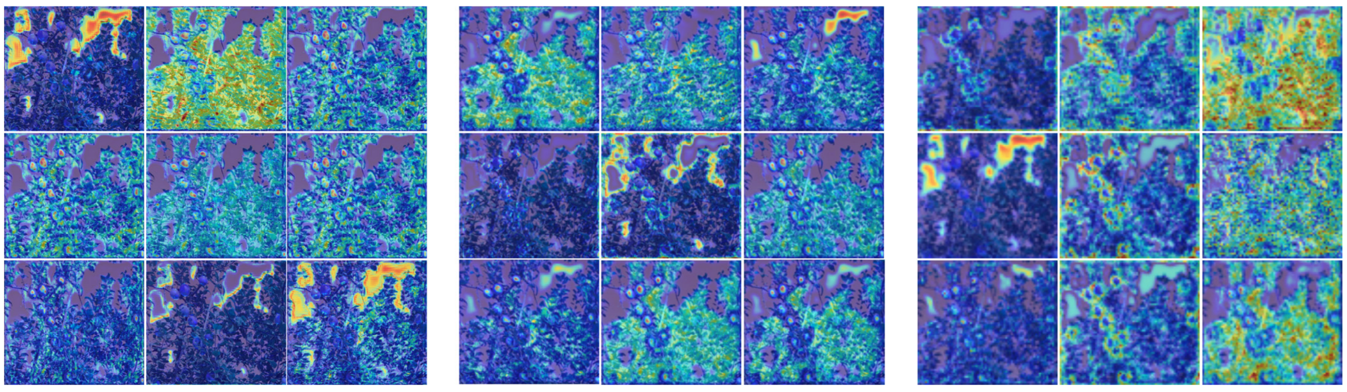


Figure 8. Neck network feature map. (**a**). C3ghost output 80 × 80 characteristic figure. (**b**). C3ghost output 40 × 40 characteristic figure. (**c**). C3ghost output 20 × 20 characteristic figure.

4. Conclusions

Incorporating CA and CBAM attention techniques, ShuffleNet and GhostNet are lightweight networks built on YOLOv5s. A theoretical foundation for the continued use of deep learning algorithms in embedded devices is provided by the detection algorithm YOLOv5-PRE of the apple, which is appropriate for complex environments and significantly reduces the model size, speeds up detection and reasoning, and achieves better detection accuracy. The main conclusions of the study are:

(1) The average precision AP of the YOLOv5-PRE algorithm was 94.03%, which is 0.58% higher than that of YOLOv5s. While the model volume is reduced by 65.72%, the detection speed is increased by 17.93% on GPU devices and 35.23% on CPU devices.

(2) The YOLOv5-PRE algorithm has more obvious advantages in improving the detection speed in non-GPU devices, and is more suitable for running in embedded devices with poor performance. To improve the detection speed while lowering the requirements for detection accuracy, the YOLOv5s-SG algorithm, without the attention mechanism, can be selected; it is 42.37% faster than YOLOV5s on CPU devices, while the accuracy is only reduced by 0.11%.

(3) This study uses a visual feature heatmap to visually display the feature extraction process of the YOLOv5-PRE algorithm. The shallow feature network, in the backbone network part, extracts more image details due to the smaller receptive field. In the neck network, more attention is paid to the background and the outline of the object, which is conducive through the use of a lower computing power and a faster real-time detection speed to achieve higher detection accuracy. The visual feature heatmap is displayed graphically in the middle part of the model, as it improves its interpretability.

The YOLOv5-PRE algorithm proposed in this study provides a reference for the rapid detection of apple yield to detect apples in the orchard environment in real-time. It has broad application prospects in smart agriculture. Using the algorithm in this paper to generate a yield map in the process of yield detection, can show the different growth conditions of each tree, which can provide a reference for fruit tree cultivation in the next year. In the future, we will focus on improving the detection accuracy of the algorithm, while preserving the existing detection speed.

Author Contributions: Conceptualization, L.S. and J.C.; methodology, L.S.; software, L.S.; validation, L.S., J.C., G.H. and C.C.; investigation, L.S., J.C., G.H., H.C., C.L., S.Z. and C.C.; writing—original draft preparation, L.S.; writing—review and editing, L.S., J.C. and G.H.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 32272001.

