Enhanced progressive fusion method for the efficient detection of multi-scale lightweight citrus fruits

Yanlin Zeng[1], Yao Lin[1], Yiting He[1], Tong Li[1,2], Jing Li[3], Baijuan Wang[3], Yi Yang[1,2*]

(1. *College of Big Data, Yunnan Agricultural University, 650201, Kunming, China*;
2. *Yunnan Agricultural University, The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Kunming 650201, China*; 3. *The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Kunming 650201, China*)

Abstract: Human labor efficiency has become unable to keep the pace with gradually annual citrus increasing production. Highly efficient and intelligent citrus picking and accurate yield estimation is the key to solve the problem. Success heavily depends on detection accuracy, prediction speed, and easy model deployment. Traditional target detection methods often fail to achieve balanced results in all those aspects. An improved YOLOv8 network model with four significant features is proposed. First, a lightweight FasterNet network structure was introduced to the backbone network, which reduced the number of parameters and computations while maintaining high-precision detection. Second, a progressive feature pyramid network AFPN structure was added to the neck network. Third, a parallel multi-branch attention mechanism PMBA was added before the detection head to improve the sensing ability after the feature fusion network. Fourth, a Wise-IoU was introduced to replace the original CIoU loss function to make the whole training process converge faster. Based on this, this study proposes an improved version of the YOLOv8 model: the FAP-YOLOv8. This improved model achieved an average accuracy (mAP@0.5) of 97.2% on the citrus datasets, with an accuracy that was 4.7% higher than the original YOLOv8, which was 19.2%, 7.4%, 5.1%, 4.9%, and 5.2% higher than the other models: Faster R-CNN, CenterNet, YOLOv5s, YOLOx-s, and YOLOv7, respectively. The number of parameters was reduced by 55.45%, the computation was reduced by 20% compared to the YOLOv8 benchmark, and the frame rate reached 46.51 fps to meet the detection requirements of lightweight networks. The experiments showed that the FAP-YOLOv8 models all outperformed the comparison models. Consequently, the proposed FAP-YOLOv8 model can help solve the citrus detection problem in orchards, which can be better applied to edge devices and provides strong support for intelligent orchard management.

Keywords: citrus fruit detection, enhanced progressive fusion model, multi-scale lightweight, attention mechanism
DOI: 10.25165/j.ijabe.20241706.8802

Citation: Zeng Y L, Lin Y, He Y T, Li T, Li J, Wang B J, et al. Enhanced progressive fusion method for the efficient detection of multi-scale lightweight citrus fruits. Int J Agric & Biol Eng, 2024; 17(6): 218–229.

1 Introduction

With the rapid popularization of smart agriculture, the intelligence of China's citrus industry is in a rapid development stage. The realization of automated citrus harvesting, accurate citrus yield prediction, and intelligent management have become essential goals.

In recent years, although the field of agricultural artificial intelligence has taken off[1], three key problems need to be solved to realize agricultural picking intelligence[2] and accurate prediction of citrus yield[3]. First, the recognition accuracy needs to be further improved; second, the inference speed of the model should match the production demand; and finally, lightweight deployment is crucial for citrus recognition. Deep learning[4] has made rapid development in the past decades, and many excellent modules and networks have been proposed, but many of them are still in the theoretical stage, lack practical applications, or cannot completely solve the above problems, and need further improvement and progress. In terms of fruit detection[5-8], there are many researchers who have achieved good research results, and the research in this study must be carried out on the basis of the previous work to promote the development of intelligence in the citrus industry.

In response to the above problems, scholars at home and abroad have researched fruit recognition and detection and proposed many new algorithms. In the traditional digital image processing, Gao et al.[9] proposed a multilevel apple detection method based on fast regional convolutional neural networks, which can detect apples under different conditions but does not address the recognition in high-density complex environments. Kukreja et al.[10] proposed a dense CNN network, which provides ideas for citrus quality detection by designing data enhancement and pre-processing techniques. However, in real orchard environments, the accuracy of traditional machine vision inspection is often unsatisfactory due to the different levels of occlusion between leaves, fruits, and branches[11]. To solve the lightweight problem of the model, Liu et al.[12] used MobileNetv2 as the backbone network for citrus disease detection. In addition, Qiu et al.[13] investigated a model compression method based on knowledge distillation, successfully achieving optimized results in reducing parameters and improving detection speed. However, Liu et al.[14] used a convolutional neural network that can detect leaves, branches, and fruits hidden by branches or leaves. The hierarchical contour analysis algorithm proposed by Lu et al.[15] was able to detect green citrus on trees, but the time efficiency of their algorithm still needs to be improved. Bi et al.[16]

Received date: 2024-01-09 **Accepted date:** 2024-10-12
Biographies: Yanlin Zeng, ME, research interest: computer vision, Email: 786823791@qq.com; **Yao Lin**, ME, research interest: computer vision, Email: 1252760643@qq.com; **Yiting He**, ME, research interest: computer vision, Email: 790194799@qq.com; **Tong Li**, Professor, research interest: smart agriculture, Email: tli@ynu.edu.cn; **Jing Li**, Professor, research interest: smart agriculture engineering, Email: lijing69@ynau.edu.cn; **Baijuan Wang**, Professor, research interest: smart tea, Email: 314566690@qq.com.
***Corresponding author: Yi Yang**, Professor, research interest: agricultural informatization. College of Big Data, Yunnan Agricultural University, Kunming 650201, China. Tel: +86-538-8241865, Email: yyang66@126.com.

proposed a citrus visual recognition model using multiple segmentation methods, but the detection effect is poor in complex situations. Zhang et al.[17] overcame inconsistent fruit detection accuracy and repeated counts of the same fruit. Utilizes video sequences to help overcome these problems. Mitigated the double counting problem associated with occluded fruits. Zhuang et al.[18] proposed a citrus fruit detection method that achieves robust citrus region localization under different lighting conditions through a multi-step process of local isomorphic filtering, threshold processing, and morphological operations. However, the performance of the method under cloudy conditions still needs to be further improved, and there are 13 false detections. In addition, Lin et al.[19], based on RGB-D image analysis for citrus detection and localization in the field is greatly affected by the complex background. Although Chen et al.[20] implemented a citrus detection algorithm in an orchard environment using a multi-scale lightweight and efficient model of YOLOv7, the algorithm was less effective in detecting dense citrus trees. Similarly, Lyu et al.[21] implemented citrus detection and counting in orchards using the YOLOv5 algorithm and an edge computing system, which optimized resource utilization, but the detection efficiency was still not high in complex scenarios. Yang et al.[22] implemented an apple target detection method using YOLOv7, but the algorithm cannot detect dense scenes and is highly affected by fruit tree leaves and branches.

2 Introduction to Yolov8

YOLOv8 is the latest work in the YOLO (You Only Look Once)[23]series, open-sourced by Ultralytics in 2023, and is the one of most advanced target detection model. The single-stage detection algorithm first came into the limelight in 2015 when YOLOv1 was proposed. It effectively solved the problem of slow inference in two-stage detection networks and excelled in detection accuracy. Subsequently, YOLOv3[24], as an improved version of the previous work, introduced the residual module Darknet-53 and the FPN architecture, which realized the prediction of objects at three different scales and multi-scale fusion. Since then, YOLOv4[25] and YOLOv5 have added a number of tricks to the versions. In 2022, YOLOv7[26] was created, innovating the Extended-ELAN architecture, which improves the self-learning ability of the network without destroying the original gradient paths. In addition, it uses a cascade-based model scaling approach to generate models of appropriate scale for real-world tasks to meet the detection needs.

YOLOv8, on the other hand, is a significant improvement based on the YOLOv5 project. It refers to the design idea of YOLOv7 ELAN and improves the backbone network and the neck part, replaces the C3 structure of YOLOv5 with the C2f structure, which is richer in gradient flow, and adjusts the number of channels for the models of different scales. The head part has a big change compared to YOLOv5 and adopts the current mainstream decoupled head structure, which separates the classification head and the detection head and also shifts from anchor-based detection to anchor-free detection. These improvements enable YOLOv8 to achieve higher performance and accuracy in target detection. Figure 1 shows a diagram of the YOLOv8 architecture.

