Journal of Real-Time Image Processing (2024) 21:164
https://doi.org/10.1007/s11554-024-01543-4

RESEARCH

YOLOv8s-CFB: a lightweight method for real-time detection of apple fruits in complex environments

Bing Zhao[1] · Aoran Guo[2] · Ruitao Ma[1] · Yanfei Zhang[2] · Jinliang Gong[1]

Received: 21 June 2024 / Accepted: 17 August 2024

Abstract

With the development of apple-picking robots, deep learning models have become essential in apple detection. However, current detection models are often disrupted by complex backgrounds, leading to low recognition accuracy and slow speeds in natural environments. To address these issues, this study proposes an improved model, YOLOv8s-CFB, based on YOLOv8s. This model introduces partial convolution (PConv) in the backbone network, enhances the C2f module, and forms a new architecture, CSPPC, to reduce computational complexity and improve speed. Additionally, FocalModulation technology replaces the original SPPF module to enhance the model's ability to recognize key areas. Finally, the bidirectional feature pyramid (BiFPN) is introduced to adaptively learn the importance of weights at each scale, effectively retaining multi-scale information through a bidirectional context information transmission mechanism, and improving the model's detection ability for occluded targets. Test results show that the improved YOLOv8 network achieves better detection performance, with an average accuracy of 93.86%, a parameter volume of 8.83 M, and a detection time of 0.7 ms. The improved algorithm achieves high detection accuracy with a small weight file, making it suitable for deployment on mobile devices. Therefore, the improved model can efficiently and accurately detect apples in complex orchard environments in real time.

Keywords Apple detection · YOLOv8s · PConv · BiFPN · Embedded systems

1 Introduction

The annual global apple output is approximately 8,433,057 tons, with the largest producers being China, the United States, Poland, and Turkey. This output continues to increase year by year [1]. Currently, apple harvesting heavily relies on manual labor, a process that is time-consuming, labor-intensive, and inefficient [2]. Apple-picking robots are specially engineered devices designed to automate the harvesting process [3], effectively tackling challenges posed by labor shortages and inefficiencies in manual picking, ultimately leading to cost reductions [4]. These robots are equipped with mobility devices, robotic arms, end effectors, vision systems, and control systems [5]. The vision system plays a pivotal role in enabling apple-picking robots to autonomously harvest apples. Therefore, quick and precise apple detection is crucial for advancing automated apple picking and achieving intelligent agricultural production [6].

Computer vision is an advanced and objective detection technology. The development of multi-camera combined imaging systems further enables computer vision technology to meet target accuracy and quality requirements [7]. In recent years, significant progress has been made in this area of research. Gao et al. [8] proposed an apple detection system and showed that the system can achieve robotic apple picking performance. Yoshida et al. [9] used RGB-D sensors to detect and locate fruits, improving the positioning accuracy of robot picking. Furthermore, Linker et al. [10] used color and texture information to classify green apples. By comparing detection circles with heuristic models, it can be concluded that color texture affects the results.

With advancements in computer technology, convolutional neural networks and deep learning-based object detection models have been widely applied across various industries, including fruit recognition, disease detection, and yield estimation in agriculture [11]. Currently, there are two main types of deep learning-based fruit detection models. The

✉ Jinliang Gong
84374294@qq.com

[1] School of Mechanical Engineering, Shandong University of Technology, Zibo 255049, China

[2] School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo 255049, China

Published online: 31 August 2024

first type uses region proposal methods, which generate candidate bounding boxes through box generation techniques. These candidate boxes are then classified and localized using convolutional neural networks such as Faster R-CNN [12], R-FCN [13], Mask R-CNN [14], etc. These models achieve high accuracy but require large computational resources and are slower in terms of speed. Zhu et al. [15], for example, used the Faster R-CNN model to recognize blueberries at different stages of ripeness, achieving recognition accuracies of 97% for ripe fruits, 95% for semi-ripe fruits, and 92% for unripe fruits. However, the average frames processed per second (FPS) was only 8, which does not meet the real-time requirements of fruit-picking robots. The second type is single-stage detection models, which directly predict the position and class of objects in an image. Common single-stage detection models include the YOLO [16] series and SSD [17]. Among them, YOLO stands out as a single-stage detection algorithm and is increasingly used in fruit detection due to its excellent performance. Although YOLO's detection speed surpasses that of two-stage algorithms, its network design, intended for general scenarios with multiple categories, involves a large number of parameters and computations. In orchard fruit detection, which typically involves only a few fruit types, such complex network structures can be overly redundant. Additionally, given the cost-effectiveness and compactness requirements for harvesting robots, edge computing devices have become the preferred choice for online detection tasks. However, these devices often have lower computing performance compared to personal computers or servers, raising the challenge of effectively reducing the detection network's computational burden under limited resources. Faced with these problems, researchers have proposed some lightweight detection algorithms, especially those based on YOLO. In the field of apple detection, Tian et al. [18], used the DenseNet method to process the low-resolution feature layer in the YOLO-V3 network based on the improved YOLOv3 model. It effectively enhanced feature propagation, promoted feature reuse, and improved network performance. It explored real-time detection of apples throughout the growth stage, which can continuously monitor crop growth and nutritional status. Yan et al. [19], proposed an improved YOLOv5 model based on the BottleneckCSP module and attention mechanism, with recognition recall rate, accuracy, mAP, and F1 of 91.48%, 83.83%, 86.75% and 87.49%, respectively, which effectively improved the detection accuracy of apple fruits. It further distinguished the fruit occlusion caused by branches with high precision and realized the automatic recognition of graspable fruits. Yang et al. [20], proposed to introduce the MobileOne module in the backbone network of YOLOv7 to realize parameter fusion, and improved the SPPCSPS module, changing the serial channel to a parallel channel to increase the speed of image feature fusion. Finally, an auxiliary detection head was added to the head structure. The improved YOLOv7 algorithm has an accuracy increase of 6.9%, a recall increase of 10%, and a mAP increase of 3.8%. Ma et al. [21] proposed a lightweight model YOLOv8n-ShuffleNetv2-Ghost-SE. The model uses the ShuffleNetv2 module to replace the Backbone of YOLOv8n, the Ghost module to replace the Conv module, the C2fGhost module to replace the C2f module in the Neck part, WIoU to replace CIoU to calculate the bounding box regression loss, and embeds the SE module. The model has a mAP of 91.4%, a model size of 2.6 MB, 1.18 M parameters, and 3.9G FLOPs. Table 1 summarizes the performance of the YOLO model in the field of apple detection.

Although existing research has made progress in network lightweighting, detection accuracy is often maintained by adding attention modules or additional detection layers. While these methods help ensure accuracy, they also increase the network's computational burden, which contradicts the goal of lightweighting. The results indicate that existing networks still contain a large number of parameters and computational requirements, suggesting that current lightweight improvements are insufficient. Furthermore, many studies lack validation on edge computing devices such as Raspberry Pi, Jetson, and Arduino. Developing a lightweight apple detection algorithm that is suitable for edge computing devices without compromising accuracy remains a significant challenge.

