Smart Agricultural Technology 12 (2025) 101590

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

YOLO-FCAP: An improved lightweight object detection model based on YOLOv8n for citrus yield prediction in complex environments

Tiwei Zeng [a,b,c,d], Jintao Tong [a,d], Xudong Sun [d,e], Jiacheng Liu [f], Xiangguo He [b,c], Zhenzhen Guan [b,c], Lingfeng Liu [a,d], Nan Jiang [a,d], Tao wan [a,d,*]

[a] School of Information and Software Engineering, East China Jiaotong University, Nanchang 330013, China
[b] Jiangxi Provincial Investment Group CO., Ltd, Nanchang 330029, China
[c] Jiangxi Tongji Construction Project Management CO., Ltd, Pingxiang 337000, China
[d] Key Laboratory of Advanced Network Computing (East China Jiaotong University), Jiangxi, 330013, China
[e] School of Mechatronics and Vehicle Engineering, East China Jiaotong University, Nanchang 330013, China
[f] Ganzhou Citrus Research Institute, Ganzhou 341004, China

ARTICLE INFO

Keywords:
Cirtus
Lightweight
Object detection
YOLOv8
Yield prediction

ABSTRACT

In citrus orchard, accurate object detection serves as a critical technology and an essential prerequisite for achieving precise yield prediction. However, the complex orchard environments where fruit and leaves obscure each other and the limited capabilities of edge devices pose challenges for the implementation of high-precision, lightweight detection algorithms. Using object detection algorithms to model data based on predicted and actual yields, it is feasible to predict the overall production of citrus trees in orchards effectively. To address the aforementioned challenges for efficient and accurate yield prediction, this study proposed YOLO-FCAP, a lightweight model based on an improved version of YOLOv8n. Firstly, the smaller network scaling factor called pico was introduced to reduce network redundancy and achieve a significant decrease in model size while ensuring performance. Secondly, all of the C2F modules were replaced with the FasterNetBlockWithCA module, which possesses stronger anti-background interference capability and a lighter-weight structure. Thirdly, the last three downsampling convolutional blocks of the backbone network were replaced with the ADown module to enable multi-scale feature fusion and improve the model's ability to retain detailed information. Finally, the small object detection layer was added to improve the model's accuracy for detecting smaller and occluded targets. The experimental results obtained on the self-constructed citrus dataset showed that YOLO-FCAP's parameters and FLOPs were reduced to 0.81 M and 5.5 G, respectively. Its precision, recall, and average precision reached 90.4 %, 84.3 %, and 92.5 % respectively, while the frame rate achieved 168.42 frames per second. In addition, this study developed a linear regression equation that uses YOLO predictions of fruit counts to estimate actual quantities with excellent performance (R^2=0.983, MAE=0.95, RMSE=1.20). The results demonstrated that YOLO-FCAP can efficiently predict yield in complex orchard environments, which provides technical support for orchard planning at various stages.

1. Introduction

Citrus fruits are one of the most widely grown fruits in the world, with their health-promoting effects primarily attributed to abundant phytochemicals—including flavonoids (e.g., hesperidin, naringin), carotenoids (e.g., β-carotene, lutein), and bioactive compounds (e.g., vitamin C, limonoids), which synergistically exert antioxidant, anti-inflammatory, immunomodulatory, and potential anticancer effects [1]. The Gannan navel orange, a species of citrus, is widely cultivated in the Gannan region of China. Renowned for its delicious flavour, nutritional value and attractive appearance, the Gannan navel orange has been designated the National Geographic Indication and has gained popularity among consumers [2]. Their antioxidant, anti-inflammatory, anti-cancer and cardiovascular protective properties position them as a significant player in the fruit market and greatly impact the local agricultural economy [3]. However, increasing market demand presents many challenges for traditional citrus orchard management. The Gannan navel oranges are currently mainly harvested by skilled

* Corresponding author.
E-mail address: i779669@163.com (T. wan).

https://doi.org/10.1016/j.atech.2025.101590
Received 4 August 2025; Received in revised form 28 October 2025; Accepted 30 October 2025
Available online 31 October 2025

labourers, which greatly reduces harvesting efficiency and increases costs [4]. Using intelligent equipment to efficiently manage orchards can alleviate the labour burden and ensure fruit quality. The application of artificial intelligence technology in the field of agriculture offers new guidelines and approaches for smart agriculture [5].

Autonomous fruit detection technology plays a particularly prominent role in many branches of smart agriculture, facilitating yield prediction, fruit picking and disease control. In yield prediction, accurate and fast detection allows managers to efficiently count the number of fruits and make rational planning for subsequent planting, harvesting or selling. Previous studies have used machine learning to predict yields. For instance, Khan et al., [6] compared nine machine learning models for predicting corn yield in the current season and concluded that the support vector regression model demonstrated relatively superior predictive performance, achieving an R^2 value of 0.875 for seasonal yield forecasts. Sapkota et al., [7] combined multispectral images acquired via unmanned aircraft systems with five machine learning methods for analysis. Ultimately, the extra trees regressor demonstrated the best predictive performance across most corn growth stages, achieving an R^2 value of 0.85 at the late vegetative stage. Zhu et al., [8] combined difference vegetation index and random forest models to predict winter wheat yield, with prediction results of $R^2 = 0.67$, Root Mean Square Error (RMSE) = 644.93 kg ha^{-1}, and Concordance Correlation Coefficient (CCC) = 0.80, respectively. Imtiaz et al., [9] employed yield monitoring data from harvester sensors and manual digging for training and validation, evaluating three machine learning algorithms (random forest regression, classification and regression trees, and gradient tree boosting) where gradient tree boosting demonstrated optimal performance, with R^2 values ranging from 0.71 to 0.78, RMSE values between 2.82 and 5.96 t/ha, and Mean Absolute Error (MAE) values from 2.33 to 4.2 t/ha. Although most machine learning models can achieve relatively accurate yield predictions, they underperform in complex environments and rely heavily on manually designed features, which raises the threshold for research.

Nowadays, experts are applying deep learning techniques in the field of fruit object detection with increasing sophistication. Yield prediction can be performed efficiently and accurately by a suitable object detection model. Deep learning-based object detection models usually employ convolutional neural networks for end-to-end feature extraction, which in turn identifies and localizes fruit objects in images [10]. Typically, those Convolutional Neural Network (CNN)-based object detection models are categorized into two-stage methods and one-stage methods based on two different approaches [11]. The two-stage object detection method generates Regions of Interest (ROI) containing positive samples from the input image in the first stage, and further classifying and positional refinement of each ROI in the second stage. Representative two-stage object detection methods include Region-based Convolutional Neural Networks (R-CNN) [12], Fast R-CNN [13], and Faster R-CNN [14]. Gao et al., [15] proposed a multi-class apple detection method based on faster R-CNN, achieving an average accuracy rate of 87.9 % for images of apples with four types of occlusions. Siricharoen et al., [16] proposed a detection and classification framework for recognising pineapple ripeness based on multisampling technique and mask R-CNN, with an average accuracy of 86.7 %. Despite their high accuracy, two-stage algorithms face a pair of significant challenges. Firstly, slower detection speeds do not meet real-time requirements. Secondly, the increased storage requirements make such algorithms difficult to deploy on edge devices [17].

The one-stage object detection method simplifies the detection task into one stage, skipping the step of generating candidate regions, and directly obtaining the category and localization of the object from the input image through a network [18]. This method optimizes computational efficiency and can quickly detect objects. Representative one-stage object detection methods include Single Shot MultiBox Detector (SSD) [19] and You Only Look Once (YOLO) [20]. Liu et al., [5] constructed a green crisp plums detection network with YOLOv8s-p2 as the baseline. This model achieved an average accuracy of 89.4 % with a model size of 67 MB. An et al., [21] designed a strawberry growth detection algorithm SDNet, which is based on the YOLOX model. Achieving an average accuracy of 94.26 %, with a model size of 54.6 MB. Xu et al., [18] proposed the HPL-YOLOv4 citrus detection model can quickly detect citrus in complex environments. The results demonstrated an average accuracy of 98.21 %, with a model size of 43.5 MB. Although these single-stage models performed well in terms of detection and required less storage than two-stage algorithms, there is still room for improvement in terms of model size and parameters.

In recent years, many researchers have further reduced the size of citrus detection models. Chen et al., [22] proposed a lightweight citrus detection model based on YOLOv7. It introduced small object detection layer, GhostConv and Convolutional Block Attention Module (CBAM) to achieve multi-scale feature fusion as well as a reduction in the parameters, lowering it to 24.26 M. Lu et al., [23] proposed a lightweight green citrus detection method that achieved accurate localisation of green citrus fruits in orchards with a model size of 12.4 MB. The model was improved by using the transformer mechanism to model global information in the backbone network and integrating the CBAM with the original module to ensure detection accuracy while maintaining a lightweight design. Gu et al., [24] proposed the YOLO-DCA model by replacing ordinary convolution in ELAN with Depthwise Separable Convolution (DWSConv), integrating Coordinate Attention (CA) into ordinary convolution in the neck network and utilizing Dynamic Detection Head in the head network. This citrus detection model achieved a size of only 4.5 MB and 2.1 million parameters in complex environments. Liao et al., [25] proposed YOLO-MECD, a citrus detection model based on YOLOv11, compressing the model size to 4.66 MB. The model introduced the Efficient Multi-scale Attention (EMA) to replace the original C2PSA, and replaced the C3K2 module with a partially convolution-based CSPPC module, the above improvements effectively reduced the parameters and computational complexity, and finally the Minimum Point Distance Intersection over Union (MPDIoU) loss function was used to improve the detection accuracy. Although the size of existing citrus detection models was greatly reduced, they performed poorly when faced with challenging samples, particularly heavily occluded targets. The frequency of misdetections and omissions would affect the benefits of applications. In terms of model size, existing networks still exhibited a large amount of parametric and computational redundancy, suggesting that improvements to existing lightweight models are incomplete. The lightweight convolutions of many networks adopted DWSConv [26] or GhostConv [27]. While such convolution can effectively reduce the parameters, they also incurred a certain computational expense, resulting in slower inference speeds. Inspired by the above, this study proposed an improved lightweight citrus detection model based on YOLOv8n named YOLO-FCAP. This model can accurately detect and easily deploy in complex citrus orchards.

The main contributions of this paper are as follows:

1) The study constructed a citrus dataset containing complex lighting, various degrees of obscuration and ripeness, and blurred scenes. Rich samples were provided for citrus detection.
2) The YOLOv8 network was improved in four steps to increase the efficiency and accuracy of the model for citrus detection. Firstly, the smaller network scaling factor called pico was introduced. Secondly, all of the C2F modules were replaced with the FasterNet-BlockWithCA module. Thirdly, the last three downsampling convolutional blocks of the backbone network were replaced with the ADown module. Finally, the small object detection layer was added.
3) The experimental comparison results on the citrus dataset, which contains many challenging samples, showed that YOLO-FCAP outperformed other object detection models in terms of overall performance. This study also fitted the data based on the linear relationship between the model's predicted yields and the corresponding actual yields from some of the samples, then established a linear regression

equation for predicting the yields of citrus fruit trees that were not actually counted or the approximate yields of the entire citrus orchard, which provided technical support for subsequent orchard planning and management by the managers as well as serving as a reference for the development of more lightweight models.