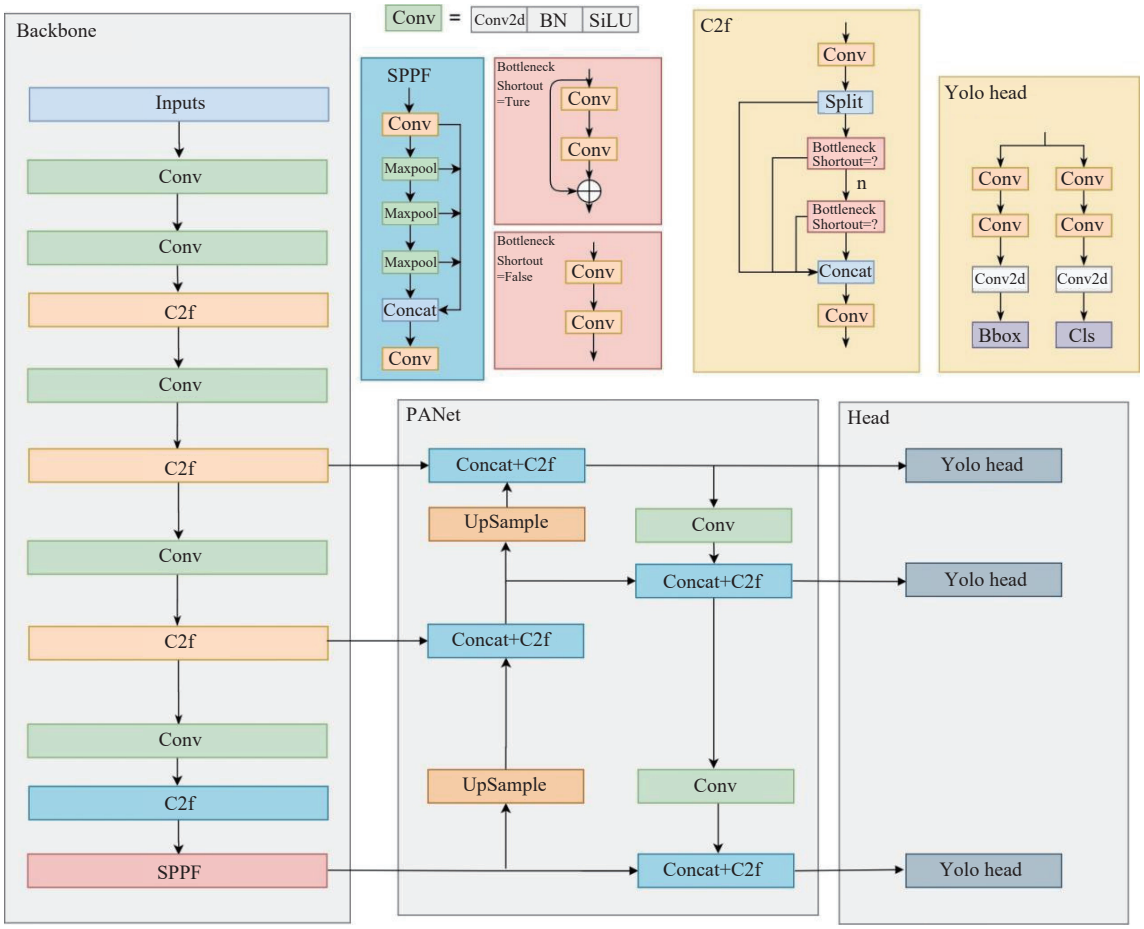


Figure 1 YOLOv8 structural frame diagram

3 Yolov8 algorithm improvement

3.1 Improvement strategy

The citrus fruit detection task is prone to problems such as background interference, occluded fruits to be detected, and different sizes. This study proposes to improve the YOLOv8 network model, and the specific structure is shown in Figure 2. Based on the YOLOv8 model, four improvements are proposed:

first, the backbone part of the backbone network is improved, and the original C2f structure is replaced by the improved C2f_Faster structure, which reduces the number of parameters and computations while increasing the effective information associated with location awareness. The improved AFPN feature pyramid structure is then introduced to strengthen the neural network's ability to perceive the feature region, and the problem of feature information loss or degradation in the traditional feature pyramid method is solved by gradually fusing the features of non-adjacent layers for feature fusion so that the model can more accurately locate and identify the target of interest. The improved parallel multi-branch attention mechanism PMBA is added again to enhance the ability to extract semantic features and improve the accuracy of feature extraction; finally, the loss function in the prediction part is improved, and Wise-IoU replaces the existing CIoU loss calculation to improve the detection of targets with different sizes and shapes.

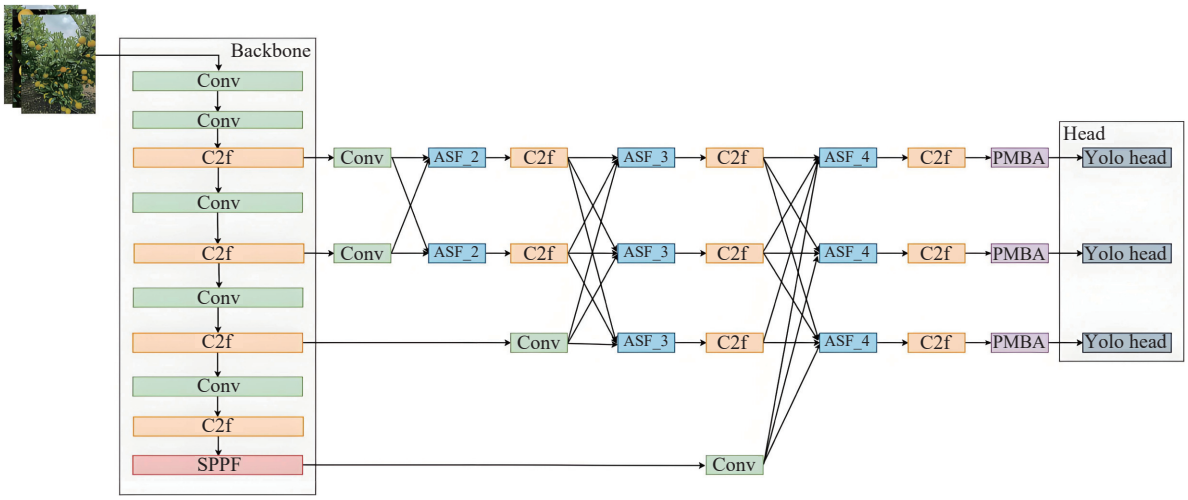


Figure 2 Improvement of YOLOv8 structural frame diagram

3.2 FasterNet structure

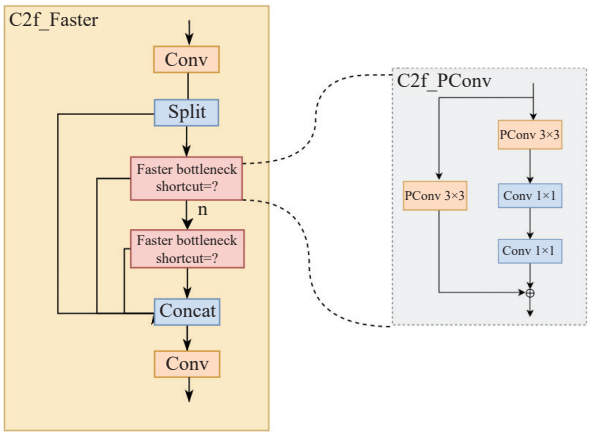
In traditional neural network optimization, reducing the number of floating point operations (FLOPs) is considered an effective way to improve network performance. However, practical observations have shown that reducing FLOPs alone does not significantly reduce the computational latency of neural networks. This is mainly due to the fact that the inefficient number of floating point operations per second (FLOPS) becomes a major bottleneck in network computation.

To overcome this problem, academic researchers have re-examined commonly used operators and found that the inefficient FLOPS are mainly due to the frequent memory accesses of operators, especially deep convolutional operations. Therefore, a novel approach called Partial Convolution (PConv)[27] has been proposed to extract spatial features more efficiently.

It is designed to reduce both memory accesses and computational redundancy, thus optimizing the execution efficiency of the entire neural network. Compared to conventional convolution, PConv requires only 1/16 of the number of FLOPs and 1/4 of the number of memory accesses.

Based on PConv, researchers also proposed the FasterNet structure[27], which runs much faster than other networks. The FasterNet structure enables the neural network to extract features and perform computations more efficiently during execution, thus improving overall performance.

In this study, the C2f module of YOLOv8's backbone network is redesigned based on partial convolution (PConv). The improved module can effectively reduce the number of parameters and the amount of computation. The module uses PConv to replace the original Conv operation, which has the advantage of improving the spatial feature extraction for some input channels while remaining unchanged for the rest of the channels. This is illustrated in Figure 3. The application of the C2f_Faster module of this paper in the improved backbone network can effectively reduce computational redundancy and the number of parameters.



Note: The packet bottleneck structure is shown by the small figure on the right, which includes 2 PConv and 2 Conv.