This paper considers embedded device deployment and apple fruit detection system, and studies a lightweight apple detection algorithm based on YOLOv8 target detection algorithm. The main contributions of this paper are as follows:

1. PConv is introduced in the backbone network, and a new structure, CSPPC, is proposed to replace the C2f module. In the backbone network, CSPPC reduces computational complexity and the number of floating-point operations (FLOPs) by operating on fewer channels, which also decreases memory access—beneficial for edge devices. Additionally, CSPPC ensures effective flow of feature information.
2. FocalModulation technology is introduced to enhance feature detection by focusing on specific areas. This allows the model to more accurately capture the detailed features of apples, especially in complex environments

Table 1 Performance of the YOLO models

Model	mAP%	Parameters (M)	Detect time (ms)	FLOPs (G)
YOLOv3	92.76	61.5	11.35	32.8
YOLOv5s	84.8	13.7	1.8	16.4
YOLOv7	92.9	75	65.4	104.7
YOLOv8s	92.22	11.13	1.0	28.6

and occlusions. The technology adaptively adjusts the focus area, improving both the detection accuracy and robustness for apples.

3. A bidirectional feature pyramid network (BiFPN) is designed in the neck network. BiFPN enhances detection accuracy through bidirectional feature fusion, allowing for effective multi-scale feature integration. This capability ensures accurate detection of apples of various sizes and degrees of occlusion. Additionally, BiFPN uses a weighted feature fusion strategy to reduce the number of model parameters and computations, improving real-time detection efficiency. In complex orchard environments, BiFPN significantly enhances the model's adaptability to lighting changes and background interference by strengthening feature extraction capabilities, thereby performing well in practical applications.

2 Materials and methods

2.1 Source of experimental data

The accuracy of model detection relies on the quality of the dataset. To ensure the model can recognize apple fruit targets in various environments and sizes, experimental data for this study were collected at an eco-unmanned farm smart orchard. This orchard is a collaborative effort between Shandong University of Technology and Shandong Zhong Yi Modern Smart Agriculture Co., Ltd., located in Yiyuan County, Zibo City, Shandong Province. The orchard adheres to high-standard planting practices, with the apple variety being Red Fuji. A total of 2000 images were captured and saved in JPEG format.

In this experiment, the LabelImg annotation tool was used to draw bounding boxes around the apple objects in the original images for manual annotation. The label "apple" was assigned to the apples, and the saved format was PASCAL VOC.

To enrich the image training set, effectively extract image features, and avoid overfitting, the data set was enhanced before network training [22], including flipping, rotation, Gaussian blur and reduction, and Gaussian noise addition [23]. The images were flipped horizontally and vertically, the rotation angle was controlled between − 15° and + 15°, and the images were blurred (1.5 px) to remove noise or subtle irregularities in the image, making the image clearer and smoother. In addition, to simulate the noise that may be generated by the device during image acquisition, we enhanced the data by adding Gaussian noise with a variance of 0.02 to the original image. After data enhancement, a total of 6179 images were obtained for network training and parameter optimization. 5052 images were randomly selected from 6179 images as the training set, and the rest were used as the training set and validation set, with 564 images respectively.

2.2 Target detection method for apple fruits

2.2.1 YOLOv8 network model

Among various single-stage object detection algorithms, the YOLO series stands out for its excellent balance of speed and accuracy. It can quickly and accurately identify targets and is highly suitable for deployment on various mobile devices. Therefore, the YOLO series has been widely used in various fields such as object detection, tracking, and segmentation. YOLOv8 consists of five networks: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, each with differences in model depth and width. It is currently one of the most advanced network models and is well-suited for apple image object detection. The performance of yolov8 version in detecting Apple is shown in Table 2

In this experiment, considering factors such as model size, recognition efficiency, and accuracy, it was decided to base the apple recognition model on YOLOv8s and make improvements to design a more efficient and accurate model for apple detection.

YOLOv8s is an object detection model consisting of four parts: Input, Backbone, Neck, and Head. The model uses mosaic data augmentation techniques to improve robustness and generalization performance. In the Backbone layer, it changes the kernel size of the first convolutional layer from 6×6 to 3×3 and references the design philosophy of YOLOv7 ELAN. In the Neck layer, it removes two convolutional connection layers and replaces all C3 modules with C2f modules. In the Head layer, it transitions from Anchor-Based to Anchor-Free. During the training process, the mosaic data augmentation operation is turned off to improve accuracy. The loss calculation adopts a positive and negative sample allocation strategy. Conv, C2f, and SPPF modules are used in the model to speed up convergence and improve model performance.

Table 2 Performance of the YOLOv8 models

Model	mAP%	Parameters (M)	Detect time (ms)	FLOPs (G)
YOLOv8n	90.1	6.2	1.2	15.8
YOLOv8s	92.22	11.13	1.0	28.6
YOLOv8m	92.9	75	65.4	104.7
YOLOv8l	93.5	180.7	89.4	216.5
YOLOv8x	94.2	245.6	123.3	324.8

2.2.2 Improvements to the YOLOv8s model

The original YOLOv8s algorithm has problems such as low detection accuracy, slow detection speed, and large model size, which makes it inconvenient to deploy, especially when applied to the apple dataset discussed in this article. To meet the requirements of this task, this study mainly improves YOLOv8s in the following aspects: (1) PConv is introduced into the backbone network to improve C2f to form a new architecture CSPPC, which reduces the computational complexity and improves the detection speed. (2) FocalModulation technology is used to replace the original SPPF module to improve the model's regional recognition ability on key issues. (3) Lightweight bidirectional feature pyramid (BiFPN) is introduced to adaptively learn the importance weights of each scale, effectively retain multi-scale information through the bidirectional context information transmission mechanism, and improve the model's detection ability for occluded targets. The enhanced algorithm is named YOLOv8s-CFB, and its structure is shown in Fig. 1.

The main improvements of this article are:

1. CSPPC module

In actual deployment scenarios, the object detection model needs to be deployed on embedded devices with limited hardware memory and computing resources. In addition, to detect apples in complex environments, a certain degree of real-time performance is required. To meet these requirements, the model must be lightweight. The C2f module in YOLOv8 uses more bottleneck structures to extract more features, but it also leads to the problem of excessive redundancy of channel information. At this stage, lightweight models, such as MobileNet [24], SENet [25] and ShuffleNet [26], all use deep convolution to extract spatial features. Although the number of floating point operations (FLOPs) is reduced, it causes an increase in memory access. PConv in FasterNet uses the redundant information in the feature map to only apply conventional convolution on a part of the input channel for spatial feature extraction, while leaving the remaining channels unchanged. This can reduce computational complexity and memory access, and can maintain Feature information flow [27]. Its working principle is shown in Fig. 2. The figure (a) shows regular convolution,