2. Materials and methods

2.1. Study area description

As demonstrated in Fig. 1, the experimental citrus orchard is located in Ganzhou Citrus Scientific Research Institute (114°50′48″ ~ 114°50′57″ E, 25°46′40″ ~ 25°46′44″ N), Jiangxi Province, China. The area is characterized by smooth undulations in the terrain that are suitable for cultivating a variety of crops, and benefits from a mild climate and fertile soil. Consequently, owing to its unique geographical location, favourable climatic conditions, rich soil and abundant water resources, this region provides an ideal environment for cultivating citrus fruits.

2.2. Data collection

The progression of the study is shown in Fig. 2. Under the guidance of the Institute's experts, two separate citrus samples from various periods were collected using the environmental condition sampling approach in this experimental area: One sample with mainly orange-yellow fruits at the ripening stage was taken on the 9th of December 2024, with images collected at 09:00 (weak light) and 12:00 (strong light); The other sample with all green fruits at the immature stage was taken on the 13th of October 2024, with images gathered at 08:00 (weak light) and 14:00 (strong light). This study used mobile devices to photograph fruit trees at different angles (horizontal, low-angle and overhead) and distances (50 cm ~ 250 cm) in different lighting conditions. The Redmi K60 and the iPhone 16 were selected as the cameras for capturing the citrus images, which were taken at a resolution of 3000 × 4000 with their rear cameras. Total of 4943 data samples containing citrus fruits exhibiting various light variations, levels of obscuration and ripeness, as well as blurred scenes, were obtained during the ripening stage. Since there were only a few green immature fruits in the dataset, another 1500 data samples from the immature period were selected to supplement the

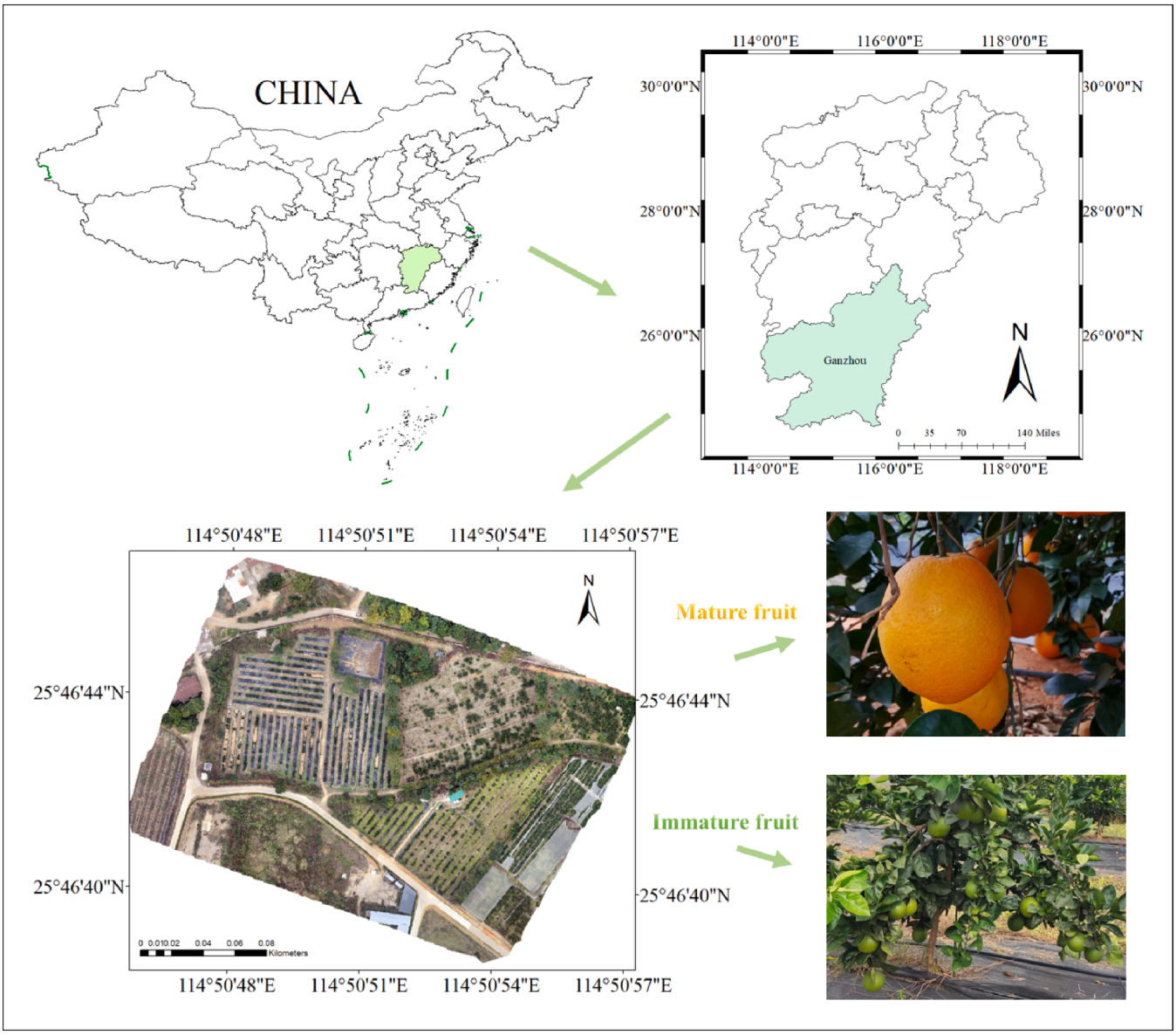

Fig. 1. Geographic location of the test site.

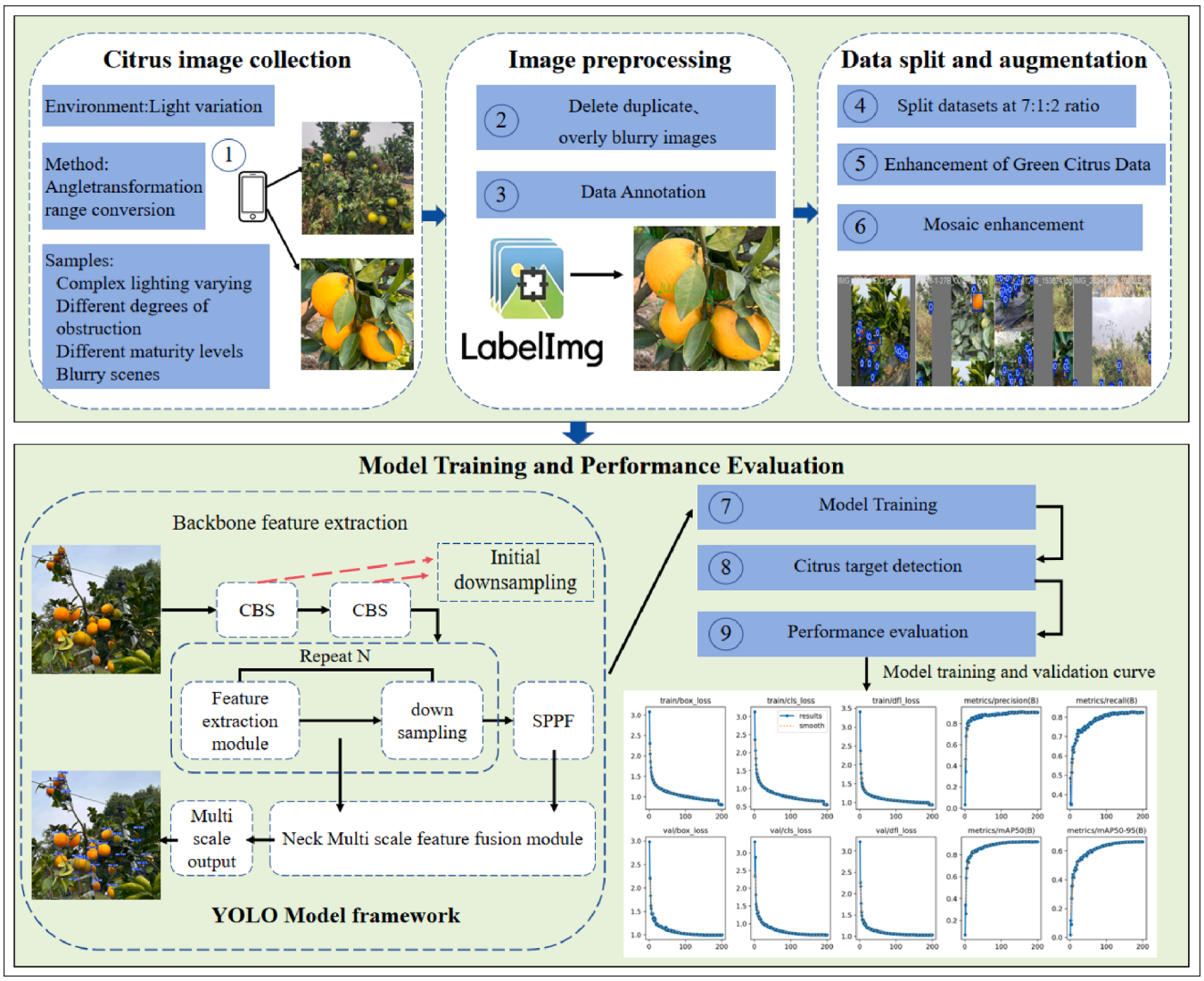

Fig. 2. Framework diagram of the experimental process.

dataset and increase the sample number. After careful screening (deleting duplicate images and overly blurred images), the two datasets included a total of 5816 citrus images.

The data was then annotated using the LabelImg software, with only one data label, NAO, being assigned to represent navel oranges. Each citrus target was enclosed within a precise bounding box. Annotations for obscured portions of the fruit were made based on their actual positions and fruits located further away in the image were not annotated. The annotations were then saved as the YOLO-format text file, with the file name matching that of the corresponding image. The preprocessed dataset comprises 4616 mature-stage images and 1200 immature-stage images. The 4616 mature-stage images and the 1200 immature-stage images were separately divided into training, validation, and testing datasets in a 7:1:2 ratio. Subsequently, a series of data augmentation techniques including random brightness variations, mirroring, and random angle rotation were applied exclusively to the immature images to enhance the diversity of the dataset. This increased the number of immature images to 2400, with the number of corresponding annotations growing accordingly. Merging the training, validation and testing sets from the two maturity stages produced a total of 7016 samples. To improve the model's performance of detecting objects in complex environments, this study applied mosaic data augmentation to the training samples. This process involved only randomly scaling, cropping, and arranging every four images before splicing them into a new image, without any additional enhancement. During model training and performance evaluation, data samples were input into various YOLO-based models. These models typically consist of three components: feature extraction, feature fusion and multi-scale output, which are further developed to enhance performance. After training, the obtained weights can be used for real-time citrus object detection. Training and validation curves, generated during the training process, can be used to effectively assess the model's convergence capability and detection performance.