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

Abbreviations

Abbreviation	Meaning
YOLOv5	You Only Look Once version 5
YOLOv5-PRE	YOLOv5 Prediction
CA	Coordinate Attention
CBAM	Convolutional Block Attention Module
YOLOv5s	You Only Look Once version 5 small
SVM	Support Vector Machine
CNN	Convolutional Neural Networks
R-CNN	Regions with CNN features
SPP	Spatial Pyramid Pooling
Fast R-CNN	Fast Region-based Convolutional Network method
RPN	Region Proposal Network
ROI	Region of Interest
YOLO	You Only Look Once
SSD	Single-Shot MultiBox Detector
mAP	Mean Average Precision
SE	Squeeze-and-Excitation
YOLOv4	You Only Look Once version 4
Conv	Standard Convolutional Layer
C3	CSP bottleneck with 3 convolutions
SPPF	fast spatial pyramid pooling
Conv_2d	Two-Dimensional Convolution
BN	Batch Normalization
SiLU	Sigmoid-Weighted Linear Unit
MaxPool	Maximum Pooling
FPN	Feature Pyramid Network
PAN	Path Aggregation Network
SHUF	Shuffle
S_C	Shuffle_Conv
Channel-Shuffle	Channel Shuffling Mechanism
AP	Average Precision
P	Precision Rate
R	Recall Rate
YOLOv5s-SG	YOLOv5s ShuffleNet GhostNet
IoU	Intersection over Union
HOG and LBP	Oriented Gradient and Local Binary Pattern
ANN-GDX	Artificial Neural Network, Variable Learning Rate Backpropagation

References

1. Gao, F.; Wu, Z.; Suo, R.; Zhou, Z.; Li, R.; Fu, L.; Zhang, Z. Apple detection and counting using real-time video based on deep learning and object tracking. *Trans. Chin. Soc. Agric. Eng. Trans. CSAE* **2021**, *37*, 217–224. [CrossRef]
2. He, L.; Fang, W.; Zhao, G.; Wu, Z.; Fu, L.; Li, R.; Majeed, Y.; Dhupia, J. Fruit yield prediction and estimation in orchards: A state-of-the-art comprehensive review for both direct and indirect methods. *Comput. Electron. Agric.* **2022**, *195*, 106812. [CrossRef]
3. Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* **2020**, *197*, 245–256. [CrossRef]
4. Xu, P.; Fang, N.; Liu, N.; Lin, F.; Yang, S.; Ning, J. Visual recognition of cherry tomatoes in plant factory based on improved deep instance segmentation. *Comput. Electron. Agric.* **2022**, *197*, 106991. [CrossRef]
5. Qi, J.; Liu, X.; Liu, K.; Xu, F.; Guo, H.; Tian, X.; Li, M.; Bao, Z.; Li, Y. An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. *Comput. Electron. Agric.* **2022**, *194*, 106780. [CrossRef]
6. Guan, Z.; Li, H.; Zuo, Z.; Pan, L. Design a robot system for tomato picking based on YOLOv5. *IFAC-Pap. OnLine* **2022**, *55*, 166–171. [CrossRef]
7. Jian, Y.; Zhen, Q.; Yanjun, Z.; Qin, Y.; Liao, H. Real-time recognition of tomatoes in complex environments based on improved YOLOv4-tiny. *Trans. Chin. Soc. Agric. Eng. Trans. CSAE* **2022**, *38*, 215–221. [CrossRef]
8. Ning, Z.; Luo, L.; Liao, J.; Wen, H.; Wei, H.; Lu, Q. Recognition and the optimal picking point location of grape stems based on deep learning. *Trans. Chin. Soc. Agric. Eng. Trans. CSAE* **2021**, *37*, 222–229. [CrossRef]