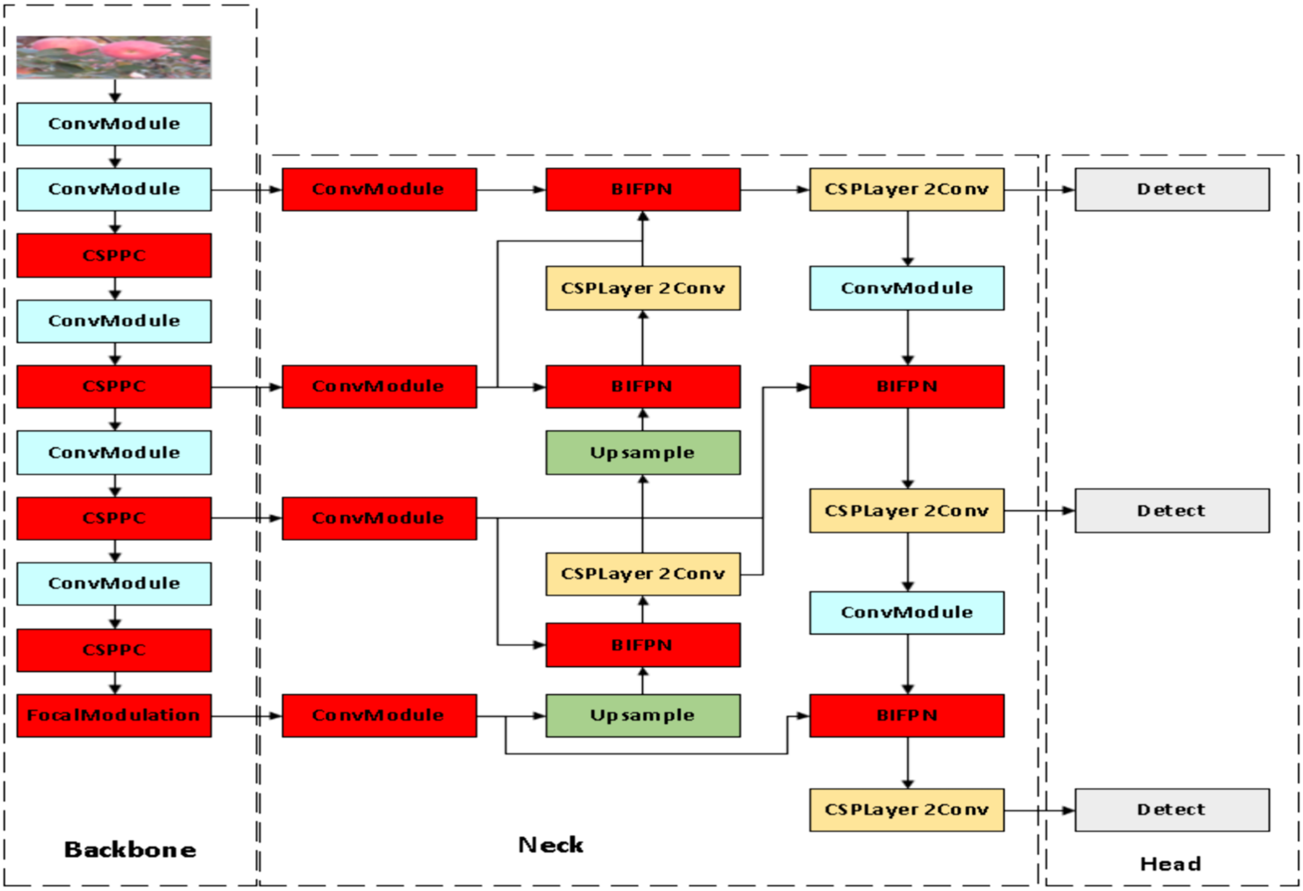


Fig. 1 YOLOv8s-CFB network structure diagram

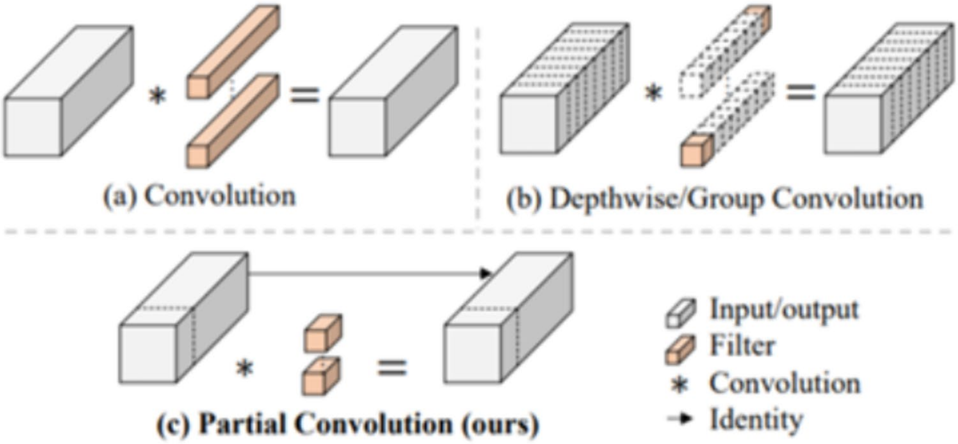

(a) Convolution
(b) Depthwise/Group Convolution
(c) Partial Convolution (ours)
Input/output
Filter
∗ Convolution
→ Identity

Fig. 2 Working principle of Pconv

(b) shows depth/grouped convolution, and (c) shows our partial convolution method. In PConv, a part of the channel is passed directly through the identity operation without convolution processing.

PConv only performs convolution operations on a part of the input channels to extract spatial features while keeping other channels unchanged. Assume that H and W represent the length and width of the input feature map, c represents the number of input channels, c_{p} represents the channels participating in the convolution, k is the convolution kernel size, r represents the participating convolution rate, and its calculation amount expression is as follows (1), the memory access meter formula is as formula (3).

$$F_{\mathrm{PConV}} = H \times W \times k^2 \times c_{\mathrm{p}}^2, \tag{1}$$

$$r = \frac{c_{\mathrm{p}}}{c}, \tag{2}$$

$$M_{\mathrm{AC}} = H \times W \times 2c_{\mathrm{p}} + k^2 \times c_{\mathrm{p}}^2 \approx h \times w \times 2c_{\mathrm{p}}. \tag{3}$$

PConv's participation convolution rate is 1/4, and its FLOPs are only 1/16 of conventional convolution. Due to the low memory consumption of the convolution process, the memory access is about 1/4 of conventional convolution.

This article uses PConV to design the C2f module, and introduces PConV by building CSPPC to further reduce the amount of calculation and floatingpoint numbers. The CSPPC module structure diagram is shown in Fig. 3.