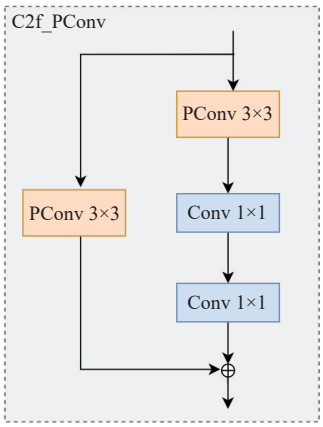
Figure 3 C2_Faster structure diagram

In the deep learning target detection task, the neck is the module or layer between the backbone and head networks. Its role is to process further and fuse the features extracted from the backbone network to improve the target detection performance.

However, neck modules also have some drawbacks. Some neck structures may introduce a bottleneck effect, i.e., the feature dimensions may be constrained or compressed during the feature fusion process, resulting in information loss and model performance degradation. Some complex neck structures can introduce significant computational and memory overhead, increasing the training and inference time of the model and limiting its use in resource-constrained environments. In some cases, the neck module may not be able to fully utilize and convey feature information extracted from the backbone network at different scales and levels, resulting in degraded target detection performance.

To address these shortcomings, this study proposed an improved bottleneck structure for backbone and neck networks, as

shown in Figure 4, to reduce computational and storage overheads, increase flexibility, and improve the balance of information delivery, further enhancing target detection's accuracy and efficiency.



Note: The structure consists of a 3×3 PConv followed by two 1×1 Convs plus a 3×3 PConv residual structure.

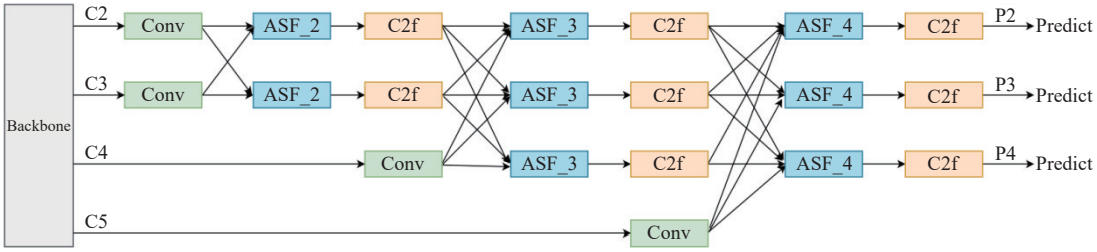
Figure 4 Improved C2f_PConv structure

3.3 AFPN structure

Asymptotic Feature Pyramid Network (AFPN)[28]is a feature pyramid structure for target detection tasks. AFPN solves the problem of feature information loss or degradation in the traditional feature pyramid approach by incrementally fusing the features of non-adjacent hierarchical levels. The main idea is to start from the bottom and avoid large semantic gaps between non-adjacent levels by fusing low-level features and gradually introducing high-level features. AFPN also introduces adaptive spatial fusion operations to solve the multi-target information conflicts that may occur in the feature fusion process.

Specifically, the feature fusion process of AFPN starts from low-level features, gradually fuses deep-level features, and finally fuses the highest-level features. This gradual fusion approach brings the semantic information of features at different levels closer together and reduces the semantic gap between non-adjacent levels. For example, fusing the features of C2 and C3 reduces the semantic gap between them; meanwhile, fusing the features between C3 and C4 reduces the semantic gap between C2 and C4.

By introducing AFPN, the feature pyramid network can better maintain the continuity and consistency of feature information and improve target detection performance. AFPN is of great significance in multi-scale feature coding, and it can effectively solve the problem of the poor effect of feature fusion between non-adjacent layers, as is shown in Figure 5.



Note: The small target detection layer is introduced in the backbone network through the progressive fusion network, the original large target detection layer (P5) is removed, and finally the P2, P3, and P4 layers are retained.

Figure 5 Improved AFPN architecture diagram

In the adaptive spatial fusion process, the adaptive spatial fusion (ASF) mechanism is used to assign different spatial weights to features at different levels, thereby enhancing the importance of key-level features and mitigating the influence of conflicting information from different targets on feature fusion. The ASF mechanism weights features for fusion according to the spatial importance of features at different levels. For those key hierarchical features with high importance, ASF assigns higher weights so that they have more influence in the fusion process. ASF assigns appropriate weights for other hierarchical features based on their effectiveness at a particular location. In this way, ASF emphasizes those features that are more important for the target detection task and improves the model's ability to perceive key features.

At the same time, ASF can also mitigate the effect of conflicting information from different targets on feature fusion. In the target detection task, there may be semantic and morphological differences between different targets, and these differences may lead to the conflict of feature information. By adjusting the fusion of features according to the spatial weights of different targets, the ASF makes the features more balanced and stable in the fusion process and reduces the effect of conflicting information. The adaptive spatial fusion process of different network layers is shown in Figure 6, where it can be seen that the ASF module fuses the features of different network layers through horizontal connectivity, downsampling, and upsampling.

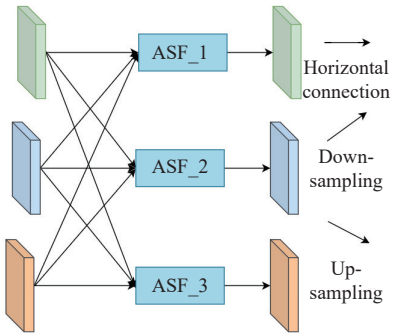


Figure 6 Adaptive spatial fusion

3.4 Parallel branching attention mechanism

In the input citrus images, which are often accompanied by noise, a large amount of redundant information will be generated with the increase of convolution depth, which will result in the loss of some useful information and decrease target detection accuracy. For this reason, this study proposes an improved Parallel Multi-Branch Attention (PMBA) module and chooses to incorporate it into the YOLOv8 network at effective network locations, which can enable the network to more accurately localize and identify regions of interest.

The attention module is divided into the Channel Attention Module (CAM) and Spatial Attention Module (SAM). The CAM is used to compute the weights corresponding to each channel of the

input features, and its structure improves the channel module of the NAM[29] attention mechanism. The SAM is used to compute the weights corresponding to each feature point, and its structure adopts the spatial module of the CBAM[30] attention mechanism.

The NAM attention mechanism improves the performance of the attention mechanism by introducing a contribution factor for the weights. It uses a batch-normalized scale factor to represent the importance of the weights, effectively avoiding the fully connected and convolutional layers used by the ECA[31] and CBAM modules. Specifically, NAM adopts the modular integration of CBAM with a redesign of the channel attention submodule. The feature map F is multiplied by the BN scaling factor and weights, and finally, after a sigmoid activation function, the output channel features are obtained. The channel attention submodule adopts the scaling factor in the batch normalization and uses the scaling factor to compute the channel variance to measure the importance of the weights, as shown in Equation (1):

$$B_{\text{out}} = BN(B_{\text{in}}) = \gamma \frac{B_{in} - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} + \beta \quad (1)$$

where, B_{in} is the input feature; B_{out} is the output feature; γ and β are the parameters of trainable affine transformation; μ_β and σ_β are the mean and standard deviation of the small batch, respectively; and ε is the error. The channel module in NAM is shown in Figure 7, where, F is the input feature; M_c is the output feature; BN denotes the normalization; "weight" denotes the weights, which is expressed as $W_i = \gamma_i / \sum_{j=0} \gamma_j$; γ_i denotes the scale factor of each channel; and σ denotes the sigmoid activation function, which is given in Equation (2):

$$M_c = \sigma(W_i(BN(F))) \quad (2)$$

The structure of the SAM module is shown in Figure 8. Firstly, global maximum pooling and average pooling operations are performed on the input features F to obtain two features $F_{\max}$ and F_{avg}. Then, $F_{\max}$ and F_{avg} are spliced together and normalized by convolution operation and sigmoid activation function to obtain the two-dimensional spatial attention weights M_s, with the formula as in Equation (3):

$$M_s(F) = \sigma(\text{Conv}(\text{AvgPool}(F) \oplus \text{MaxPool}(F))) \quad (3)$$

where, AvgPool is the average pooling function over space, MaxPool is the maximum pooling function over space, Conv is the convolution function,⊕is the feature merging operation, and σ is the sigmoid activation function; F is the input feature and M_s is the output feature.