2.3. Novel network YOLO-FCAP construction

YOLOv8 [28] is a new YOLO version open-sourced by Ultralytics on the 10th of January 2023. Compared to Yolov5 it replaces the C3 module with the C2F module which has a richer gradient flow. It also adopts the anchor-free idea and the decoupled head structure. These improvements make the network more efficient during training and inference. Additionally, the feature fusion as well as bounding box regression capabilities of the model are further enhanced by utilizing Path Aggregation Feature Pyramid Network (PAFPN) [29] and Complete-IoU (CIoU) loss [30]. However, the model's intricate design introduces additional parameters and computational expense, which limits its applicability on edge devices with limited computational resources.

The YOLOv8n network consists of three components: the backbone network, the feature fusion network and the detection head. To enhance object detection accuracy in intricate citrus orchard environments and facilitate model lightweighting, this study proposed an optimized design for the aforementioned core modules. The YOLO-FCAP model (illustrated in Fig. 3) was devised to enhance the detection performance and inference efficiency of the model. Initially, the pico scaling factor was employed to minimize network redundancy. Subsequently, the lightweight feature extraction module named FasterNetBlock was developed to replace the original C2F module. The module was enhanced with an attention mechanism to improve the model's capacity to differentiate between the background and the object while maintaining a lightweight design. By utilizing ADown as the novel lightweight downsampling module to replace the last three downsampling convolutional modules of the backbone network, this improvement enhanced multi-scale feature fusion in complex scenes while concomitantly improving detection efficiency. Additionally, to better meet the requirements of the complex environment for citrus multi-object recognition (small and occluded objects), this study added the additional small object detection layer.

2.3.1. YOLOv8pico

In the citrus detection task, the model's performance in real orchard environments and its feasibility for deployment on edge devices should be prioritized. By adjusting the network scaling factors (depth, width, max_channels), YOLOv8 can realize different versions. The depth adjusts the number of times the modules are stacked to regulate the depth of the network. The width scales the number of channels in each layer to change the width of the network. The max_channels constrains the upper limit of channels in the deep feature map. Although deeper, wider networks and more channels facilitate learning complex features, the model size and computational requirements also increase. Even though the YOLOv8n version is the smallest of the five, it still exhibits redundancy in the citrus detection task. Therefore, this study introduced a smaller network scaling factor pico to build a lighter model. YOLOv8n has a depth of 0.33, a width of 1.25, and max_channels of 1024. To halve the number of channels at each layer of the network, the width was set to 0.125 and all other values remained unchanged in this study. Each layer

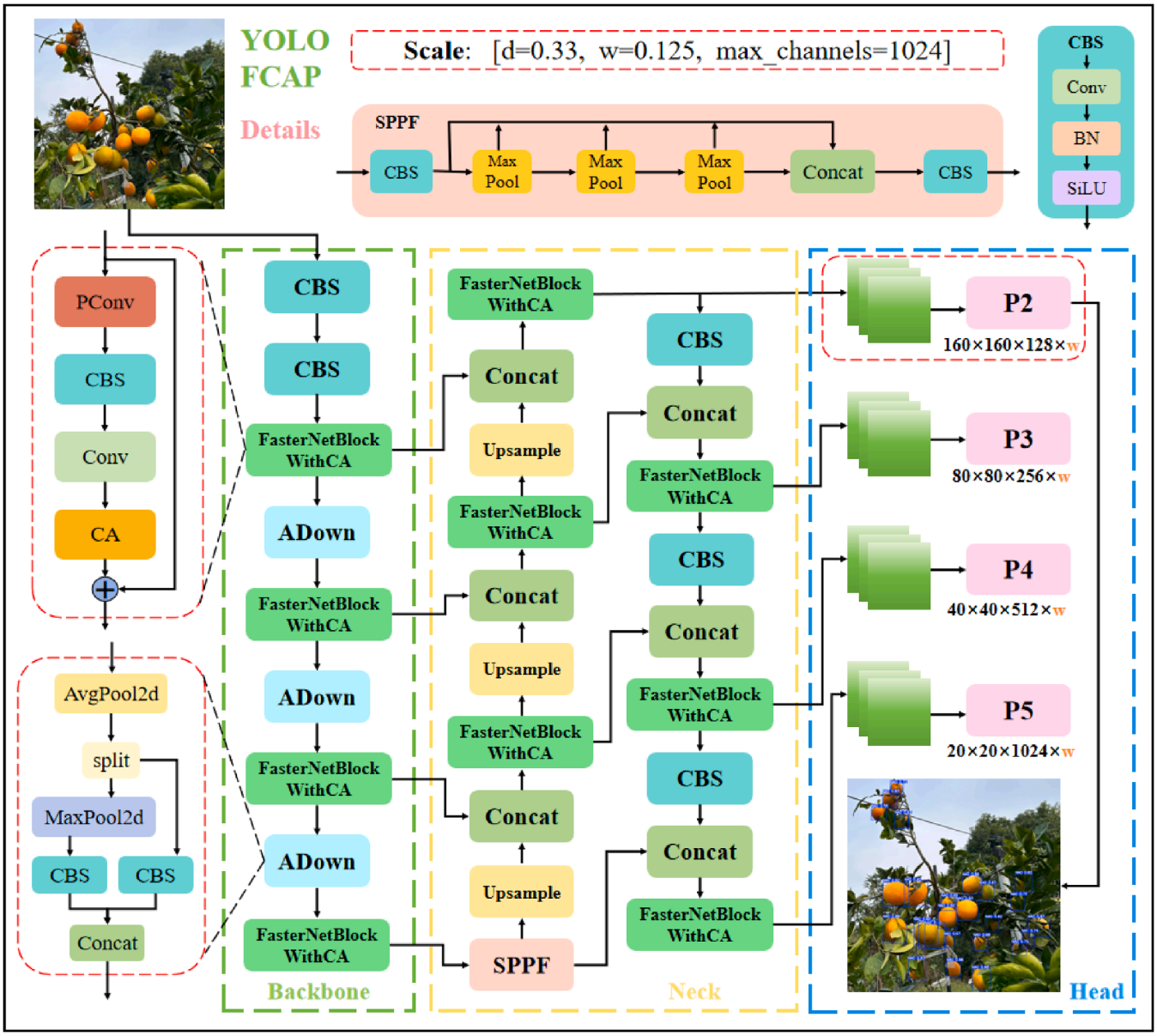

Fig. 3. Schematic diagram of the YOLO-FCAP network. The contents of the red dashed box in the figure are the improvement of the paper.

of the network decreased in the number of channels by half. Thus, there were 512 channels in the deepest layer of the improved network.

2.3.2. FBCA

As shown in Fig. 4, by utilizing the Grad-CAM method to visualize the YOLOv8p citrus detection model in the middle layer, it was observed that the majority of channels could capture citrus features with significant similarity. The C2F feature extraction module was introduced in YOLOv8. While its rich gradient flow branching can extract multi-scale features more efficiently, this structure also incurs greater computational expense. This excess feature extraction capacity makes the module complicated and redundant for citrus detection tasks. Additionally, background interference from tree trunks, leaves, and other extraneous objects poses a notable challenge when detecting objects in an orchard environment. Using a suitable attention mechanism can suppress such interference, enabling the network to focus more effectively on citrus objects. Based on the aforementioned points, this study improved the feature extraction module in this section. The standard convolution method uses all channels for feature extraction, and its FLOPs are calculated as follows:

$$\mathrm{FLOPs}_{\mathrm{Conv}} = h \times w \times k^2 \times c^2 \tag{1}$$

Where h and w represent the height and width of the input feature map, respectively, k represents the size of the convolution kernel, and c represents the number of channels, which are generally categorized as C_{in} and C_{out}. In this case, the convolutional computation generates the same number of output feature maps as there are input channels.

Liu et al., [31] proposed Partial Convolution (PConv), which convolves only a portion of the input channel and maintains the remaining channel features. This approach efficiently captures key features and minimizes redundant computations by processing feature information from only part of the channels. Its FLOPs are calculated as follows:

$$\mathrm{FLOPs}_{\mathrm{PConv}} = h \times w \times k^2 \times c_{\mathrm{p}}^2 \tag{2}$$

When the partial ratio C_{p}/C is 1/4, the FLOPs of a PConv are only 1/16 of the standard convolution. The calculation is as follows:

$$\frac{\mathrm{FLOPs}_{\mathrm{PConv}}}{\mathrm{FLOPs}_{\mathrm{Conv}}} = \frac{\mathrm{h} \times \mathrm{w} \times \mathrm{k}^2 \times \left(\frac{\mathrm{c}}{4}\right)^2}{\mathrm{h} \times \mathrm{w} \times \mathrm{k}^2 \times \mathrm{c}^2} = \frac{1}{16} \tag{3}$$

Based on PConv and Pointwise Convolution (PWConv), Liu et al., [31] proposed FasterNet. The main module of FasterNet is FasterNet-Block which is efficient and lightweight. Its specific structure is shown in Fig. 5.(a). The module first extracts channel features through partial convolution. Then, two point-by-point convolutions establish full channel interaction, doubling the number of channels in the first layer but reducing it in the second. Residual concatenation is used for the input features to multiplex. To balance performance and efficiency, apply Batch Normalization (BN) [32] and Sigmoid Linear Unit (SiLU) [33] only in the first convolutional block following PWConv. Although the module can focus on key channel features, it still lacks the capability to distinguish between the background and objects. This study considered introducing an attentional mechanism to improve feature selectivity.

Hou et al., [34] proposed a lightweight attention mechanism CA which accurately captures spatial location information of objects. As shown in Fig. 5.(b), to preserve positional information and capture long-range dependencies between different positions, the global pooling utilized by previous attention mechanisms is decomposed into horizontal and vertical feature encoding operations. Then, the pooling kernels $H \times 1$ and $1 \times W$ are used to obtain aggregated feature maps with two different directions from feature maps with an input size of $C \times H \times W$. The calculations for the pooling process are as follows:

$$\mathrm{Z}_{\mathrm{c}}^{\mathrm{h}}(H) = \frac{1}{\mathrm{W}} \sum_{0 \le \mathrm{i} < \mathrm{W}} \mathrm{x}_{\mathrm{c}}(\mathrm{h}, \mathrm{i}) \tag{4}$$

$$\mathrm{Z}_{\mathrm{c}}^{\mathrm{w}}(\mathrm{W}) = \frac{1}{\mathrm{H}} \sum_{0 \le \mathrm{i} < \mathrm{H}} \mathrm{x}_{\mathrm{c}}(\mathrm{j}, \mathrm{w}) \tag{5}$$

Following bi-directional feature encoding, the CA module can capture single-direction, long-range dependencies and retain position information in the other direction. Next, the two feature maps are spliced together using 1×1 convolutional features and integrated. Immediately after, the batch-normalized output is split into two separate feature