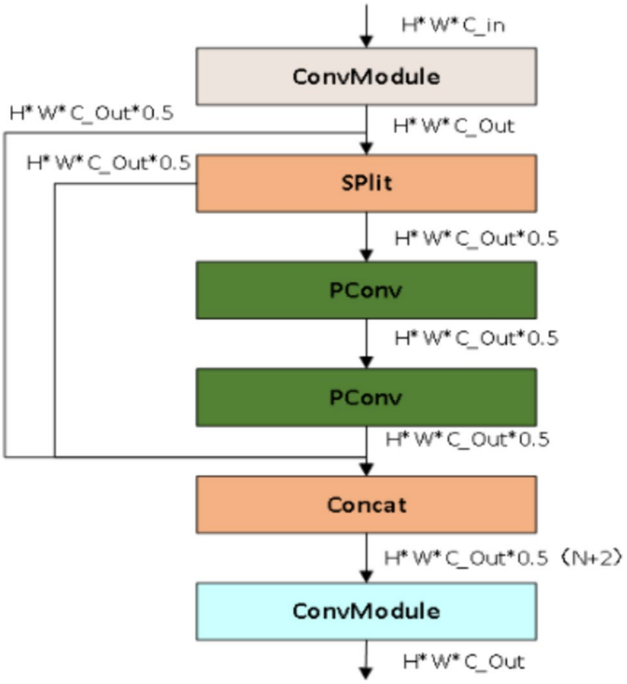


Fig. 3 CSPPC module structure diagram. C is the number of channels of the input feature map; * is the convolution operation; H and W are the height and width dimensions of the feature map respectively; Conv is the conventional convolution module; PConv is the partial convolution; Split is the channel segmentation module

2. FocalModulation technology

Due to the complex and variable environment of apple detection, improving the model's ability to extract key apple features is crucial. To address this, FocalModulation technology is introduced in the YOLOv8 backbone network to replace the original SPPF module. This technology minimizes the interference from irrelevant background information and enhances the focus on more effective apple feature information. The self-attention mechanism, while offering strong long-range dependency and adaptability, has limitations. It neglects the two-dimensional structure of images and incurs significant computational costs, particularly when processing large convolution kernels. FocalModulation technology is introduced to address the excessive computational load associated with self-attention using smaller computational costs, thereby achieving higher performance.The calculation process for self-attention involves an initial interaction followed by aggregation. The formula for this process is:

$$y_i = M_1(T_1(x_i, X), X) \tag{4}$$

where M_1 represents the aggregation process and represents the interaction process.

The working principle of FocalModulation technology covers three key aspects. First, the aggregated features are fused into the query via modulation or element-level affine transformation. This process adopts an interaction first and then aggregation method, which is different from the self-attention (SA) model, which first aggregates features and then interacts the query with the aggregated features to fuse contextual information. Second, hierarchical semantics are introduced to extract contextual information at different granularity levels from local to global scope. This step captures local and global contextual information by projecting the input features to a new feature space and then using L depth-wise convolutions to obtain a hierarchical representation. Finally, gated aggregation is applied to condense contextual features at different granularity levels into a single feature vector, forming a modulator. Spatially and level-aware weights are obtained via linear layers, and then a weighted sum is performed via element-wise multiplication to obtain a single feature map of the same size as the input. This process models the relationship between different channels and forms the calculation process of the overall focus modulation. Combining the interaction and aggregation formulas, the overall focus modulation formula can be expressed as:

$$y_i = q(x_i) \times h\left(\sum_{\ell+1}^{L+1} g_i^{\ell} \times z_i^{\ell}\right), \tag{5}$$

where g_i^{ℓ} and z_i^{ℓ} are the gating value and visual feature at location i of G^{ℓ} and Z^{ℓ}, respectively.

Its structure diagram is shown in the Fig. 4, Fig. 4 shows FocalModulation technology on the left. Right: Detailed illustration of context aggregation in FocalModulation.

3. Feature fusion layer improvements

The goal of multi-scale feature fusion is to integrate features of different resolutions. PANet uses two paths, bottom–up and top–down, which are better than FPN, but require more calculations. BiFPN is developed on the basis of PANet, removing the situation of only one input node, because this node contributes less to feature fusion. To better integrate features, an additional path is added to connect the input nodes and output nodes of the peer network [28]. The original BiFPN structure is shown in Fig. 5a. In this paper, we improved the BiFPN structure and added a P2 large-size feature layer to improve the feature fusion ability of apple fruit distant targets. The improved BiFPN structure is shown in Fig. 5b.

During the feature fusion process, the resolution differences of different input features lead to their different contributions to the output features. Under field conditions, collected multiscale apple targets are common. In order to balance the weights of different features, mine the deep information of apple targets, and reduce false detections and missed detections caused by environmental complexity,