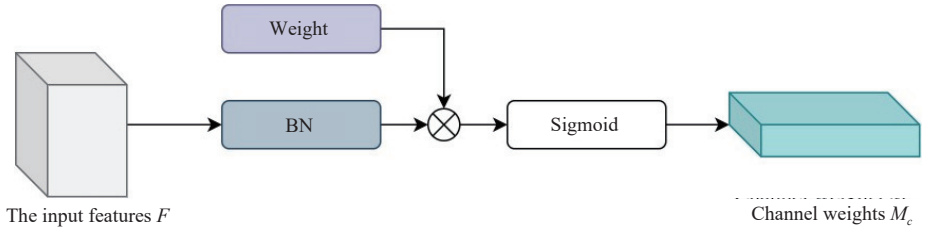


Figure 7 Structure of channel attention module

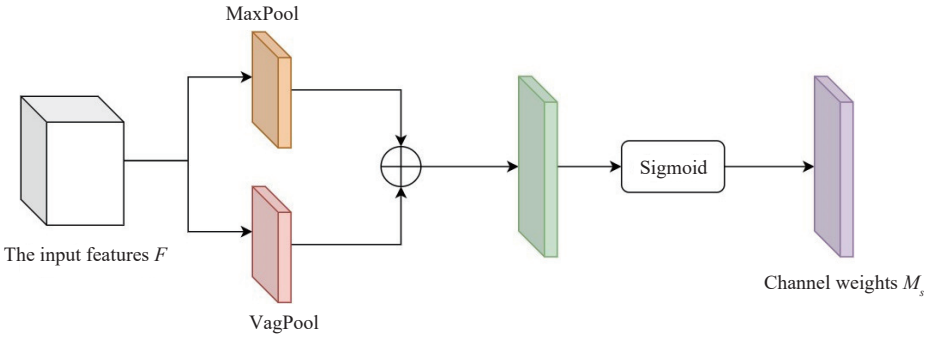


Figure 8 Structure of spatial attention module

Improvements are made to solve the problem of the tandem connection of the channel and spatial attention modules in ordinary attention mechanisms (e.g., CBAM and NAM), and the possible loss of channel and spatial information during training. The improved formula is shown in Equation (4) as follows: in the improved attention module, the input features are first processed with channel attention to obtain the channel attention-weighted features. Then, spatial attention processing is performed on the features after channel attention processing to obtain the final output features. With this improvement, the channel and spatial information can be better preserved when the network depth is deepened, and the performance of the attention mechanism can be improved.

$$F' = (M_c(F_1) \otimes F_1) \oplus (M_s(F_2) \otimes F_2) \quad (4)$$

where, F_1 and F_2 are channel features and spatial features, respectively; F' is the output feature; ⊗ is the feature multiplication operation; and ⊕ is the feature merging operation. The improved parallel multi-branch attention module (PMBA) is shown in Figure 9.

3.5 Loss function

The traditional YOLOv8 uses CIoU[32] as the loss function for regression loss, and the CIoU loss function is very sensitive to changes in the size and shape of the target frame. This means that the model may not perform stably when dealing with targets of different sizes and shapes, and for small targets, since their IoU values tend to be low, the advantage of CIoU is not obvious, which may lead to poor detection of small targets by the model.

Since the training data inevitably contains low-quality samples, geometric factors such as distance, aspect ratio, etc. will aggravate the punishment of low-quality samples, thus reducing the model's generalization performance. To solve the above problems, this study introduces Wise-IoU[33]to improve the original loss function.

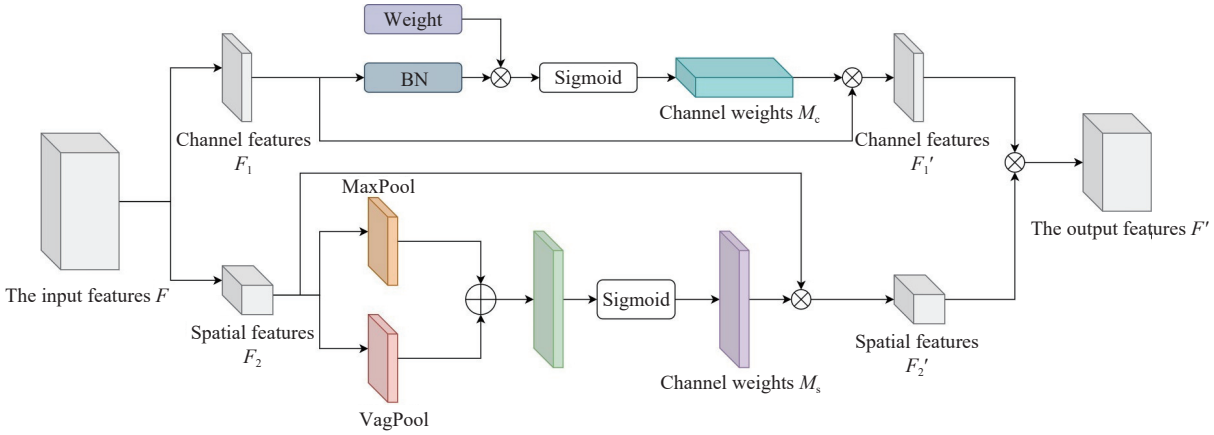


Figure 9 Structure of parallel multi-branch attention module

The marked frame is denoted as $\vec{B}=[x,y,w,h]$, and the target frame is denoted as $\vec{B}_{gt}=[x_{gt},y_{gt},w_{gt},h_{gt}]$. In the following figure, (x,y)denotes the center coordinates of the marked frame and(x_{gt},y_{gt})denotes the center coordinates of the target frame. IoU is used to measure the degree of overlap between the predicted frame and the real frame in the target detection task, and its loss is defined in Equation (5):

$$\begin{aligned}&\mathrm{IoU}=\frac{W_i\times H_i}{w\times h+w_{gt}\times h_{gt}-W_i\times H_i}\\&L_{\mathrm{IoU}}=1-\mathrm{IoU}\end{aligned}\tag{5}$$

In the smallest enclosing box (green) and the central points' connection (blue), the area of the union is $w\times h+w_{gt}\times h_{gt}-W_i\times H_i$; $w\times h$ and $w_{gt}\times h_{gt}$ denote the area of the width and height of the labeled and target frames, respectively; and W_i and H_i denote the width and height of the intersection of the labeled and target frames, respectively. IoU denotes the loss intersection and merger ratio of the labeled frame to the target frame, as shown in Figure 10.

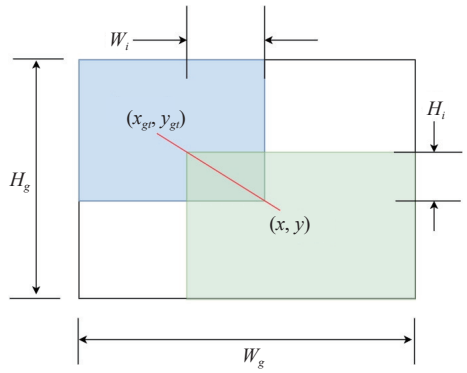


Figure 10 Loss of intersection ratio

Wise-IoU can better focus on the target and increase the detection frame regression accuracy. Its formula is shown in Equation (6):

$$\begin{aligned}&L_{\mathrm{WIoUv1}}=R_{\mathrm{WIoU}}(1-\mathrm{IoU})\\&R_{\mathrm{WIoU}}=\exp\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)}\end{aligned}\tag{6}$$

where, W_g and H_g are the sizes of the smallest enclosing frames of the labeled frame and the target frame. R_{WIoU} refers to a weighted form of IoU (intersection and concurrency ratio), specifically distance-attention-weighted IoU, which is used to significantly amplify the IoU values of ordinary quality anchor frames. It optimizes the detection performance of the model by introducing the concept of distance attention, which enables the model to pay more attention to those anchor frames that are moderately distant from the target frame and have a better fit during the training process. L_{WIoUv1} is the first version of the Wise-IoU loss function, which significantly amplifies the IoU values of normal-quality anchor frames and reduces the attention of high-quality anchor frames by introducing Rwiou.

Wise-IoU v3 defines the outlier to describe the quality of the anchor frame. The outlier of the anchor frame is denoted by the ratio of L_{IoU} and $\overline{L_{\mathrm{IoU}}}$: $\beta=\frac{L_{\mathrm{IoU}}}{\overline{L_{\mathrm{IoU}}}}\in[0,+\infty)$, where a small outlier implies that the anchor frame is of high quality, and it is assigned a small gradient gain to bring the bounding box back into focus on the anchor frame of normal quality. A non-monotonic focusing factor β is constructed using $r$and applied to Wise-IoU v1. The formula for implementing Wise-IoU v3 is shown in Equation (7):

$$\begin{aligned}&L_{\mathrm{WIoUv3}}=rL_{\mathrm{WIoUv1}}\\&r=\frac{\beta}{\delta\alpha^{\beta-\delta}}\end{aligned}\tag{7}$$

where$r$ =1 when $\beta=\delta$, at which point the degradation is Wise-IoU v1. If the degree of outliers of the anchor frame satisfies $\beta=C$ (C is a constant value), the anchor frame will receive the highest gradient gain. L_{WIoUv3} is the third version of the Wise-IoU loss function, which further introduces a dynamic non-monotonic focusing mechanism on the basis of L_{WIoUv1}, which dynamically adjusts the gradient gain according to the outliers of the anchor frames, enabling the model to focus on high-quality anchor frames in the early stage of the training, while focusing more on the ordinary-quality anchor frames in the late stage of the training, thus optimizing the model's detection effect.