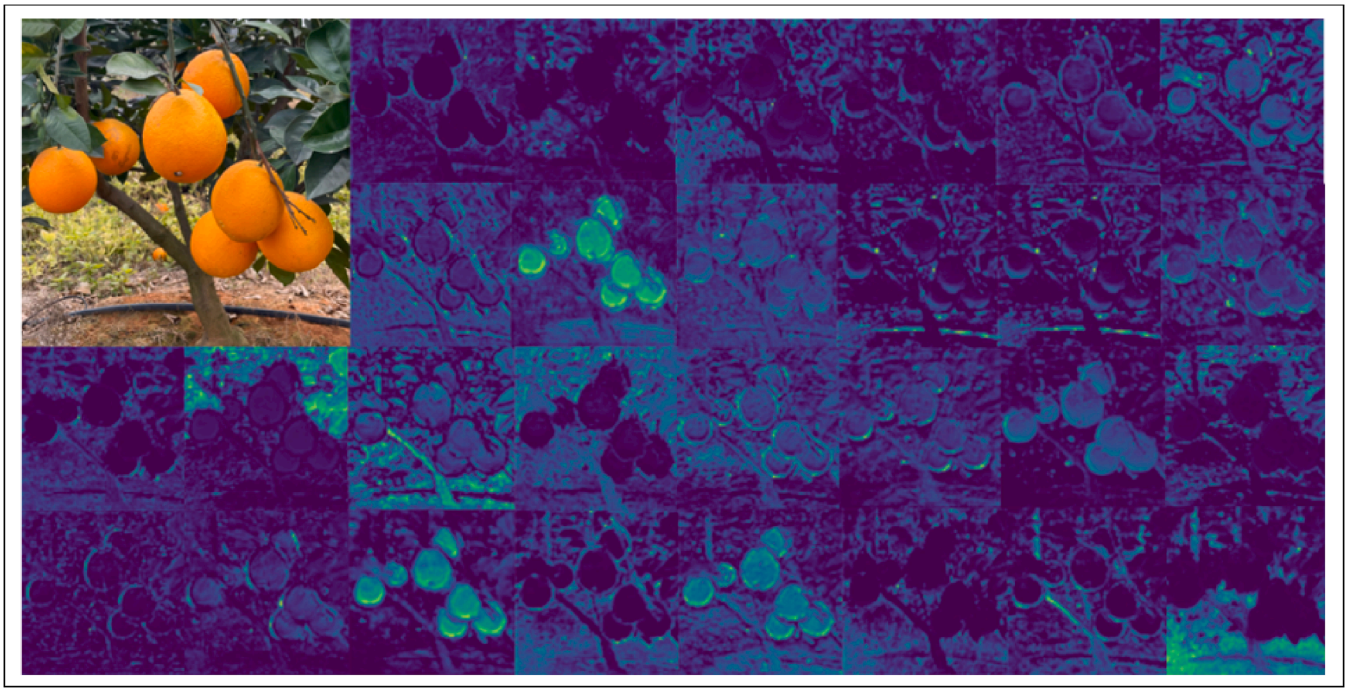

Fig. 4. Feature visualization of the YOLOv8p intermediate stage layer.

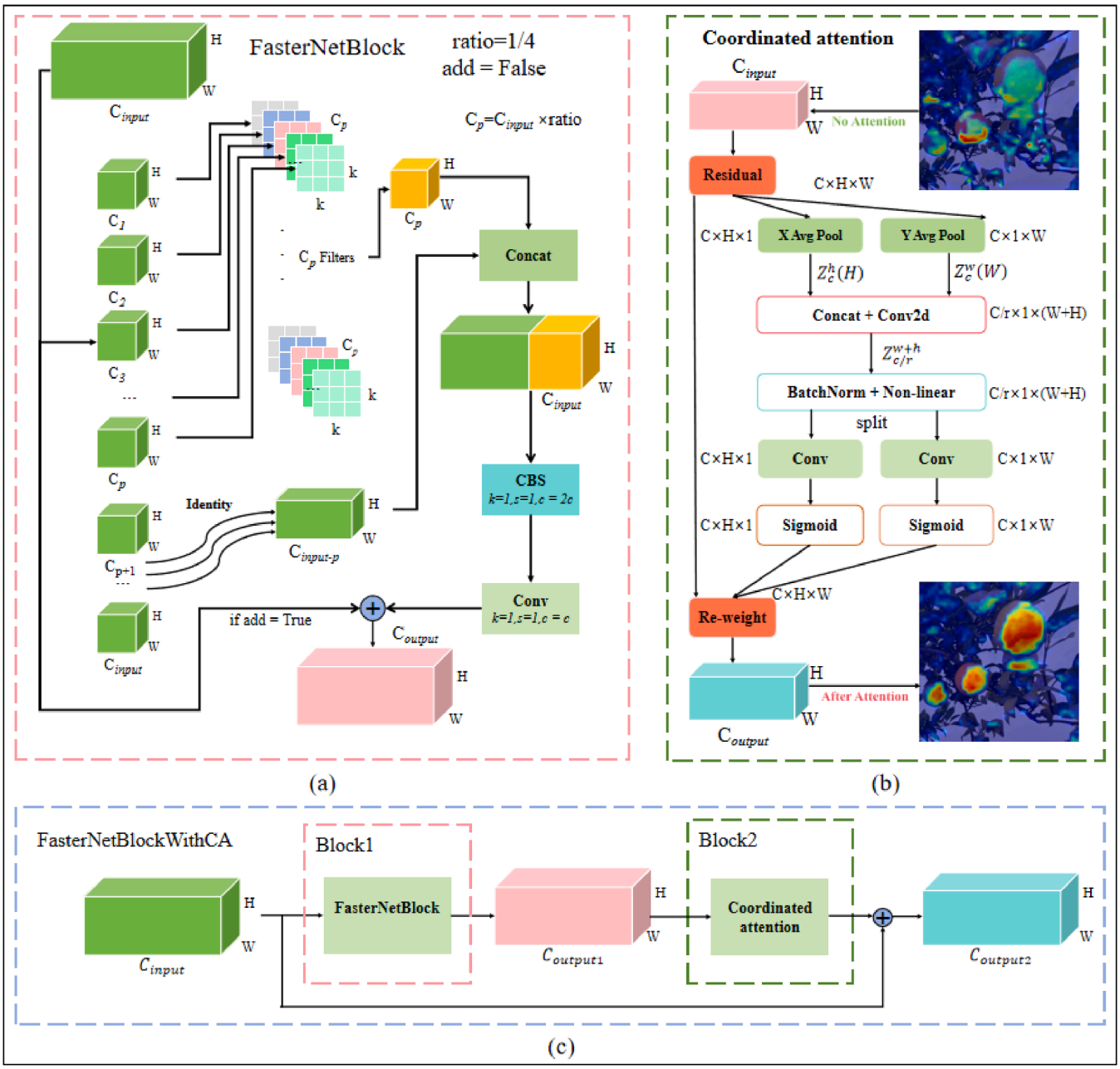

Fig. 5. Schematic of the FBCA (FasterNetBlockWithCA).

maps in both directions. The calculation process procedure is as follows:

$$f = \delta\left(\mathrm{BN}\left(Z_{c/r}^{w+h}\right)\right) \tag{6}$$

The final output, weighted by attention, is obtained using the following computational procedure:

$$\mathrm{Out_c}(i,j) = \mathrm{In_c}(i,j) \times g_c^h\left(\delta\left(F_h\left(f^h(i)\right)\right)\right) \times g_c^w(\delta(F_w(f^w(j)))) \tag{7}$$

Where, r is the scaling down ratio of the number of channels in the middle layer, δ is the nonlinear activation function, BN is the batch normalization operation, $Z_{c/r}^{w+h}$ is the joint tensor obtained after the splicing and convolution operation, f^h and f^w are the two independent tensors after the splitting, F_h and F_w are the convolution operations, while g_c^h and g_c^w are the two weights in different directions.

Fig. 5.(c) shows the structure of the FasterNetBlockWithCA (FBCA). Following FasterNetBlock, the coordinated attention mechanism is embedded, and residual connection is placed after the attention mechanism. This approach compensates for object information missed during PConv feature extraction. Consequently, the module enhances the ability to distinguish citrus objects from the background, improving detection performance. This study eventually replaced all C2F modules in the network by using FBCA as a feature extraction module. With this replacement, the ability to focus on and identify critical features will be enhanced while computational costs are minimized.

2.3.3. ADown

The citrus dataset in this study predominantly contains multi-scale targets, including small and medium-sized objects. Effectively detecting such targets requires considering the ability of feature fusion and detail retention. However, traditional downsampling methods such as the pooling layer or the stepwise convolution are prone to losing details when compressing features. Their monolithic structure also makes it

challenging to fuse multi-scale features effectively.

Balancing multi-scale feature fusion with detail preservation, the ADown module is structured with two branches. Features are first pre-processed using average pooling to expand the receptive field and preserve global information. Following this, the feature map splits into two channels. Branch 1 performs a traditional 3 × 3 convolution with a step size of 2 for standard downsampling. Branch 2 extracts significant features utilizing max pooling and realizes the global interaction of channel dimensions using a 1 × 1 convolution. Consequently, the capability of multi-scale feature fusion and detail preservation for small and medium-sized objects is significantly improved while reducing the parameters and computational expense. This design is well-suited to the citrus detection task of this study, and its specific structure is shown in Fig. 6.

2.3.4. Small object detection layer

The head network of YOLOv8 employs three detection heads of scale 80 × 80, 40 × 40, and 20 × 20, which correspond to small, medium, and large object detection, respectively. As demonstrated in Fig. 7, the citrus dataset of this study contains a large number of small objects (both their normalised width and height are less than 0.1 times the dimensions of the image., and most of the objects are concentrated at 0.05), aligning with standard definitions of small objects. When the input image size is 640 × 640 pixels, objects around 32 × 32 pixels gradually lose spatial information due to downsampling. After downsampling to an 80 × 80 pixels feature map, these objects are only 4 pixels. Moreover, since the exposed area is limited, many occluded citrus fruits are also classified as special small objects. Downsampling continuously results in a notable loss of detail for these two types of objects, which hinders the model's capability to extract effective features. There is consequently an increase in leakage and a decrease in accuracy.

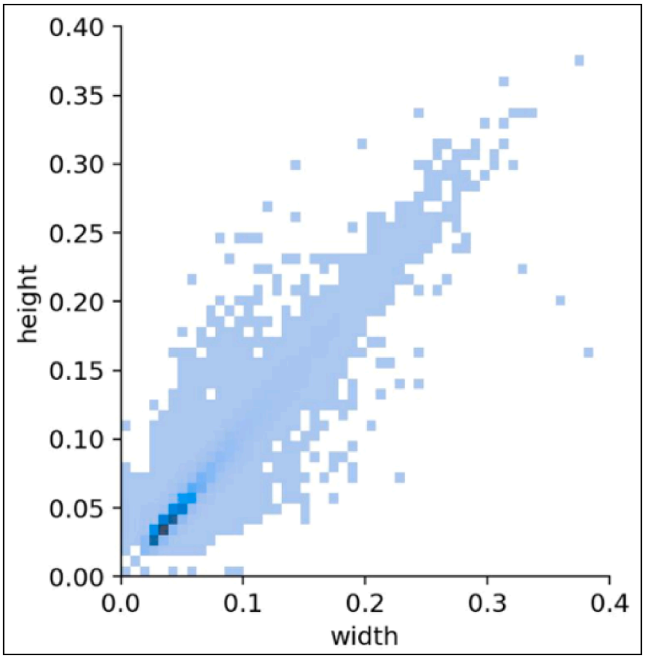

Fig. 7. Plot of the size distribution of the normalized experimental samples.