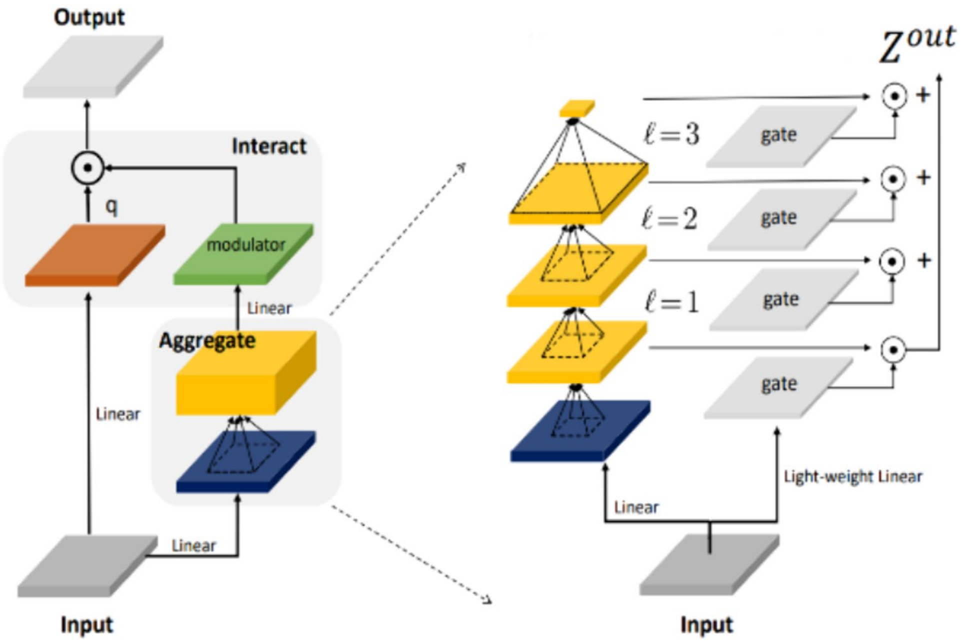


Fig. 4 FocalModulation

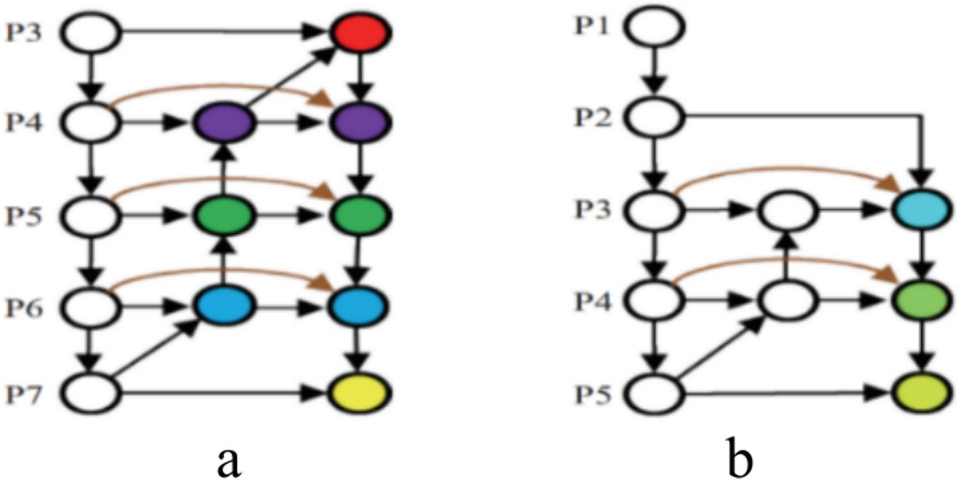


Fig. 5 BiFPN structure

BiFPN adopts a fast normalization fusion module. This module is designed to effectively adjust the weights between features to better reflect their importance to the object detection task. Therefore, the relationship between the input and output of BiFPN can be expressed as.

$$O = \sum_i \frac{\omega_i}{\in + \sum_j \omega_j} \times I_i, \tag{6}$$

where, ω_i is the learning weight corresponding to the input feature I_i, which is ensured by applying ReLU in the subsequent stage $\omega_i \geq 0$. To avoid numerical instability, set the initial learning rate ε = 0.000 1, and the value of the normalized weight decreases between 0 and 1.

To further improve efficiency, we employ depthwise separable convolutions in the feature fusion stage and subsequently add batch normalization and activation operations. This paper replaces the feature fusion network with BiFPN, so that features of different scales can obtain different weights, and performs cross-scale weight suppression or feature expression to enhance feature fusion, thereby further improving target detection performance.

2.3 Model training and evaluation

2.3.1 Experimental environment

The operating system used in this experiment is Win11 and the in-depth learning framework development environment of Py3.9.18, CUDA11.6 and Pyr1.12.1 is used. The processor carried by the computer is Intel Core i7-12700H@3.80 GHz, memory of 32 GB, video card of RTX3060 and video memory of 6 GB. The input picture size is adjusted to, the Batchsize is set to 8, the thread is set to 2, and a total of 300 epochs are trained.

2.3.2 Evaluation index

Evaluation indicators are important tools for quantitatively evaluating model performance. For accuracy evaluation, we used the mAP indicator, which is calculated as follows

$$\mathrm{MAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \times 100\%, \tag{7}$$

mAP is the average of the mean precision (AP) and is determined by integrating the precision–recall (P–R) curve:

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R. \tag{8}$$

In this study, there is only one apple class N equal to 1. Precision (P) is the ratio of correctly predicted Apple instances to the total number of predicted Apple instances, and recall (R) is the ratio of correctly predicted Apple instances to the total number of true Apple instances. They are calculated as:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\%, \tag{9}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\%. \tag{10}$$

Among them, true positives (TP) represent the number of instances that are actually apples and are predicted to be apples. False positives (FP) represent the number of instances that are not actually apples but are predicted to be apples. False negatives (FN) represent the number of instances that are actually apples but are not predicted to be apples. In terms of speed and efficiency, we use parameters, detection time and FLOPs, and to evaluate the computational complexity and real-time performance of the model.

3 Results and analysis

3.1 Comparison with other lightweight networks

This experiment is based on YOLOv8s as the base network. The backbone is replaced by mainstream lightweight networks (such as MobileNetV3, SENet, ShuffleNetV2) and improved light backbone networks. While keeping other parameters the same, the training effects of different backbone networks on the target are compared. The evaluation indicators include mAP, detection time, FLOPs and Parameter.

As shown in Table 3, the CSPPC improvement method has great advantages in parameter quantity, average precision, detection time and floatingpoint number compared with MobileNetV3, SENet and ShuffleNetV2. The average precision is improved by 3.5, 7.47 and 10.01 percentage points respectively compared with other networks. The number of parameters of some lightweight model improvement networks is smaller than that of this method, but the model has poor feature extraction ability and cannot meet the deployment requirements.

Table 3 Comparison of different lightweight feature extraction backbone networks

Model	Map%	Parameters (M)	Detection time (ms)	FLOPs (G)
YOLOv8s-M	88.92	11.51	0.9	23.4
YOLOv8-SE	84.95	6.28	0.4	23.0
YOLOv8s-S	82.41	6.31	0.4	22.1
YOLOv8s-CSPPC	92.42	9.31	0.6	23.3

YOLOv8s-M new network after Backbone is replaced by MobileNetV3, *YOLOv8s-SE* new network after Backbone is replaced by SENet, *YOLOv8s-S* new network after Backbone is replaced by ShuffleNetV2, *YOLOv8s-CSPPC* new network after the C2f module is designed with PConv

3.2 Ablation experiment

To verify the impact of the three improvement strategies in this study on the model, eight ablation experiments were conducted and the results are shown in Table 4.

As shown in Table 4, Experiment 1 uses the original YOLOv8s network model. The model's mAP for apple recognition is 92.22%, the number of parameters is 11.13 M, and the average detection time of each photo is 1.0 ms, FLOPs at 28.6 G; Experiment 2 is to combine the backbone of the original YOLOv8s model. The network introduces partial convolution (PConv) to improve the C2f module to form a new architecture CSPPC. Compared with Experiment 1, mAP is increased by 0.2%, the number of parameters is reduced by 1.82 M, and the average time of each photo is reduced by 5.3 ms, FLOPs decrease, The reason is that only some channels are convolved, which effectively reduces the model weight and computational redundancy, while losing a small number of features that may be contained in the remaining channels, resulting in a slight decrease in average accuracy; Experiment 3 On the basis of Experiment 1, FocalModulation technology was used to replace the original SPPF module. Compared with Experiment 1, mAP increased by 1.08% and detection time was increased by 0.1 ms, FLOPs remain almost unchanged, It shows that FocalModulation technology can improve the fitting degree of the model and improve the model recognition accuracy; Experiment 4 introduced the BiFPN idea in the Neck layer. The model Compared with Experiment 1, mAP, parameter amount, detection time and FLOPs are almost all better than Experiment 1, This shows that BiFPN can fuse more features and assign more weights to the correct features through the weighted feature fusion mechanism; Experiments 5, 6, and 7 are two combinations of the improved strategies. Compared with Experiment 1, the improved model is almost better than Experiment 1 in terms of mAP value, number of parameters, detection time and FLOPs; Experiment 8 is the YOLOv8s model improved in this study, with mAP reaching 93.86%, parameter amount reduced by 2.3, and detection time only 0.3 ms. The ablation test proves that this experiment is

Table 4 Comparison of ablation test results

Test no.	CSPPC	FocalModulation	BiFPN	Mean average precision Map%	Parameters (M)	Detection time (ms)	FLOPs (G)
1	×	×	×	92.22	11.13	1.0	28.6
2	√	×	×	92.42	9.31	0.6	23.3
3	×	√	×	93.30	11.54	1.1	29.0
4	×	×	√	92.56	10.24	0.6	27.8
5	√	√	×	93.62	9.73	0.6	23.6
6	√	×	√	92.88	8.42	0.6	23.5
7	×	√	√	93.81	10.65	0.7	29.1
8	√	√	√	93.86	8.83	0.70	23.8

effective for YOLOv8s. The improvements to the model all have positive effects.