4 Experiment and analysis

4.1 Data collection

In this experiment, the datasets of citrus fruits in a citrus orchard environment were generated. This study's citrus images were taken in a Chu-style agricultural plantation in Yunnan Province. The shooting device was an iPhone (Apple Inc, USA), the shooting distance was 1-3 cm, automatic exposure, the resolution pixel of the near image was 1984×1448, the resolution pixel of the far image was 2840×1563, and it was saved in *.JPG format.

The datasets were taken around November 2021, and a total of 1315 raw images were collected, all taken under natural daylight conditions, including whole multi-fruit images and partial multi-fruit images taken under different weather conditions, such as

sunny, cloudy, and after rain. In order to reduce the number of duplicate images as well as the interference of fruitless images on model training, a manual screening method was used to clean the data from the captured raw images, and a total of 1275 raw images containing citrus fruits were finally obtained.

This dataset contains four types of disturbances, including overlap, occlusion, light-dark variation, and distance variation, which were designed to best reproduce the real condition of the human eye when observing citrus fruits in a natural environment. These disturbances are common in real-world picking scenarios, so the datasets can provide challenging data for citrus fruit recognition and related tasks.

Some of the samples in the datasets are shown in Figure 11, demonstrating citrus images from different viewpoints and under different environmental conditions. With such a dataset, it is expected to improve the robustness and generalization ability of the model in real scenarios, which will make an important contribution to the research and application of citrus fruit recognition and agricultural automation technology.

a. Single fruit b. Multi-fruit shade c. Dense fruit

d. Multi-fruit overlap e. Back-lighting f. Contre-jour

Figure 11 Diversity dataset demonstration

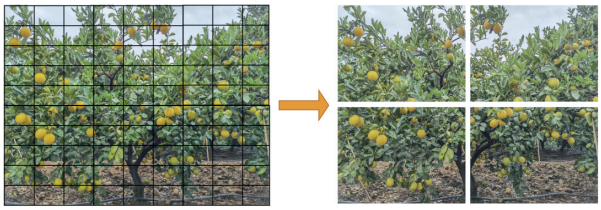
4.2 Data enhancement

Since the large telephoto image contains complex information and it is not easy to distinguish the characteristics of the small citrus fruit, this study manually selected the telephoto image for further data enhancement processing. First, the telephoto image was segmented, which has the advantage of highlighting small targets. Second, the dataset was expanded using the data enhancement method.

Data enhancement is a technique commonly used in machine learning and deep learning, which aims to increase the variety and amount of data by transforming and expanding the original data to improve the generalization ability and robustness of the model. In this study, 1275 raw data were collected, including far-view and near-view images, and the far-view images were divided into four parts using the image segmentation operation.

The far-view and near-view images provide different perspectives and scales, and their presence in the dataset can increase the diversity of the data. By performing the segmentation operation on the telephoto image, the original image was divided into 4 pieces, each of which represents a localized region in the telephoto image, as shown in Figure 12. This segmentation operation can effectively increase the amount of data and introduce different spatial context information.

Data augmentation aims to enable the model to better adapt to different scenarios and changes and reduce the model's dependence on specific samples in the dataset. With the segmentation operation, more training samples are generated, each containing a different region of the telepresence map, which helps the model learn the representational capabilities of different local features.

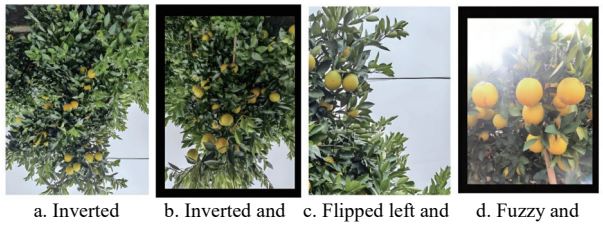


Original image Split image

Figure 12 Partial datasets segmentation demonstration: splitting dense images into 4 parts

In the model training process, the segmented image is used in combination with other original data through rotation, mirror flipping, enhancement, scaling, brightness adjustment, and blurring. Such data enhancement operations can further expand the training dataset and expose the model to more different image variations during the learning process, thus improving its generalization ability and reducing the risk of over-fitting.

It should be noted, however, that in deep learning model training, data augmentation cannot dominate the entire training process but should be auxiliary. Augmenting each image can destroy the features of the original data, so a certain amount of randomness and probability needs to be added to the augmentation operation. This means that in each training iteration, only some of the images are subjected to data enhancement operations, while the others are left as they are. These enhancement means are randomized individually or superimposed to enhance the original image, such as rotating 20% of the image left and right, mirroring 50% of the image, scaling the image between 80% and 95%, multiplying each pixel by a number between 0.5-1.5 to darken or lighten the image, and blurring the image with one of Gaussian, Mean, and Median. Some examples of data enhancement are shown in Figure 13.



a. Inverted b. Inverted and scaled c. Flipped left and right d. Fuzzy and scaled

e. Brightness f. Brightness enhanced g. Inverted and blurred

Figure 13 Example data enhancement diagram

By choosing the probabilities and operations of data augmentation wisely, the model can be helped to learn the commonalities and patterns of the data better and thus perform better on unseen data. Data augmentation is an effective technique to improve the generalization performance of the model, but care must be taken to balance the degree of augmentation when applying it so as not to cause excessive distortion of the original data.

There are 7566 images of citrus fruits in the datasets, out of

which 93 561 citrus fruits were captured and divided into training, validation, and test sets in the ratio of 8:1:1. The training set consists of 6054 images containing 74 688 citrus fruits, the validation set consists of 756 images containing 9270 citrus fruits, and the remaining 756 images containing 9603 citrus fruits constitute the test set. In addition, about 40% of the individual citrus images in the dataset fall into the category of small target objects, about 20% of the citrus images are large target objects, and nearly 50% of the images contain more than 3 citrus fruits. All datasets were stored in JPG format. Table 1 shows the distribution of the datasets.

Table 1 Division of the datasets

Dataset classification	Delineation	Proportion	Number of pictures	Number of fruits
Datasets	Training set	80%	6054	74 688
	Validation set	10%	756	9270
	Test set	10%	756	9603
Total		100%	7566	93 561

4.3 Pre-training

The operating system used for training in this study was Ubuntu 20.04, the CPU model is Intel(R) Xeon(R) Platinum 8255C CPU, the GPU model was RTX 3080, and the frameworks were Pytorch 1.10.0 and CUDA version 11.3 framework.

4.4 Evaluation criteria

In the field of target detection, the main metrics used to evaluate the performance of the network are Mean Average Precision (mAP) and Average Precision (AP). The comprehensive consideration can evaluate the effectiveness of the model and calculate the average value of the predicted area of each category target under the recall rate R and precision rate P, i.e.,

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8}$$

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{9}$$

where, TP refers to correctly predicted as positive cases; FP refers to incorrectly predicted as positive cases; and FN refers to incorrectly predicted as negative cases.

In the field of object detection, precision (P) refers to the proportion of targets that are correctly predicted as citrus fruits among all targets predicted to be in the citrus category. Recall (R) represents the proportion of all targets that were actually citrus fruit that were correctly predicted to be in that category. These two metrics combine the model's accuracy and recall for targets and can effectively evaluate the performance of the target recognition model.

Neither the precision rate nor the recall rate provides a complete picture of the model's performance, so the $F1$ score is used as a compromise between the two, as defined in Equation (10):

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2 \times P \times R}{P + R} \tag{10}$$

Mean Accuracy Rate (mAP) is the integral of the precision rate over the recall rate in the precision-recall curve (P-R curve) in the recall range [0,1], which is calculated as shown in Equation (11). mAP represents the average of the APs of the different categories of targets, and for the citrus dataset in this study, N=1 since there is only one category. By calculating the mAP, it is able to measure the overall effectiveness of citrus detection objectively.