To address the aforementioned challenge, this study added a new small object detection head with a resolution of 160 × 160 pixels to the

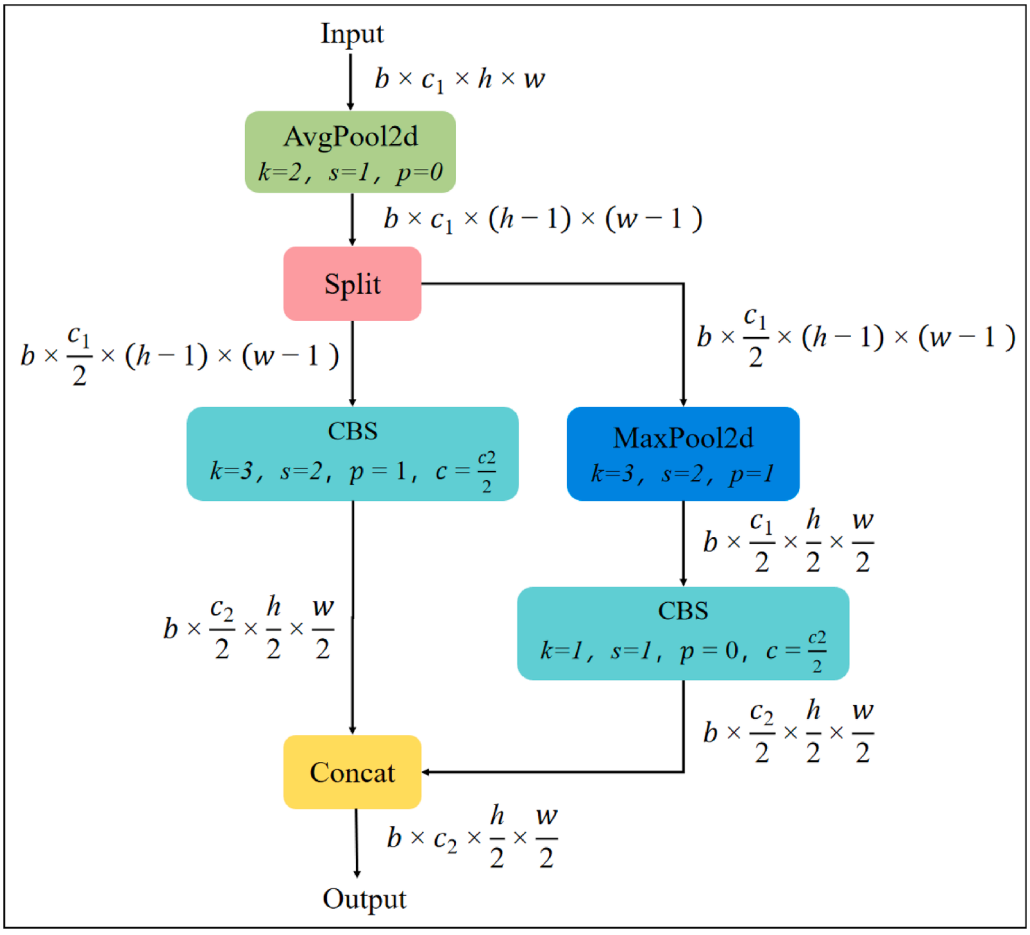

Fig. 6. Schematic of the ADown (Adaptive Downsampling).

detection head network. Correspondingly, an additional detection layer was established in the neck network. As shown in Fig. 8, the neck network upsamples 80 × 80 pixels feature maps and combines them with 160 × 160 pixels feature maps from the backbone network. The result is that more spatial information of smaller objects is retained. Then, the fused feature maps are input into the added detection head. All four detection layers significantly improved multi-scale performance, especially for smaller objects.

2.4. Accuracy assessment

2.4.1. Experimental environments

The model was trained on a desktop computer with the following specifications: Windows 10 Professional, Intel (R) Core (TM) i7–14700KF 3.40 GHz CPU, 32 GB RAM, and NVIDIA GeForce RTX 4070 GPU. The experimental code was based on CUDA 12.8, Python 3.9.21, and PyTorch 2.6.0. Visual Studio Code 1.99.3 was used as the integrated development environment (IDE). This study trained the models by resizing the input images to 640 × 640, setting the batch size at 32, and using the SGD optimizer. The momentum and weight decay were set at 0.937 and 0.0005, respectively, while the initial and final learning rates were set at 0.01. Each model finished training after 200 epochs. This study did not use pre-trained weights for any of the models, and each model was trained with the same dataset and parameters.

2.4.2. Model and yield prediction metrics

To accurately assess the performance of the improved model, this study used Precision (P), Recall (R), Average Precision (AP) and mean Average Precision (mAP) to evaluate the model's performance in detecting citrus objects. Then, model size, Floating Point Operations (FLOPs), parameters, and Frames Per Second (FPS) were used to evaluate the model's complexity and speed of detection. The IoU threshold for AP evaluation was set to 0.5, and P, R, AP were calculated as follows:

$$\mathrm{P} = \frac{TP}{TP + FP} \times 100\% \tag{8}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \times 100\% \tag{9}$$

$$\mathrm{AP} = \int_0^1 (P * R)\mathrm{d}R \tag{10}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} \mathrm{AP}_i \times 100\% \tag{11}$$

Where *TP* denotes the number of correctly detected citrus instances in the test image. *FP* denotes the number of incorrectly detected instances, and *FN* denotes the number of missed instances. AP is calculated from the area under the accuracy-recall curve and mAP is the mean of AP.

The model size represents the size of the weight file generated following the completion of model training and is affected by the network structure and parameters. By adjusting the model structure, the size can be reduced without decreasing accuracy, creating a lighter model. FLOPs, parameters, and FPS are calculated as follows:

$$\mathrm{FLOPs} = \sum (H \times W \times K \times K \times C_{\mathrm{in}} \times C_{\mathrm{out}}) \tag{12}$$

$$\mathrm{Parameters} = \sum (K \times K \times C_{\mathrm{in}} \times C_{\mathrm{out}}) \tag{13}$$

$$FPS = \frac{\mathrm{N}}{\mathrm{t_N}} \tag{14}$$

Where $H \times W$ denotes the width and height of the output feature map. K denotes the size of the convolutional kernel. C_{in} denotes the number of input channels. C_{out} denotes the number of output channels. N is one second and t_{N} denotes the total time the model spends detecting the image, including preprocessing, inference, and post-processing times.

To evaluate the accuracy of YOLO-FCAP in estimating citrus yields, this study performed a data fit with data derived from the predicted versus actual count of each citrus tree used for comparison. The formula for the simple linear regression used for data fitting is:

$$\hat{\mathrm{y}} = \beta_0 + \beta_1 \mathrm{x} \tag{15}$$

The parameters β_0 and β_1 of the linear regression are optimized by least squares with the objective of minimizing the residual sum of squares (RSS):

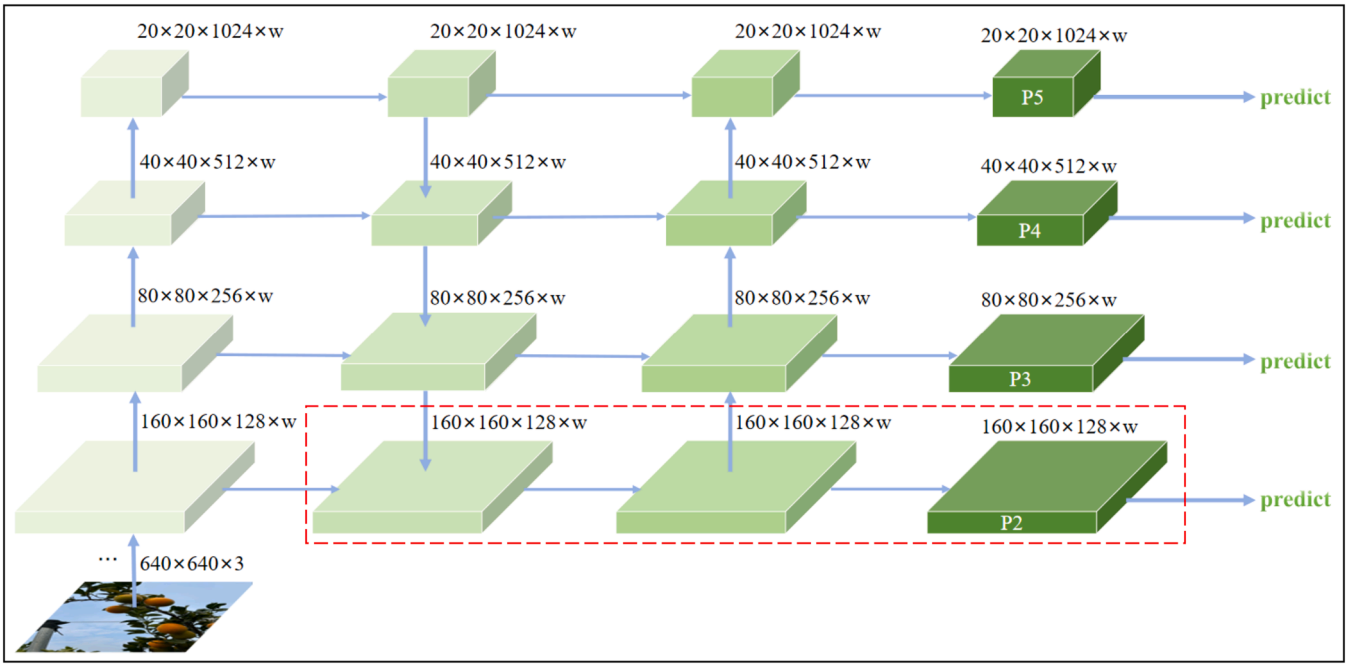

Fig. 8. Schematic after adding a small object detection layer.

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 \tag{16}$$

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{17}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \tag{18}$$

R^2 is used to measure the fit of the model to the data with the formula:

$$\mathrm{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{19}$$

$$R^2 = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} \tag{20}$$

RMSE reflects the overall magnitude of prediction errors, while MAE measures the average absolute deviation between predicted and actual values. The formulas are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{21}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{22}$$

where $\hat{y}_i$ denotes the predicted value of the fitted model for the i th tree, x_i denotes the predicted number of citrus fruits output by the YOLO-FCAP model on the i th tree, y_i denotes the actual number of citrus fruits on tree i, n denotes the total number of citrus trees used to fit the regression model, β_0 denotes the model offset, β_1 denotes the proportionality between the number of model detections and the actual quantity, and $\bar{x}$ and $\bar{y}$ denote the mean values of x and y, respectively.

3. Results

3.1. Model ablation experiments

In this study, an efficient citrus detection algorithm was developed by modifying the original YOLOv8n network. There were four improvements made to YOLOv8: the network scaling factor was adjusted, the C2F module was replaced with the FBCA module, the ADown module was employed in the last three downsampling parts of the backbone network, and the small object detection layer (P2) was introduced. Ablation experiments were conducted with the same training environment and hyperparameters. Fig. 9 demonstrates the impact of each module on the model's detection accuracy after incorporation.

The single and combined improvements were performed sequentially using YOLOv8n as the baseline. Notably, since this study scaled the final network to pico, the remaining three improvements were built on YOLOv8p instead of YOLOv8n.