3.3 Comparison before and after improvement of YOLOv8 model

To study the recognition ability of the improved YOLOv8s model to the covering condition of complex apple orchard, the samples with different illumination degree and different covering condition are selected for comparative test, as shown in Fig. 6. Under the positive light condition, the recognition accuracy of YOLOv8s and improved YOLOv8s is very high, but under the back light and low light intensity condition, the recognition accuracy of YOLOv8s to apple decreases, The improved YOLOv8s has a good effect on identifying this situation. Under different sheltering conditions, the recognition accuracy of YOLOv8s model is very high for non-sheltering conditions, but the accuracy of branch and leaf sheltering and fruit sheltering is reduced; However, the improved YOLOv8s model has high accuracy for the non-occlusion condition, and has good effect for the identification of the other two occlusion conditions.

The identification effect of improved model and original model is shown in Table 5.

3.4 Model feature visualization

To more intuitively observe the improvement of the recognition ability of the model feature fusion network BiFPN, Grad-CAM [29] (gradient-weighted class activation mapping) is used to draw a heat map, which can more intuitively see the learning of the network for different targets. Grad-CAM uses the training weight back propagation to perform global average pooling on the obtained gradient matrix in the spatial dimension, and then weightedly activates each channel of the feature layer to obtain a heat map. The brightness depth of a certain area in the heat map can show the part of the image that has a greater impact on the model output. The heat map before and after adding BiFPN is shown in Fig. 7.

Compared with Fig. 7b, the color of the apple target in Fig. 7c is brighter and the response is higher, while the brightness of the incorrectly extracted leaf features is lower. The use of the fast normalization fusion module enhances the model's perception of the correct target and suppresses the impact of the wrong samples on the overall prediction, allowing the model to focus more accurately on the apple target features.

3.5 Comparison of different inspection models

To verify the effectiveness of the YOLOv8s-CFB model in apple fruit detection, we compared it with advanced object detection algorithms such as Faster R-CNN (ResNet50), SSD, YOLOv5s [30], YOLOv7-tiny, MobileNetv3_small_Faster, ShuffleNetv2, Faster R Former [31] and DETR. The training cycle was set to 300 rounds, and the apple fruit dataset was trained based on the above object detection algorithms. The detection and evaluation of these algorithms were performed using the test set. The results are shown in Table 6. The improved model greatly reduces the model size, parameter amount, and computational complexity while maintaining high detection accuracy, which is conducive to the migration and application of the model to hardware platforms such as edge devices, embedded systems, and dedicated chips.

3.6 Edge device deployment

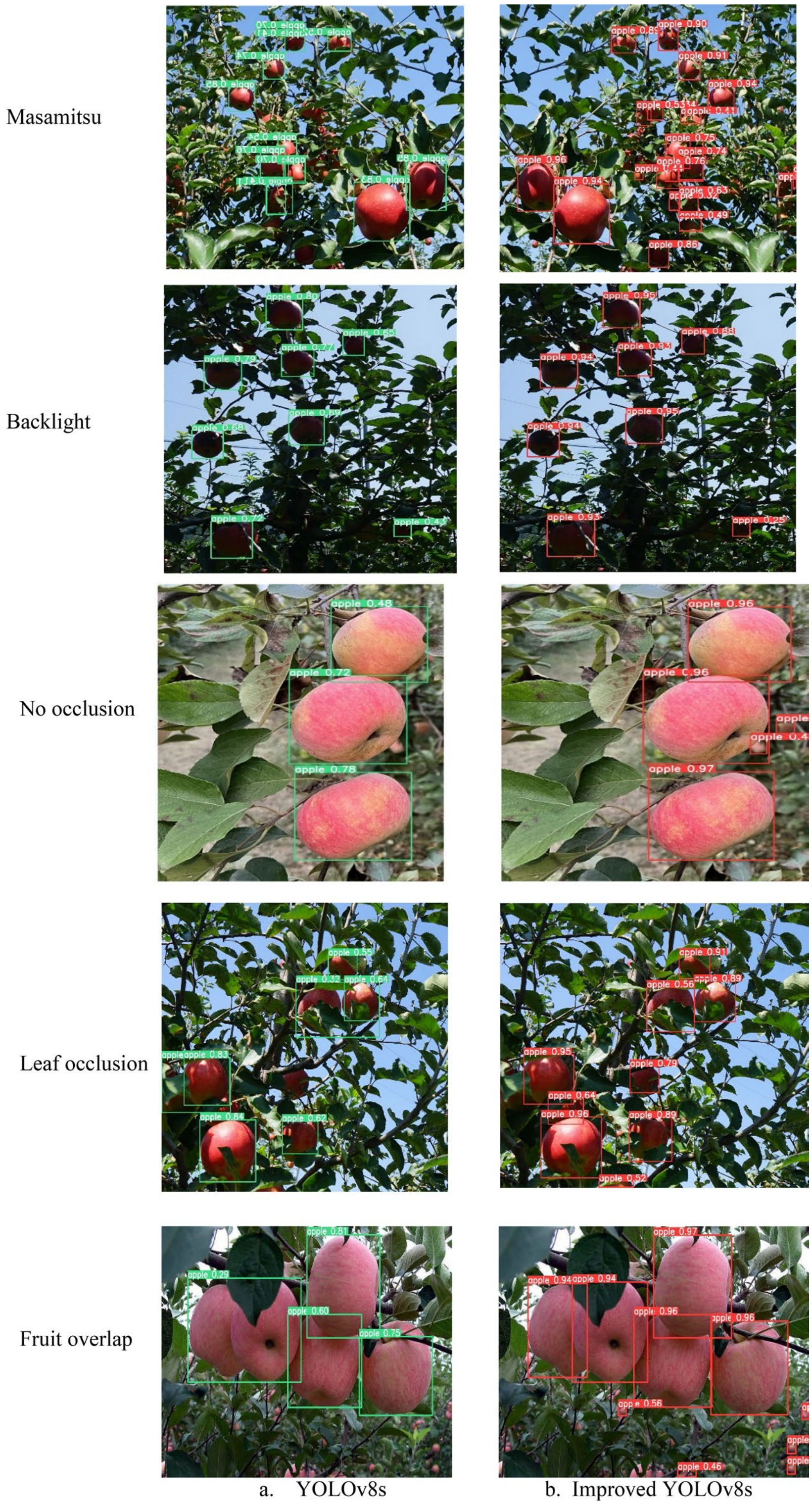
To verify and improve the edge device deployment of the YOLOv8s-CFB model, and to improve the detection speed of the model, the TensorRT inference library is selected for acceleration. TensorRT is a high-performance inference optimization framework released by Nvidia, which can provide low-latency and high-throughput deployment inference acceleration for the model on Nvidia GPU. The YOLOv8s-CFB model training weight file is converted into a wts intermediate file and imported into Jetson Nano for compilation, and the model object is serialized to generate the engine inference engine. The deserialization operation of the engine file can realize the inference and post-processing operations. The device deployment detection situation is shown in Table 7.