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R \tag{11}$$

$$\mathrm{mAP} = \frac{\sum^{N} \int_0^1 P(R)\mathrm{d}R}{N} \tag{12}$$

4.5 Results and comparison

In order to visualize the effectiveness of the improved YOLOv8s algorithm, the test images were detected by comparing two sets of dense citrus images using the traditional YOLOv8 and the improved FAP-YOLOv8 algorithms in this study, respectively, and both models use the standard size 's' (denoting the model size). From the comparison graphs, the accuracy of the traditional YOLOv8 detection is generally low, while the improved FAP-YOLOv8 detection accuracy is improved. The two sets of comparison images show that the traditional YOLOv8 has leakage detection, while the improved algorithm pays more attention to the details of dense, occluded, and poorly lit fruits, improving the leakage detection of complex and occluded fruits. As can be seen from the second group of comparison pictures, when the background is complex and the small target fruits have high similarity and are difficult to distinguish, the traditional model appears to miss detection, while the improved algorithm shows stronger recognition ability. The result comparison is shown in Figure 14.

YOLOv8 detection results FAP-YOLOv8 detection results

Note: Blue ellipses indicate leakage in the original model.

Figure 14 Comparison of results before and after improvement

4.6 Ablation experiments

Based on the original datasets, to verify whether the improved parallel multi-branch attention mechanism module (PMBA) in this study is effective for training, it was compared with the ECA attention mechanism module, NAM attention mechanism, and CBAM attention mechanism module, respectively, in a comparison experiment. The experiments were conducted by introducing ECA, NAM, CBAM, and PMBA after the YOLOv8 neck network. The purpose was to further explore the differences between the improved parallel multi-branch structure and the parallel structure of the CBAM attention mechanism, as well as the serial structure of the NAM channel-only module and the CBAM spatial module-only. These are denoted as PMBA, CBAM-B, and NCBAM-S in this

study, respectively, and are added behind the neck network of YOLOv8 under the same conditions.

The results are listed in Table 2, where the introduction of the attention mechanism in the same position as the traditional algorithm all improved, and where ECA and NAM improved over the benchmark, contributing 0.9% and 1%, respectively, indicating that the channel-only attention module added to the necking network is able to strengthen the weight share of the input features to each channel. The CBAM attention mechanism improved over the benchmark by a 1.2% improvement of the parallel structure CBAM-B attention mechanism, and both the channel-only NAM module and the spatial module-only serial structure NCABAM-S improved by one percentage point.

Table 2 Comparison of various attention mechanisms

Model	Model size/M	mAP@0.5/%	Precision/M	GFLOPS/G
YOLOv8	22.6	92.5	11.1	28.4
NAM	22.5	93.5	10.6	28.5
CBAM	22.2	93.7	10.7	28.5
ECA	22.5	93.4	10.6	28.4
CBAM-B	22.5	93.5	10.6	28.5
NCBAM-S	22.5	93.5	10.6	28.5
PMBA	23.1	93.8	11.0	28.8

The improved parallel multi-branch structure PMBA, on the other hand, improved by 1.3 percentage points, indicating that the improved parallel multi-branch structure strengthened the aggregated channel characteristics and improved the ability to perceive the target in the spatial dimension, which enabled a more balanced attention to the target region and improved the ability of the overall PMBA attention mechanism in the aggregated network.

In order to verify the need for image segmentation and the effectiveness of the improved model proposed in this study, this study firstly compared the detection effectiveness of the original dataset and the dataset processed by image segmentation in the same experimental environment. Then, the improved method was applied on the data-enhanced dataset and further experiments were implemented. Meanwhile, the optimized YOLOv8 network was tested for comparison while ensuring that the training parameters were consistent with the dataset. The experimental results are detailed in Table 3.

Table 3 Ablation experiments with improved strategies

Split Image	C2f_ Faster	AFPN	PMBA	WIoU	Model size/ M	mAP@ 0.5/%	Precision/ M	FLOPs/ G	Rise/ %
-	-	-	-	-	22.6	92.5	11.1	28.4	-
√	-	-	-	-	22.6	93.5	11.1	28.4	1.0
√	√	-	-	-	19.6	94.2	9.2	24.4	1.7
√	-	√	-	-	14.0	95.1	6.7	27.1	2.6
√	-	-	√	-	22.5	94.5	11.1	28.5	2.0
√	√	√	-	-	11.1	95.6	5.0	22.8	3.1
√	-	√	√	-	14.0	95.4	6.7	27.1	2.9
√	√	-	√	-	19.6	94.6	9.6	24.2	2.2
√	√	√	√	-	11.1	97.1	5.0	22.9	4.6
√	√	√	√	√	11.1	97.2	5.0	22.9	4.7

In the table, C2f_Faster, AFPN, PMBA, and WIoU improvement denote the incorporation of the improvement scheme into the traditional YOLOv8 network, respectively. From Table 3, it can be analyzed that there was an improvement of 1.0 percentage points after the segmentation was done only for the large image in the far view, which indicates that the network ignored some small and occluded targets when dealing with complex images. In addition, the segmented image maintained the background of the original image, but the model reduced the loss of the target of interest due to the complex background.

First, the use of the improved C2f_Faster to replace the backbone network provided a 1.7 percentage point improvement for the model, which resulted in higher accuracy of model recognition and a significant reduction in model size, number of parameters, and computation. Second, in the improvement of neck network, the overall improvement of 2.6% using the improved AFPN structure indicates that the AFPN effectively solved the problem of feature information loss or degradation in the traditional feature pyramid method by improving the fusion between feature layers and resolving the information conflict by using adaptive spatial fusion. And to a certain extent, it alleviated the problem of missed detection due to blurred images with complex backgrounds. Finally, after the introduction of the PMBA attention mechanism, the model improved by 2% over the benchmark mAP, which indicates that the addition of the PMBA attention module enabled the network to pay more attention to the fruits on the fruiting, which led to the improvement of the performance of the whole model.

With the combined form of the appeal method, it can be seen that the combination of improved backbone network C2f_Faster and improved neck network AFPN structure improved the model by 3.1% over the benchmark, reduced the model size by 51% over the benchmark, reduced the number of parameters by 55%, and reduced the amount of computation by 20%. It can be clearly seen that the combination of AFPN structure and PMBA made the model teach the benchmark to improve the model by 2.9%. The combination of improved backbone network C2f_Faster and PMBA improved the model by 2.1 percentage points. The combination of backbone network C2f_Faster, neck network AFPN structure, and PMBA attention mechanism improved the model by 4.6%. The addition of WIoU using the above improvements finally contributed to the model's mAP with a performance improvement over the baseline model of 4.7%, proving the effectiveness of the four improvement methods.

Meanwhile, Figure 15 shows the change in the loss curves before and after the improvement of learning process of YOLOv8. At the inflection point, the WIoU loss curve is lower than the CIoU loss curve, indicating that the WIoU loss value is lower and the curve is smoother. Meanwhile, after the inflection point, the loss values of both loss functions tend to decrease slowly after 40 epochs and finally become smooth, and the WIoU loss can converge the network faster and more smoothly during whole training process.

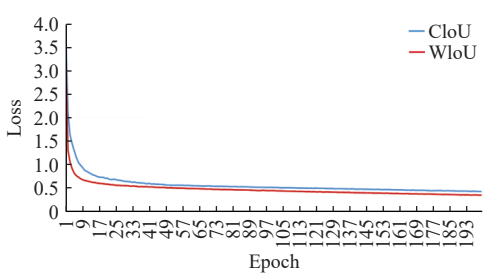


Figure 15 Curves of different loss functions

Figure 16 shows the comparison of the F1 value between improved YOLOv8 and original YOLOv8. Figure 17 shows the relationship between the improved YOLOv8 and the original algorithm in terms of accuracy, recall, and mean accuracy (mAP).

From the figure, it can be seen that in the graph of the relationship between the F1 value and confidence, the improved YOLOv8 model is larger than the original YOLOv8 value, and the area enclosed by F1 and confidence is larger. In the plot of accuracy vs. recall, the improved YOLOv8 curve is always above the original YOLOv8, and the curve fluctuates little. In the plot of mean accuracy (mAP), the improved YOLOv8 curve is above the original YOLOv8 and tends to rise steadily after 25 epochs.