As shown in Table 1, scaling the network factor to pico significantly reduces the model's parameters and FLOPs by 30.23 % and 34.15 %, respectively. Then, FBCA was employed as the new feature extraction module in YOLOv8p. The parameters decreased to 29.57 % of the baseline, and the P, R, and AP of YOLOv8p increased by 1.5 %, 1.6 %, and 2 %, respectively. Introducing ADown reduced the parameters and FLOPs to 27.91 % and 32.93 %, respectively, while slightly decreasing the AP compared to YOLOv8p. Finally, the small-target detection layer was added to significantly improve the model's accuracy in detecting small targets, despite a slight increase in parameters and FLOPs. This improvement clearly benefits the accuracy-efficiency trade-off: P, R, and AP increased by 1.4 %, 1.8 %, and 2.2 %, respectively, for YOLOv8p when P2 was incorporated. The introduction of the ADown and FBCA modules on YOLOv8p-p2 improved the AP to 90.6 % and 92.1 %, respectively, while reducing the parameters to 0.85 M and 0.88 M and the FLOPs to 5.6 G. When YOLOv8p integrated the new feature extraction and downsampling modules, the AP reached 90.2 %, and the parameters and FLOPs decreased to 0.82 M and 2.7 G, respectively. After integrating the four improved parts, the YOLO-FCAP model's parameters and FLOPs were reduced to 26.91 % and 67.07 % of YOLOv8n's, respectively. The AP improved by 0.6 % compared to the baseline. The experimental data revealed that the YOLO-FCAP effectively enhanced detection accuracy for citrus targets in complex orchard scenarios while significantly improving computational efficiency.

3.2. Comparative experiments on attentional mechanisms

To assess the effectiveness of the CA, four attention mechanisms were

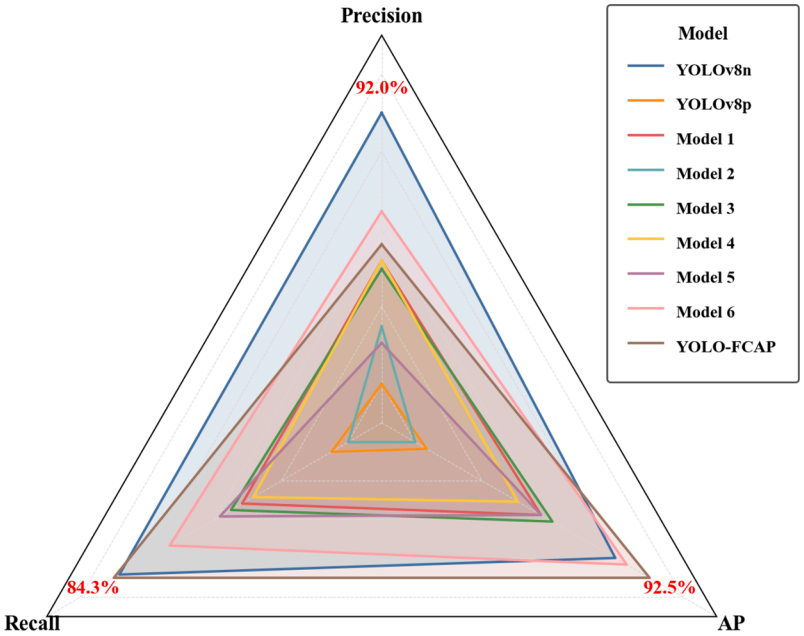

Fig. 9. Radar plots comparing the accuracy of each model.

Table 1
Results of the ablation experiments.

Model	ScalingFactors	FBCA	ADown	P2	P(%)	R(%)	AP(%)	Parameters(M)	FLOPs(G)
YOLOv8n					**92.0**	84.2	91.9	3.01	8.2
YOLOv8p	√				88.7	80.4	88.6	0.91	2.8
Model 1	√	√			90.2	82.0	90.6	0.89	2.8
Model 2	√		√		89.4	80.1	88.4	0.84	**2.7**
Model 3	√			√	90.1	82.2	90.8	0.92	5.7
Model 4	√	√	√		90.2	81.8	90.2	0.82	**2.7**
Model 5	√		√	√	89.2	82.4	90.6	0.85	5.6
Model 6	√	√		√	90.8	83.3	92.1	0.88	5.6
YOLOFCAP	√	√	√	√	90.4	**84.3**	**92.5**	**0.81**	5.5

embedded into the FasterNetBlock module of YOLOv8p for comparative experiments. Experiment results are presented in Table 2. Based on YOLOv8p-FasterNetBlock, introduced CA increased R and AP to 82.0 % and 90.6 %, respectively. Compared to the baseline, AP improved by 1.2 % and R increased by 1.6 %, while P decreased. The rest of the attention mechanisms increased AP, though the boost was relatively slight. Notably, the introduction of the attention mechanism enhanced concentration on the object region, which increased the R of all models and indirectly resulted in a slight decrease in P. However, P still meets actual demands.

As illustrated in Fig. 10, the visualization demonstrates significant variances in areas of focus among various attention mechanisms in the citrus object detection task. Although Shuffle Attention (SA) [35] and EMA [36] can capture more object regions, they also introduce substantial background noise. The EMA uses a 3×3 convolutional kernel in a parallel subnetwork to achieve multiscale feature fusion. The SA employs a rearrangement operation to increase feature diversity and expressive capability. However, their grouping strategies result in an uneven distribution of grouped features and insufficient subfeature information. The Efficient Channel Attention (ECA) [37] lacks the capacity to address global contextual dependencies in the spatial dimension. It also neglects cross-dimensional interactions between channels and space. Consequently, ECA performed relatively poorly in terms of detection. By contrast, the CA mechanism combines cross-channel information interaction with accurate position preservation and long-range spatial dependency modeling. Significantly improving focus on the citrus object region and the ability to differentiate features while suppressing background interference. Thus, the FasterNetBlock module with CA achieved superior detection accuracy.

3.3. Comparative experiments of lightweight feature extraction modules

To evaluate the effectiveness of the proposed feature extraction modules, experiments in this section were conducted utilizing four lightweight modules for comparative analysis. Additionally, the impact of deploying the FasterNetBlock module at various network locations on model performance was examined. Table 3 presents the results of these experiments. Fig. 11 illustrates the AP differences in various module and location combinations.

Based on YOLOv8p, this study sequentially performed comparison experiments by replacing the backbone C2F module with GhostNetv2, ShuffleNetv2, MobileNetV3, and FasterNetBlock. The results show that the first three replacements all decreased AP, though they reduced parameters and FLOPs more. However, the FasterNetBlock module appeared to achieve a more comprehensive effect. Using this module increased AP to 89.2 % and reduced parameters and FLOPs to 0.86 M and 2.7 G, respectively. FasterNetBlock's core innovation is its partial convolution mechanism, which focuses on feature extraction for specific channels and applies convolutional computation only to valid pixels, ignoring or masking invalid or missing pixels. This results in redundant feature suppression and key feature enhancement, increasing accuracy while reducing FLOPs.

Table 2
Comparative experimental results of different attentional mechanisms.

Baseline	Attention Module	P(%)	R(%)	AP(%)
YOLOv8p-FasterNetBlock	–	**90.4**	80.4	89.4
YOLOv8p-FasterNetBlock	SA	89.6	80.6	89.6
YOLOv8p-FasterNetBlock	EMA	89.8	81.2	89.8
YOLOv8p-FasterNetBlock	ECA	90.1	81.4	90.0
YOLOv8p-FasterNetBlock	CA	90.2	**82.0**	**90.6**

The positional ablation experiment shows that replacing only the neck part of the C2F module can increase AP to 89.0 %, while keeping the parameters unchanged and slightly raised FLOPs to 2.9 G. Two primary factors contributed to this result: the neck part's special structure for small-object detection and the inverted residual structure in the FasterNetBlock module. Completely replacing the C2F backbone resulted in model P, R, and AP values of 90.4 %, 80.4 %, and 89.4 %, respectively. Compared to the baseline, R remained constant, while the other two values improved by 1.7 % and 0.8 %, respectively. The parameters decreased to 0.85 M. The FLOPs also remain unchanged. Overall, the results demonstrate the effectiveness of the selected module for citrus detection.

3.4. Comparative experiments of different object detection models

In order to verify the advantages of YOLO-FCAP in citrus detection tasks, five lightweight YOLO models were selected to conduct comparative experiments. The results (Table 4) show comprehensive superiority of YOLO-FCAP in four indexes: R, AP, parameters, and FLOPs. Although achieved the lowest precision, which still exceeds 90 %. Fig. 12 clearly evidences that this improved model significantly optimized computational efficiency and model size while maintaining high detection performance.

In addition, five different images were randomly selected from the test set, covering a variety of real complex scenes in the orchard environment, and then visually compared with each scene containing difficult samples. For example, in (a), the cloudy and distant scene causes spatial information about the citrus objects to be easily lost, and the lack of light results in the loss of citrus color and texture features. In (b), densely clustered objects with varying degrees of occlusion obscure the citrus, with the highest degree of occlusion exceeding 90 %. High-occlusion objects exhibit background features that far exceed object features. (c) and (d) contain immature objects with varying illumination. The illumination affects the surface features of the objects. Since the immature green citrus are similar in color to the background, the network incorrectly identified leaves as citrus, which resulted in false detections. (e) mainly reflects the blurring phenomenon caused by the movement of the edge device or lens dirt. The blurring also changes the object features. These five scenarios are a suitable test of the model's ability to detect difficult citrus samples in complex environments. As shown in Fig. 13, YOLO-FCAP produced one false detection in the first scenario, one missed detection in each of the second through fourth scenarios, and correctly detected all objects in the last scenario.