As shown in Table 7, before TensorRT acceleration, the improved YOLOv8s-CFB model has a relatively low detection speed due to the limited computing power of embedded devices. After acceleration, the model detection speed is increased by 7.9 times, the detection frame rate is 49 frames/s, and the single-image detection speed is 20.41 ms.

4 Conclusion

1. This paper proposes a lightweight apple target detection algorithm, YOLOv8s-CFB, based on an improved YOLOv8s convolutional neural network. The algorithm achieves an average accuracy of 93.86% with a model parameter volume of 8.83 MB. Compared to the original model, the improved version reduces the parameter volume to 79.33% of the baseline network, increases accuracy by 1.64 percentage points, and also reduces detection time and FLOPs. It offers certain advantages over mainstream target detection models, including SSD, Faster RCNN, YOLOv5s, YOLOv7, and YOLOv8s.
2. The introduction of PConv to create C2f-PConv reduced the model weight by 25.33% and provided better detection performance compared to using lightweight back-

Fig. 6 Comparison of model effects before and after improvement

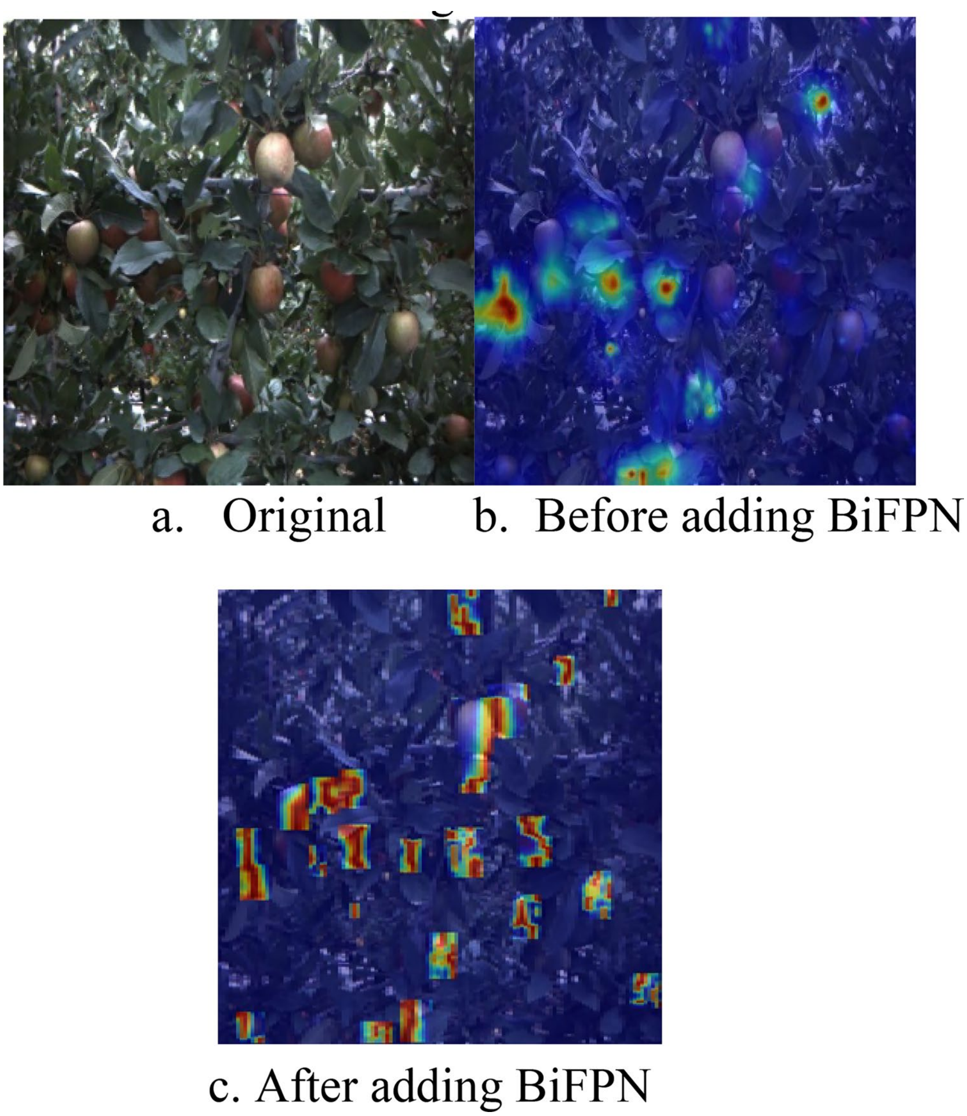


a. YOLOv8s b. Improved YOLOv8s

Table 5 Comparison of YOLOv8 model before and after improvement

Model	mAP%	Parameters (M)	Detect time (s)	FLOPs (G)
YOLOv8s	92.22	11.13	1.0	28.6
Ours	93.86	8.83	0.7	23.8

bone networks like MobileNetV3, SENet, and ShuffleNet. Additionally, replacing the backbone network with BiFPN enhanced the model's feature fusion capability, reducing the model parameters to 75.65% of the original YOLOv8s.


a. Original b. Before adding BiFPN

c. After adding BiFPN

Fig. 7 Model heat map visualization

Table 6 Comparison of results of different models

Model	mAP%	Parameters (M)	Detect time (ms)	FLOPs (G)
Faster R-CNN(ResNet50)	77.41	28.28	0.42	462.0
SSD	84.91	23.61	3.15	87.4
YOLOv5s	96.36	7.01	1.10	15.8
TPH-YOLOv5	94.22	11.80	17.05	15.4
YOLOv7-tiny	96.51	6.01	1.16	13.2
MobileNetv3_small_Faster	93.91	2.44	1.87	5.8
ShuffleNetv2	90.28	1.38	3.16	4.3
Faster R Former	69.24	43.96	3.50	93.55
Ours	93.86	8.83	0.70	23.8

3. Replacing the original SPPF module with FocalModulation technology increased accuracy by 1.08 percentage points and improved model fitting speed. After deploying the model with the TensorRT inference library, the detection frame rate reached 49 frames per second, meeting the requirements for edge device deployment.

Table 7 Comparison of device deployment detection frame rates

Models	Desktop computers	Embedded devices	TensorRT
YOLOv8s	81.9	5.4	–
YOLOv8s-CFB	107.5	6.2	49

Acknowledgements This work was funded by the Key Research and Development Program of Shandong Province (Major Innovative Project in Science and Technology) (2020CXGC010804), Shandong Provincial Natural Science Foundation (ZR2021MC026).