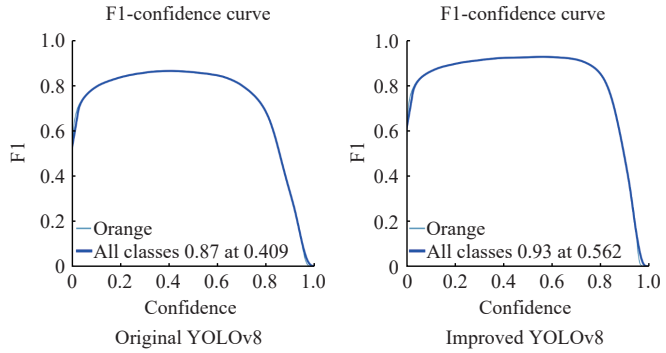


Figure 16 F1 value before and after improvement

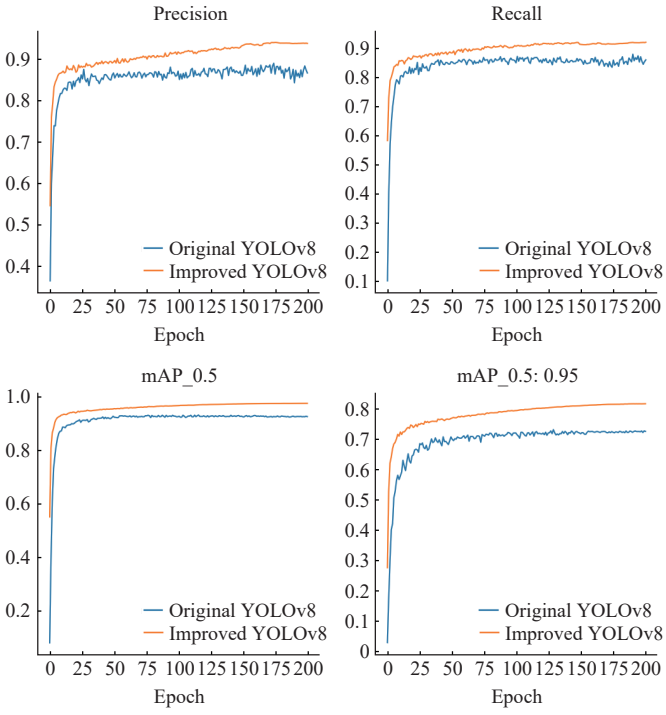
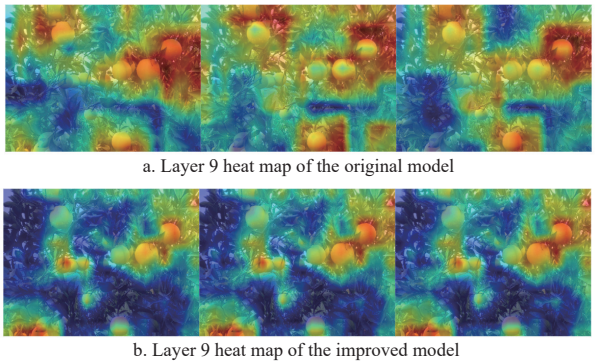


Figure 17 Curves of accuracy, recall, and average accuracy before and after improvement

4.7 Visualization experiments

In Figure 18, we compare the results of the heat map of layer 9 before and after the improvement. From the visualization graph, we can see that the results of the original model cover a larger area, including the leaves and branches around the fruits, compared to the improved YOLOv8 trunk model. This is due to the fact that the original trunk model focuses too much on the objects with obvious features in the image, resulting in insufficient attention to the fruit target. While the improved YOLOv8 trunk consists of Faster Block, it can be seen that the heat map pays more attention to the fruit targets compared to the original model trunk, which improves the performance and accuracy of target detection.

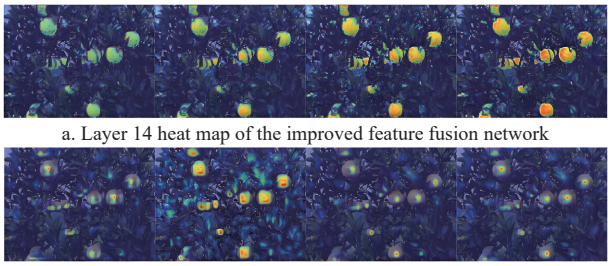
The results of the visualization process after improving the feature pyramid (AFPN) and adding the parallel branch attention (PMBA) are shown in Figure 19. The enhanced feature fusion part pays more attention to citrus fruits and less attention to other branches and foliage backgrounds, so most of the citrus fruits are covered by the heat map in the enhanced feature pyramid fusion network. Subsequently, the inclusion of the parallel branch attention mechanism further enhances the concentration of the entire network on the region of interest, with the heat map focusing more on the top of the fruits. This suggests that the improved feature pyramid and parallel branching attention approach not only focuses on the primary features but also gives appropriate attention weights to the secondary features, thus more information can be extracted, allowing more objects to be detected more accurately in recognition. These improvements help to improve the performance and accuracy of object detection, leading to better results in the citrus fruit detection task.



a. Layer 9 heat map of the original model

b. Layer 9 heat map of the improved model

Figure 18 Trunk heat map visualization



a. Layer 14 heat map of the improved feature fusion network

b. Layer 30 heat map of a network with added parallel branching attention mechanism

Figure 19 Improved feature fusion heat map visualization

4.8 Comparison experiments

In order to compare the superiority of the improved FAP-YOLOv8 target detection algorithm in this study with other algorithms, this study conducted comparison experiments with various advanced target detection algorithms: Faster R-CNN[34], CenterNet[35], YOLOv5[36], YOLOx[37], YOLOv7[26], and the original YOLOv8[38] model, in which YOLOv5 and YOLO8 are compared with the standard models YOLOv5s as well as YOLOv8s. As can be seen from Table 4, the improved FAP-YOLOv8 reached 97.2% higher than Faster R-CNN, CenterNet, YOLOv5, YOLOx-s, YOLOv7, and YOLOv8 algorithms in terms of detection accuracy by 19.2%, 7.4%, 5.1%, 4.9%, 5.2%, and 4.7%, respectively. In terms of model size, improved FAP-YOLOv8 was 97.1 M, 113.8 M, 3.3 M, 22.8 M, 61.3 M, and 11.5 M lower than Faster R-CNN, CenterNet, YOLOv5, YOLOx-s, YOLOv7, and YOLOv8, respectively. The size of the improved FAP-YOLOv8 model was smaller than that of the Faster R-CNN, CenterNet, YOLOx-s, YOLOv7, and the original YOLOv8 model in terms of the number of parameters by 131. 7 M, 31.7 M, 49.2 M, 30.4 M, and 6.1 M, respectively; and the size of the improved YOLOv8 model was

larger in terms of computational complexity than that of the Faster R-CNN, CenterNet, YOLOx-s, and YOLOv7 models. CNN, CenterNet, YOLOx-s, YOLOv7, and the original model algorithm were 346.8 G, 47.3 G, 133.1 G, 82.2 G, and 5.7 G lower, respectively. Although the FPS value of the improved FAP-YOLOv8 algorithm in this study was relatively low, it outperformed the other algorithms of other classes in terms of model size, detection accuracy, and computation volume, respectively, proving the effectiveness of the improved algorithm. This shows that the FAP-YOLOv8 algorithm in this paper is suitable for later deployment in edge devices.

Table 4 Comparison of the various algorithms

Model	Model size/M	mAP@0.5/%	Precision/M	Gflops/G	FPS/Hz·s^{-1}
Faster R-CNN	108.2	78.0	136.7	369.7	24.5
CenterNet	124.9	89.8	36.7	70.2	76.1
YOLOv5	14.4	92.1	7.0	15.9	78.7
YOLOx-s	34.3	92.3	54.2	156.0	61.9
YOLOv7	74.8	92.0	35.4	105.1	46.4
YOLOv8	22.6	92.5	11.1	28.6	78.7
FAP-YOLOv8	11.5	97.2	5.0	22.9	46.5

5 Conclusions

In this study, based on the study of YOLOv8 model, a FAP-YOLOv8 model was proposed, which is mainly used to solve the problem of citrus detection in citrus orchards, high density of fruits, and overlapping fruit tree branches and leaves. There are four main ways to improve the model. The first is to replace the traditional C2f module with a lightweight FasterNet structure module. The second is to increase the small target detection layer while removing the large target detection layer and keeping three different scale detection layers to realize multi-scale feature fusion, which can alleviate the problem that the original detection layer of YOLOv8 cannot adapt to small target objects. At the same time, the advanced feature pyramid network (AFPN) is introduced to realize multi-scale feature fusion to bring the semantic information of features in different layers closer together and reduce the semantic gap between non-adjacent layers. Again, by mimicking the human visual attention learning mechanism, PMBA, an enhanced attention module with parallel channels and spatial dimensions, is used to complete the restructuring and optimization of the feature extraction and detection parts of the neck and head of the YOLOv8 feature fusion network. Finally, by replacing the loss function of the original model, the use of Wise-IoU to replace the original CIoU can better focus on the target, increase the detection frame regression accuracy, and the network can converge faster and more smoothly during the whole training process.