The FBCA accurately localized the object and extracted effective features, the CA mechanism effectively distinguished the citrus object

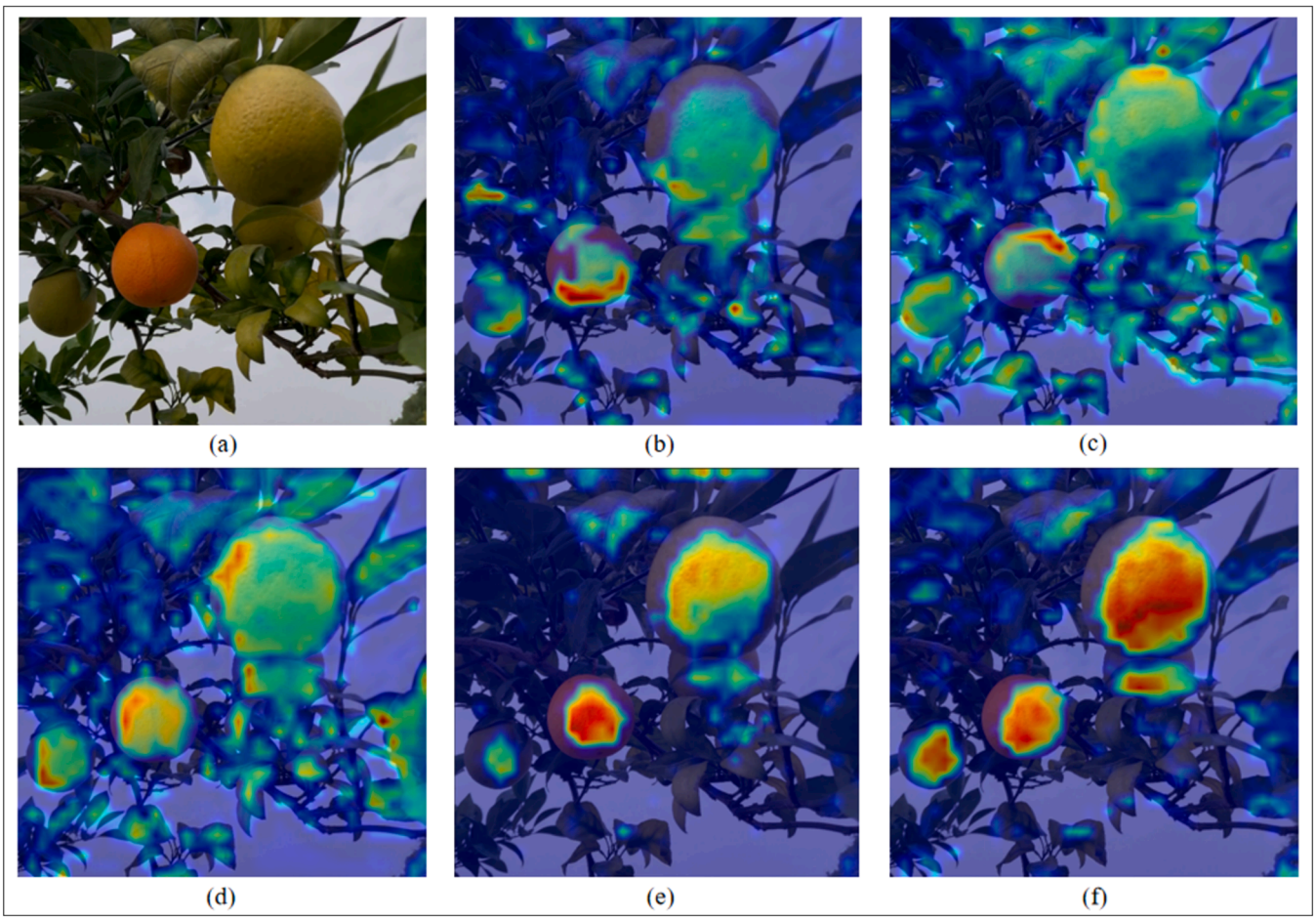

Fig. 10. Heat maps of different attention mechanisms.
(a)original picture. (b)without attention.
(c)Shuffle Attention (SA). (d)Efficient Multi-scale Attention (EMA).
(e)Efficient Channel Attention (ECA). (f)Coordinate Attention (CA).

Table 3
Comparative results of modules deployed at various locations.

Baseline	Module	P(%)	R(%)	AP(%)	Parameters (M)	FLOPs (G)
YOLOv8p	C2F(Baseline)	88.7	**80.4**	88.6	0.91	2.8
YOLOv8p	GhostNetv2-Backbone	89.3	77.1	87.1	0.78	2.4
YOLOv8p	ShuffleNetv2-Backbone	89.2	77.4	87.2	**0.75**	**2.3**
YOLOv8p	MobileNetv3-Backbone	89.1	79.4	88.1	0.85	2.6
YOLOv8p	FasterNetBlock-Backbone	89.9	80.1	89.2	0.86	2.7
YOLOv8p	FasterNetBlock-Neck	90.2	79.9	89.0	0.91	2.9
YOLOv8p	FasterNetBlock-All	**90.4**	**80.4**	**89.4**	0.85	2.8

from the complex background. The small-object detection layer and the ADown downsampling module synergistically fused the multi-scale features and preserved the object detail information. Despite a few misdetections/misses (within acceptable range), the improved lightweight model achieved a well-balanced between detection accuracy and efficiency, which fully validated its adaptability advantages for citrus detection tasks.

3.5. Comparative experiments of different citrus detection models

To validate the performance and detection speed advantages of YOLO-FCAP, the model was compared with some existing citrus detection models on two distinct citrus datasets. The CitDet Dataset [42] was collected between October 2021 and October 2022, comprising a total of 579 images covering multiple different varieties of citrus objects. In this experiment, the 5-fold cross-validation method was employed to obtain more stable and reliable evaluation results. Since most citrus detection models do not have open-source code available, this study reproduced several such models based on their corresponding papers, then trained and validated them under identical settings.

As shown in Fig. 14, YOLO-FCAP maintains moderate AP and mAP values for different citrus varieties included in various datasets across each fold, with performance comparable to the top-performing YOLOv5-CS. This indicates that the improved model possesses an excellent ability to generalize. Table 5 presents the comparative results of five citrus detection models using five-fold cross-validation on different datasets, with the metrics P, R, AP, mAP0.5 and FPS representing average performance indicators. Results indicate that on the self-built dataset, YOLO-FCAP achieved an AP value of 92.7 %, ranking second among all models and trailing the top-performing YOLOv5-CS by only 0.6 %. On the CitDet dataset, YOLO-FCAP achieved mAP value of 84.3 % to rank third, lagging behind YOLOv5-CS by 1.8 %. However, with a size of 1.95 MB and 121 FPS, the improved model demonstrates the lightest and fastest performance compared to others. Although its detection

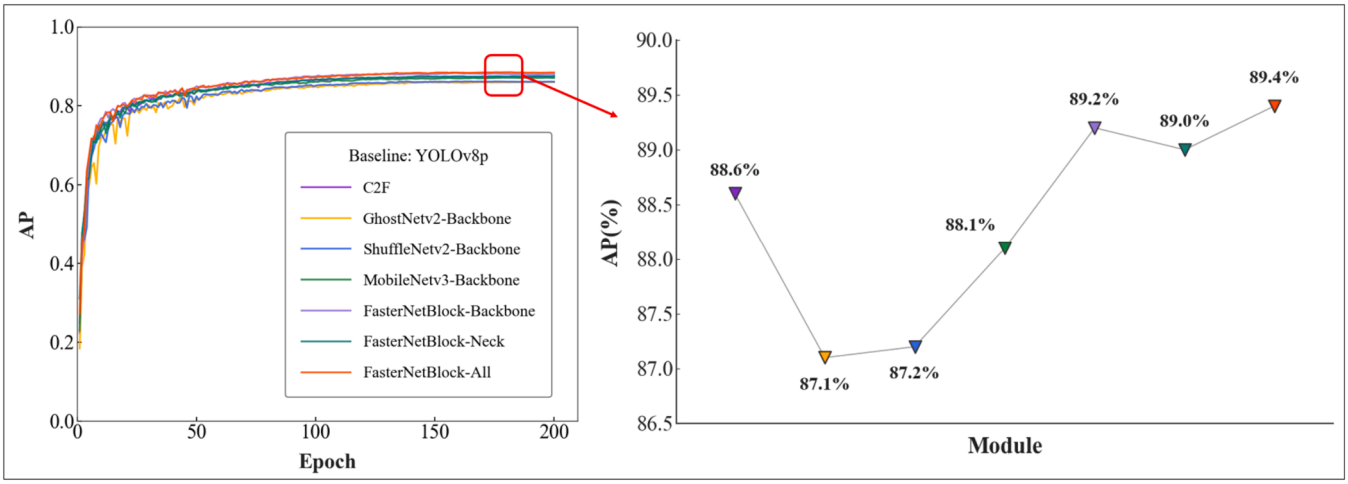

Fig. 11. Comparison of AP for each feature extraction module in different location deployments.

Table 4
Comparative experimental results of different object detection models.

Model	P(%)	R(%)	AP(%)	Parameters(M)	FLOPs(G)
YOLOv5n	91.5	83.5	91.4	2.51	7.2
YOLOv6n	**92.0**	82.7	90.5	4.24	11.9
YOLOv8n	**92.0**	84.2	91.9	3.01	8.2
YOLOv10n	91.2	82.8	90.9	2.71	8.4
YOLOv11n	91.1	83.7	91.6	2.59	6.4
YOLO-FCAP	90.4	**84.3**	**92.5**	**0.81**	**5.5**

capability is not the most outstanding, it also exhibits good robustness. All in all, YOLO-FCAP strikes a favourable balance between lightweight and accuracy.

3.6. Yield prediction based on regression of test results

To evaluate the accuracy of YOLO-FCAP in yield estimation, this study randomly selected 20 citrus trees as experimental samples, each of which was a whole citrus tree photographed at a slightly distant location. The comparison results are shown in Table 6. It can be found that there are false detections in a few samples while there are omissions in all samples, which is caused by the fact that when the sample images are input into the model for detection, only one side of the citrus tree is displayed, resulting in numerous side and back targets being severely or even completely obscured. That is considered to be a reasonable part of the error. Fig. 15 presents the linear regression model established based on the YOLO detection counts and actual counts from 20 citrus trees, with an R^2 value of 0.983, an MAE of 0.95 and an RMSE of 1.20, this indicates that the regression model exhibits strong fitting performance, minimal prediction error and is highly simple and reliable. These results confirm the feasibility of YOLO-FCAP for the yield estimation task, which can be carried out efficiently by the linear regression equations established. Fig. 16 contains three randomly selected samples, and it can be seen that almost all citrus targets on the side facing the filming device are detected, although there are a few omissions in the first sample and one false detection in the third sample, which consider to be within acceptable limits.

4. Discussion

4.1. Model performance

The new lightweight citrus detection model was developed in this study. Based on the original network, this study optimized the network

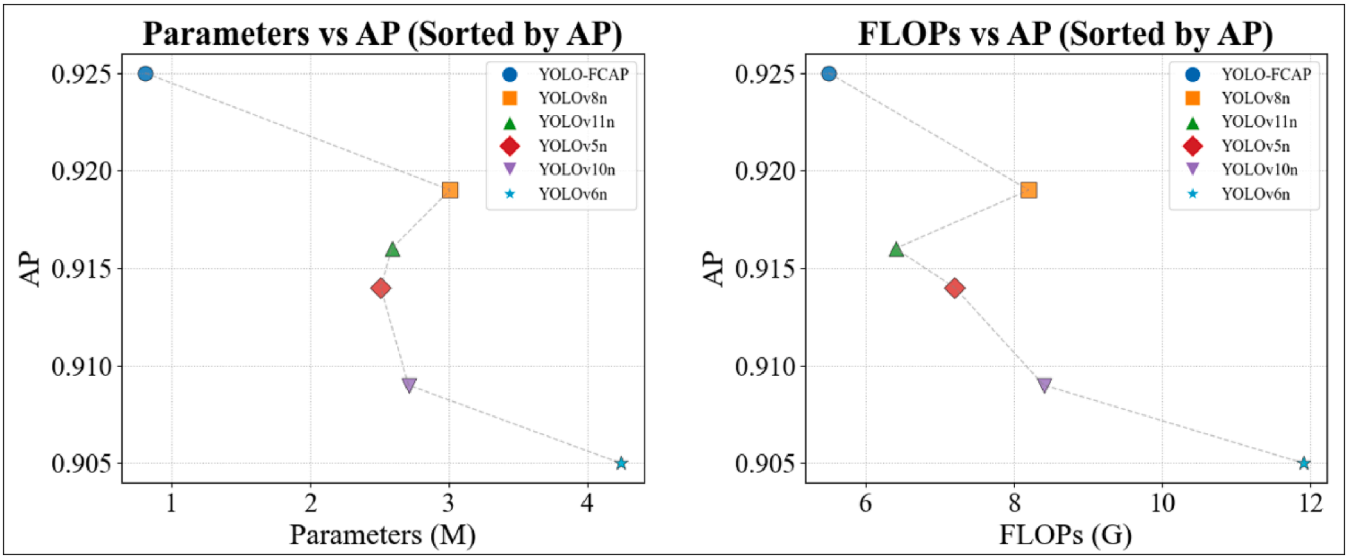

Fig. 12. Parameter vs AP and FLOPs vs AP Pareto curves.