9. Pérez-Zavala, R.; Torres-Torriti, M.; Cheein, F.A.; Troni, G. A pattern recognition strategy for visual grape bunch detection in vineyards. *Comput. Electron. Agric.* **2018**, *151*, 136–149. [CrossRef]
10. Li, H.; Li, C.; Li, G.; Chen, L. A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. *Biosyst. Eng.* **2021**, *212*, 347–359. [CrossRef]
11. Jin, Y.; Liu, J.; Wang, J.; Xu, Z.; Yuan, Y. Far-near combined positioning of picking-point based on depth data features for horizontal-trellis cultivated grape. *Comput. Electron. Agric.* **2022**, *194*, 106791. [CrossRef]
12. Olenskyj, A.; Sams, B.; Fei, Z. End-to-end deep learning for directly estimating grape yield from ground-based imagery. *Comput. Electron. Agric.* **2022**, *198*, 107081. [CrossRef]
13. Yu, L.; Xiong, J.; Fang, X.; Yang, Z.; Chen, Y.; Lin, X.; Chen, S. A litchi fruit recognition method in a natural environment using RGB-D images. *Biosyst. Eng.* **2021**, *204*, 50–63. [CrossRef]
14. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A visual detection method for nighttime litchi fruits and fruiting stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [CrossRef]
15. Osako, Y.; Yamane, H.; Lin, S.-Y.; Chen, P.-A.; Tao, R. Cultivar discrimination of litchi fruit images using deep learning. *Sci. Hortic.* **2020**, *269*, 109360. [CrossRef]
16. Sun, S.; Jiang, M.; He, D.; Long, Y.; Song, H. Recognition of green apples in an orchard environment by combining the GrabCut model and Ncut algorithm. *Biosyst. Eng.* **2019**, *187*, 201–213. [CrossRef]
17. Wang, D.; He, D. Fusion of Mask RCNN and attention mechanism for instance segmentation of apples under complex background. *Comput. Electron. Agric.* **2022**, *196*, 106864. [CrossRef]
18. Zhao, H.; Qiao, Y.; Wang, H.; Yue, Y. Apple fruit recognition in complex orchard environment based on improved YOLOv3. *Trans. Chin. Soc. Agric. Eng. Trans. CSAE* **2021**, *37*, 127–135. [CrossRef]
19. Syazwani, R.; Asraf, H.; Amin, M.; Dalila, N. Automated image identification, detection and fruit counting of top-view pineapple crown using machine learning. *Alex. Eng. J.* **2022**, *61*, 1265–1276. [CrossRef]
20. Fu, L.; Duan, J.; Zou, X.; Lin, G.; Song, S.; Ji, B.; Yang, Z. Banana detection based on color and texture features in the natural environment. *Comput. Electron. Agric.* **2019**, *167*, 105057. [CrossRef]
21. Lalitha, V.; Latha, B. A review on remote sensing imagery augmentation using deep learning. *Mater. Today: Proc.* **2022**, *62*, 4772–4778. [CrossRef]
22. Lu, Z.; Zhao, M.; Luo, J.; Wang, G.; Wang, D. Design of a winter-jujube grading robot based on machine vision. *Comput. Electron. Agric.* **2021**, *186*, 106170. [CrossRef]
23. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [CrossRef]
24. Chu, P.; Li, Z.; Lammers, K.; Lu, R.; Liu, X. Deep learning-based apple detection using a suppression mask R-CNN. *Pattern Recognit. Lett.* **2021**, *147*, 206–211. [CrossRef]
25. Dean, Z.; Rendi, W.; Xiaoyang, L.; Yuyan, Z. Apple positioning based on YOLO deep convolutional neural network for picking robot in complex background. *Trans. Chin. Soc. Agric. Eng. Trans. CSAE* **2019**, *35*, 164–173. [CrossRef]
26. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
27. Zhou, X.; Lee, W.; Ampatzidis, Y.; Chen, Y.; Peres, N.; Fraisse, C. Strawberry maturity classification from UAV and near-ground imaging using deep learning. *Smart Agric. Technol.* **2021**, *1*, 100001. [CrossRef]
28. Ning, Z.; Luo, L.; Ding, X.; Dong, Z.; Yang, B.; Cai, J.; Chen, W.; Lu, Q. Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards. *Comput. Electron. Agric.* **2022**, *196*, 106878. [CrossRef]
29. Ma, N.; Zhang, X.; Zheng, H.; Sun, J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. *arXiv* **2018**, arXiv:1807.11164. [CrossRef]
30. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. *arXiv* **2020**, arXiv:1911.11907. [CrossRef]
31. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. *arXiv* **2021**, arXiv:2103.02907. [CrossRef]
32. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional block attention module. *arXiv* **2018**, arXiv:1807.06521. [CrossRef]