The FAP-YOLOv8 model proposed in this study achieved excellent results in several metrics when comparing six target detectors on the test dataset. The mAP@0.5 value of FAP-YOLOv8 was 97.2%, and the detection accuracy was 19.2%, 7.4%, 5.1%, 4.9%, 5.2%, and 4.7% higher than that of Faster R-CNN, CenterNet, YOLOv5, YOLOx-s, YOLOv7, and YOLOv8 models, respectively. In terms of the number of parameters index, the improved model was only 5M, which was about 55.45% lower than the original model in terms of the number of parameters and six percentage points higher than YOLOv8 in terms of F1 index, which well balances the detection accuracy and completeness, and is a model with excellent detection performance. In terms of detection speed and lightness, the FPS value of 46.51 fps average detection speed of the FAP-YOLOv8 model was lower than that of 78.7 fps of YOLOv8. The number of parameters was reduced by nearly 5.7 M. The computational volume was saved by 5.7 G. The improved model proposed in this study achieved a certain balance of accuracy, speed, and lightweight deployment. The algorithm is more suitable for dense scenes of citrus target detection and can be used for target detection of citrus fruits in real orchard environments. In future work, other modular structures of the model could be further improved. For example, the issue of leaf occlusion can be further addressed, context learning using transformer structures can be leveraged, and the model's adaptability to different scenarios can be enhanced, thereby laying the foundation for automated harvesting.

Acknowledgements

First of all, I would like to express my heartfelt thanks to my supervisor, Yang Yi, who gave me selfless guidance and support from the selection of the thesis topic to the final draft. Under his careful guidance, I overcame all the difficulties and completed the research work of the thesis. Secondly, I would like to thank my labmates for their help to make the experiment go smoothly. Finally, I would like to thank the Xinping Planting Base of Yunnan Province (Chu's Agricultural Co.) for providing the experimental data. This work was financially supported by the Yunnan Provincial Major Science and Technology Special Project: Research and Development and Application Demonstration of Key Technology for Digitization of Cloud Fruit (Grant No. 202002AE09001002).

[References]

[1] Liu S Y. Artificial intelligence (AI) in agriculture. IT Professional, 2020; 22(3): 14–15.

[2] Tang Y C, Chen M Y, Wang C L, Luo L F, Li J H, Lian G P, et al. Recognition and localization methods for vision-based fruit picking robots: A review. Frontiers in Plant Science, 2020; 11: 510.

[3] Díaz I, Mazza S M, Combarro E F, Giménez L I, Gaiad J E. Machine learning applied to the prediction of citrus production. Spanish Journal of Agricultural Research, 2017; 15(2). doi: 10.5424/sjar/2017152-9090.

[4] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015; 521(7553): 436–444.

[5] Sa I, Ge Z Y, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A fruit detection system using deep neural networks. Sensors, 2016; 16(8): 1222.

[6] Bargoti S, Underwood J. Deep fruit detection in orchards. In: 2017 IEEE international conference on robotics and automation (ICRA), Singapore: IEEE, 2017; pp.3626–3633.

[7] Bargoti S, Underwood J P. Image segmentation for fruit detection and yield estimation in apple orchards. Journal of Field Robotics, 2017; 34(6): 1039–1060.

[8] Koirala A, Walsh K B, Wang Z, McCarthy C. Deep learning - Method overview and review of use for fruit detection and yield estimation. Computers and Electronics in Agriculture, 2019; 162: 219–234.

[9] Gao F F, Fu L S, Zhang X, Majeed Y, Li R, Karkee M, et al. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. Computers and Electronics in Agriculture, 2020; 176: 105634.

[10] Kukreja V, Dhiman P. A Deep Neural Network based disease detection scheme for Citrus fruits. In: 2020 International conference on smart electronics and communication (ICOSEC), Trichy, India: IEEE, 2020; pp.97–101.

[11] Horng G J, Liu M X, Chen C C. The smart image recognition mechanism for crop harvesting system in intelligent agriculture. IEEE Sensors Journal, 2019; 20(5): 2766–2781.

[12] Liu Z S, Xiang X Y, Qin J H, Tan Y, Zhang Q, Xiong N N. Image recognition of citrus diseases based on deep learning. CMC-Computers Materials & Continua, 2021; 66(1): 457–466.

[13] Qiu W J, Ye J, Hu L Q, Yang J, Li Q L, Mo J Y, et al. Distilled-MobileNet Model of convolutional neural network simplified structure for plant

disease recognition. Smart Agriculture, 2021; 3(1): 109–117.

[14] Liu Y P, Yang C H, Ling H, Mabu S, Kuremoto T. A visual system of citrus picking robot using convolutional neural networks. In: 2018 5th international conference on systems and informatics (ICSAI), Nanjing, China: IEEE, 2018; pp.344–349.

[15] Lu J, Hu X W. Detecting green citrus fruit on trees in low light and complex background based on MSER and HCA. Transactions of the CSAE, 2017; 33(19): 196–201. (in Chinese)

[16] Bi S, Gao F, Chen J W, Zhang L. Detection method of citrus based on deep convolution neural network. Transactions of the CSAM, 2019; 50(5): 181–186. (in Chinese)

[17] Zhang W L, Wang J Q, Liu Y X, Chen K Z, Li H B, Duan Y L, et al. Deep-learning-based in-field citrus fruit detection and tracking. Horticulture Research, 2022; 9: uhac003.

[18] Zhuang J J, Luo S M, Hou C J, Tang Y, He Y, Xue X Y. Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. Computers and Electronics in Agriculture, 2018; 152: 64–73.

[19] Lin G C, Tang Y C, Zou X J, Li J H, Xiong J T. In-field citrus detection and localisation based on RGB-D image analysis. Biosystems Engineering, 2019; 186: 34–44.

[20] Chen J Y, Liu H, Zhang Y T, Zhang D K, Ouyang H K, Chen X Y. A multiscale lightweight and efficient model based on YOLOv7: Applied to citrus orchard. Plants, 2022; 11(23): 3260.

[21] Lyu S L, Li R Y, Zhao Y W, Li Z, Fan R J, Liu S Y. Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system. Sensors, 2022; 22(2): 576.

[22] Yang H W, Liu Y Z, Wang S W, Qu H X, Li N, Wu J, et al. Improved apple fruit target recognition method based on YOLOv7 model. Agriculture, 2023; 13(7): 1278.

[23] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, 2016; pp.779–788.

[24] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv: 1804.02767, 2018. doi: 10.48550/arxiv.1804.02767.

[25] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934, 2020. doi: 10.48550/arXiv.2004.10934.

[26] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada: IEEE, 2023; pp.7464–7475.

[27] Chen J R, Kao S-H, He H, Zhuo W P, Wen S, Lee C-H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada: IEEE, 2023; pp.12021–12031.

[28] Yang G Y, Lei J, Zhu Z K, Cheng S Y, Feng Z L, Liang R H. AFPN: Asymptotic feature pyramid network for object detection. In: 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, Oahu, HI, USA: IEEE, 2023; 2184–2189.

[29] Liu Y C, Shao Z R, Teng Y Y, Hoffmann N. NAM: Normalization-based attention module. arXiv preprint arXiv: 2111.12419, 2021; doi: 10.48550/arXiv.2111.12419.

[30] Woo S, Park J, Lee J-Y, Kweon I S. Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), Munich, Germany: Springer, 2018; doi: 10.1007/978-3-030-01234-2_1.

[31] Wang Q L, Wu B G, Zhu P F, Li P H, Zuo W M, Hu Q H. ECA-Net: Efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, 2020; pp.11534–11542.

[32] Zheng Z H, Wang P, Liu W, Li J Z, Ye R G, Ren D W. Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence, USA: AAAI Press, 2020; pp.12993–13000. doi: 10.1609/aaai.v34i07.6999.

[33] Tong Z J, Chen Y H, Xu Z W, Yu R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. arXiv preprint arxiv: 2301.10051, doi: 10.48550/arXiv.2301.10051.

[34] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017; 39(6): 1137–1149.

[35] Duan K W, Bai S, Xie L X, Qi H G, Huang Q M, Tian Q. Centernet: Keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, 2019; 6568–6577.

[36] Ultralytics/yolov5. 2021; Available: https://github.com/ ultralytics/yolov5. Accessed on [2023-04-25].

[37] Ge Z, Liu S T, Wang F, Li Z M, Sun J. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv: 2107.08430, 2021; doi: 10.48550/arXiv.2107.08430.

[38] ultralytics, 2023. Available: https://github.com/ultralytics/ultralytics. Accessed on [2023-04-19].