Author contributions Bing Zhao designs experimental plans, collects data, and writes papers. Aoran GUO participates in data analysis, provides experimental technical support, and reviews papers. Ruitao MA designs research framework, analyzes experimental results, and writes partial paper content. Yanfei ZHANG provides professional knowledge support, participates in discussions, and explains experimental results. Jinliang GONG is responsible for project management, guiding research directions, reviewing and revising papers.

Data availability No datasets were generated or analysed during the current study.

Code and dataset availability The data presented in this article are public-ly available in [zenodo] at [https://doi.org/10.5281/zenodo.11609498]. The archived version of the code described in this manuscript can be freely accessed through[zenodo] [https://doi.org/10.5281/zenodo.11609628].

Declarations

Conflict of interest B. Zhao, A. Guo, R. Ma, Y. Zhang, and J. Gong declare that they have no competing interests.

References

1. Liu, J., Liu, Z.: YOLOv5s-BC: an improved YOLOv5s-based method for real-time apple detection. J Real-Time Image Process **21**, 88 (2024). https://doi.org/10.1007/s11554-024-01473-1
2. Nesterov, D.A., Shurygin, B.M., Solovchenko, A.E., Krylov, A.S., Sorokin, D.V.: A CNN-based method for fruit detection in apple tree images. Comput. Math. Model. **33**, 354–364 (2022). https://doi.org/10.1007/s10598-023-09577-2
3. Mao, Z., Wang, W., Yang, H.: Apple object detection in natural environment based on YOLO v5. In: 2023 2nd International Conference on Applied Statistics, Computational Mathematics and Software Engineering (ASCMSE 2023), vol. 12784. SPIE (2023). https://doi.org/10.1117/12.2691833
4. Jia, W., et al.: Apple harvesting robot under information technology: a review. Int. J. Adv. Robot. Syst. **17**, 1729881420925310 (2020). https://doi.org/10.1177/1729881420925310
5. Yoshida, T., Onishi, Y., Kawahara, T., Fukao, T.: Automated harvesting by a dual-arm fruit harvesting robot. ROBOMECH J **9**, 19 (2022). https://doi.org/10.1186/s40648-022-00233-9
6. He, B., Qian, S., Niu, Y.: Visual recognition and location algorithm based on optimized YOLOv3 detector and RGB depth camera. Vis. Comput. **40**, 1965–1981 (2024). https://doi.org/10.1007/s00371-023-02895-x
7. Zhao, Y., Gong, L., Huang, Y., Liu, C.: A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. **127**, 311–323 (2016). https://doi.org/10.1016/j.compag.2016.06.022
8. Gao, F., et al.: Multi-class fruit-on-plant detection for apple in SNAP system using faster R-CNN. Comput. Electron. Agric. **176**, 105634 (2020). https://doi.org/10.1016/j.compag.2020.105634
9. Yoshida, T., Kawahara, T., Fukao, T.: Fruit recognition method for a harvesting robot with RGB-D cameras. ROBOMECH J **9**, 15 (2022). https://doi.org/10.1186/s40648-022-00230-y
10. Linker, R., Cohen, O., Naor, A.: Determination of the number of green apples in RGB images recorded in orchards. Comput. Electron. Agric. **81**, 45–57 (2012). https://doi.org/10.1016/j.compag.2011.11.007
11. Prakash, A.J., Prakasam, P.: An intelligent fruits classification in precision agriculture using bilinear pooling convolutional neural networks. Vis. Comput. **39**, 1765–1781 (2023). https://doi.org/10.1007/s00371-022-02443-z
12. Kasinathan, T., Uyyala, S.R.: Detection of fall armyworm (spodoptera frugiperda) in field crops based on mask R-CNN. Signal Image Video Process **17**, 2689–2695 (2023). https://doi.org/10.1007/s11760-023-02485-3
13. Cen, H.: Target location detection of mobile robots based on R-FCN deep convolutional neural network. Int. J. Syst. Assur. Eng. Manag. **14**, 728–737 (2023). https://doi.org/10.1007/s13198-021-01514-z
14. Zhang, J., et al.: Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. Comput. Electron. Agric. **173**, 105384 (2020). https://doi.org/10.1016/j.compag.2020.105384
15. Zhu, X., et al.: Detecting and identifying blueberry canopy fruits based on Faster R-CNN. J South Agric (2020). https://doi.org/10.3969/j.issn.2095-1191.2020.06.032
16. Badgujar, C.M., Poulose, A., Gan, H.: Agricultural object detection with You Only Look Once (YOLO) Algorithm: a bibliometric and systematic literature review. Comput. Electron. Agric. **223**, 109090 (2024). https://doi.org/10.1016/j.compag.2024.109090
17. Wu, D., Lv, S., Jiang, M., Song, H.: Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput. Electron. Agric. **178**, 105742 (2020). https://doi.org/10.1016/j.compag.2020.105742
18. Tian, Y., et al.: Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. **157**, 417–426 (2019). https://doi.org/10.1016/j.compag.2019.01.012
19. 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 **13**, 1619 (2021)
20. Yang, H., et al.: Improved apple fruit target recognition method based on YOLOv7 model. Agriculture **13**, 1278 (2023)
21. Ma, B., et al.: Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. Artif Intell Agric **11**, 70–82 (2024). https://doi.org/10.1016/j.aiia.2024.02.001
22. Guan, C., Jiang, J., Wang, Z.: Fast detection of face masks in public places using QARepVGG-YOLOv7. J Real-Time Image Process **21**, 95 (2024). https://doi.org/10.1007/s11554-024-01476-y
23. Chen, J., et al.: An improved Yolov3 based on dual path network for cherry tomatoes detection. J. Food Process Eng **44**, e13803 (2021). https://doi.org/10.1111/jfpe.13803
24. Liu, G., Wen, H.: Printed circuit board defect detection based on MobileNet-Yolo-Fast. J J Electron Imaging (2021). https://doi.org/10.1117/1.JEI.30.4.043004

25. Jin, X., et al.: Delving deep into spatial pooling for squeeze-and-excitation networks. Pattern Recognit **121**, 108159 (2022). https://doi.org/10.1016/j.patcog.2021.108159
26. Zhang, X., Zhou, X., Lin, M., Sun, J.: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6848–6856 (2018)
27. Chen, J., et al.: In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12021–12031 (2023)
28. Tan, M., Pang, R., Le, Q. V.: In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10781–10790 (2020)
29. Selvaraju, R.R., et al.: In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2017)
30. Li, F., Zheng, Y., Liu, S., Sun, F., Bai, H.: A multi-objective apple leaf disease detection algorithm based on improved TPH-YOLOV5. Appl Fruit Sci **66**, 399–415 (2024). https://doi.org/10.1007/s10341-024-01042-7
31. Kong, X., Li, X., Zhu, X., Guo, Z., Zeng, L.: Detection model based on improved faster-RCNN in apple orchard environment. Intell Syst Appl **21**, 200325 (2024). https://doi.org/10.1016/j.iswa.2024.200325

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.