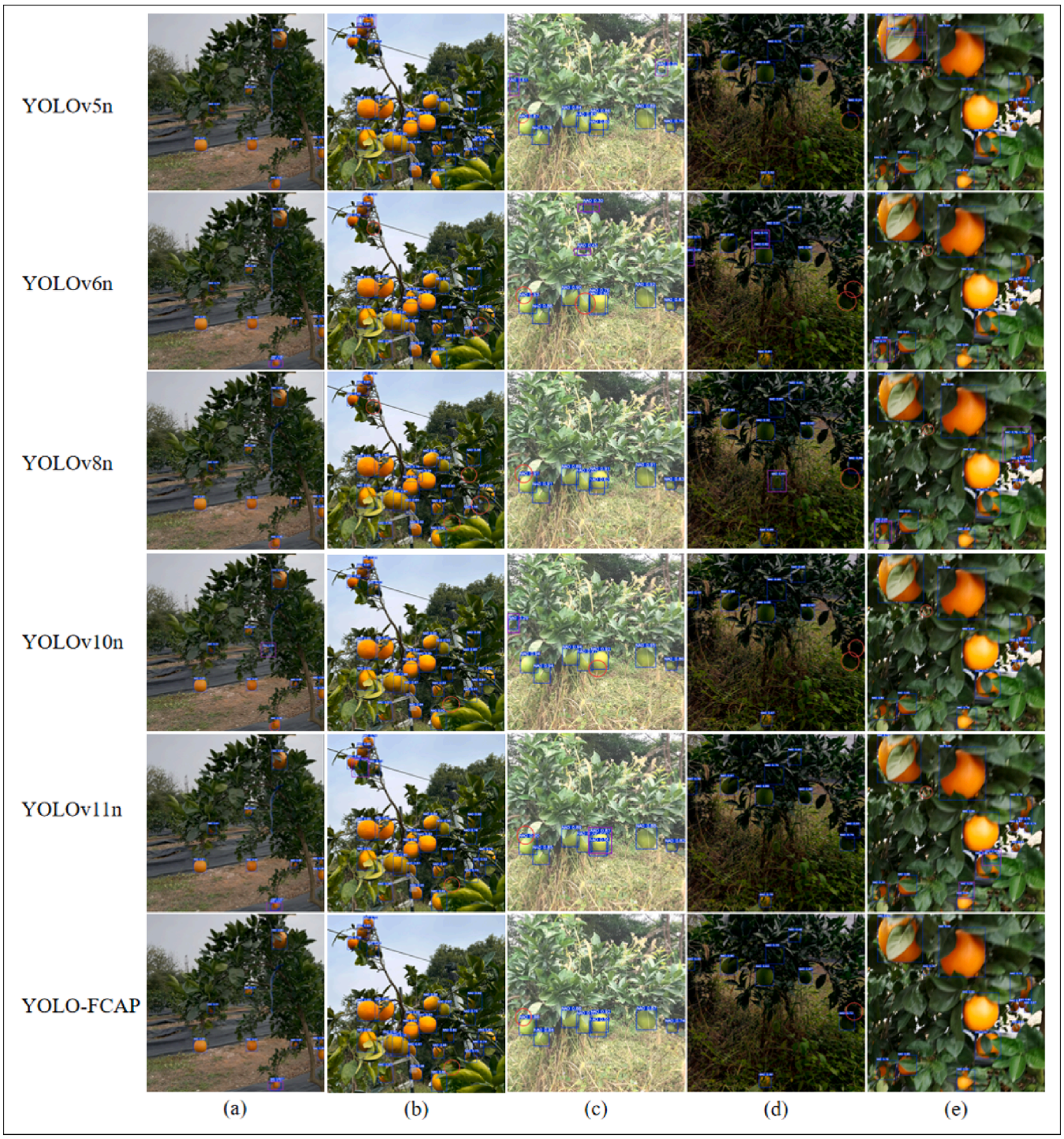

Fig. 13. Visualization results for each model. (a)cloudy and distant view. (b)different degrees of obstruction. (c)immature fruits in bright light. (d)immature fruits in dim light. (e)blurred scene.

scaling factor, the feature extraction module, the downsampling module, and the detection head network. These optimizations greatly reduced the parameters and FLOPs of the model while improving the AP and recall, which made the model easier to deploy on edge devices with excellent detection performance. Additionally, yield estimates were calculated using the improved model on 20 randomly selected citrus trees. By performing linear fitting between the detection model's predicted results and the actual counted quantities, the established regression equation enables high-precision prediction of citrus fruit counts for trees not physically counted. The experimental results fully demonstrate that yield prediction based on YOLO-FCAP achieves high accuracy. Compared to existing citrus detection models such as YOLOv5-CS [38] and YOLOv8-MEIN [39], YOLO-FCAP achieves a better balance between model lightweighting and detection performance. Experimental results demonstrate that this model outperforms the comparison models across multiple metrics, further validating the effectiveness of the proposed optimization strategy. This achievement significantly advances the practical deployment of automated citrus detection systems

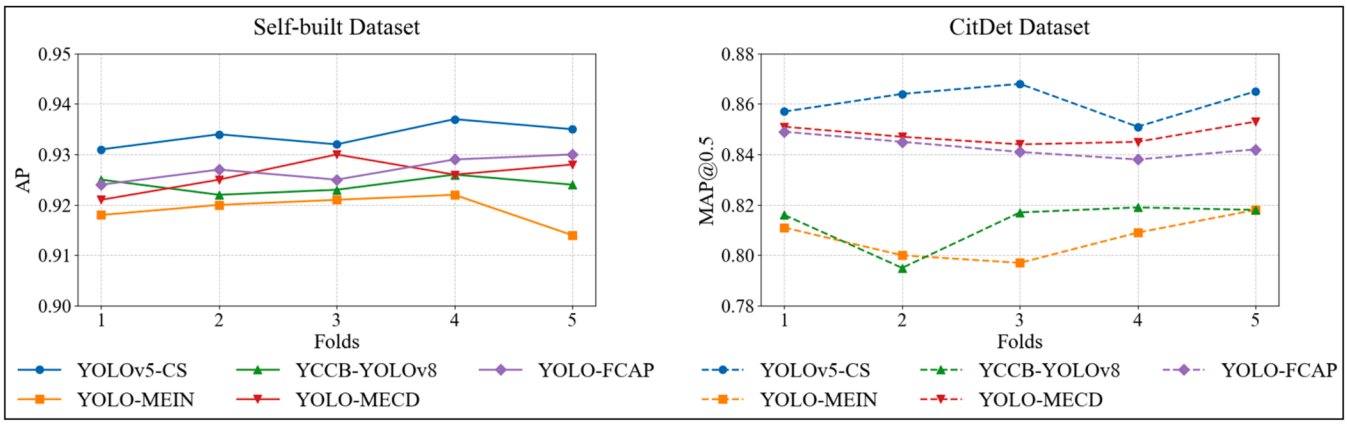

Fig. 14. Results of five-fold cross-validation.

Table 5
Comparative experimental results of different citrus detection models.

Model	Self-built Dataset			CitDet Dataset			Model Size(MB)	FPS
	P(%)	R(%)	AP (%)	P(%)	R(%)	mAP0.5(%)		
YOLOv5-CS [38]	**92.9**	**83.1**	**93.3**	**87.2**	**76.6**	**86.1**	71.51	82
YOLOv8-MEIN [39]	91.9	80.3	91.9	83.5	71.9	80.7	5.81	109
YCCB-YOLOv8 [40]	91.2	82.8	92.4	84.5	72.3	81.3	5.41	97
YOLO-MECD [41]	91.8	82.4	92.5	85.7	75.2	84.8	4.66	101
YOLO-FCAP	92.1	82.6	92.7	85.1	74.9	84.3	**1.95**	**121**

Table 6
Counting results of yield estimation experiments.

Tree Number	Actual Count	Forecast Production	Misdetection	Omission
(1)	33	27	0	6
(2)	36	29	0	7
(3)	21	17	0	4
(4)	18	15	1	4
(5)	20	19	1	2
(6)	22	19	0	3
(7)	16	13	0	3
(8)	31	27	2	6
(9)	11	7	1	5
(10)	30	25	1	6
(11)	29	24	0	5
(12)	26	20	0	6
(13)	31	27	0	4
(14)	27	22	2	7
(15)	17	13	0	4
(16)	26	20	0	6
(17)	44	38	0	6
(18)	46	38	0	8
(19)	15	12	0	3
(20)	34	29	0	5

in resource-constrained environments.

4.2. Limitations

Although YOLO-FCAP demonstrates strong detection performance, it still faces challenges such as false and missed detections. Most of the missed detections were caused by over-occlusion or even complete occlusion of the target. In practical scenarios, altering the observation angle can partially or fully reveal previously occluded targets, demonstrating that incorporating multi-view information enhances the detection of occluded objects, thereby reducing the number of missed detections [43]. However, this study relies solely on smartphone cameras for image acquisition, which limits the range of viewpoints available and makes it difficult to obtain target information from more diverse angles.

Furthermore, this study was primarily conducted under relatively ideal experimental conditions, and the model's performance was validated under these conditions. However, actual orchard environments are complex and variable, with meteorological factors such as rainfall and haze being common occurrences that can significantly reduce image clarity and interfere with target feature extraction, making detection tasks particularly challenging [44].

To validate the model's generalization ability across different citrus varieties, this study employed two datasets encompassing multiple citrus varieties in the experiments described in Section 3.5. Due to the large number of citrus varieties and the limitations imposed by the experimental conditions and data acquisition channels, this study was unable to include all varieties, leaving comprehensive coverage of all citrus varieties as a practical challenge.

4.3. Recommendations for future research

To further enhance the practical application value and generalization capability of the YOLO-FCAP, future research can be conducted in the following directions:

1) Existing research indicates that many citrus yield prediction models employ Unmanned Aerial Vehicle (UAV) for image acquisition [45–47]. UAVs are cost-effective and convenient to deploy, allowing for flexible and frequent data collection directly within target areas, this flexibility allows for the selection of a unique and flexible viewpoint, facilitating the acquisition of more detailed information about the fruit trees [48]. Inspired by this, more accurate estimates by using devices such as UAV for yield prediction at more angles are suggested in subsequent studies.
2) To enhance the model's robustness in complex weather conditions, it is recommended to supplement the dataset with images of orchards captured in rain or fog during the data collection phase. Alternatively, relevant scene features can be strengthened through data augmentation techniques [49].

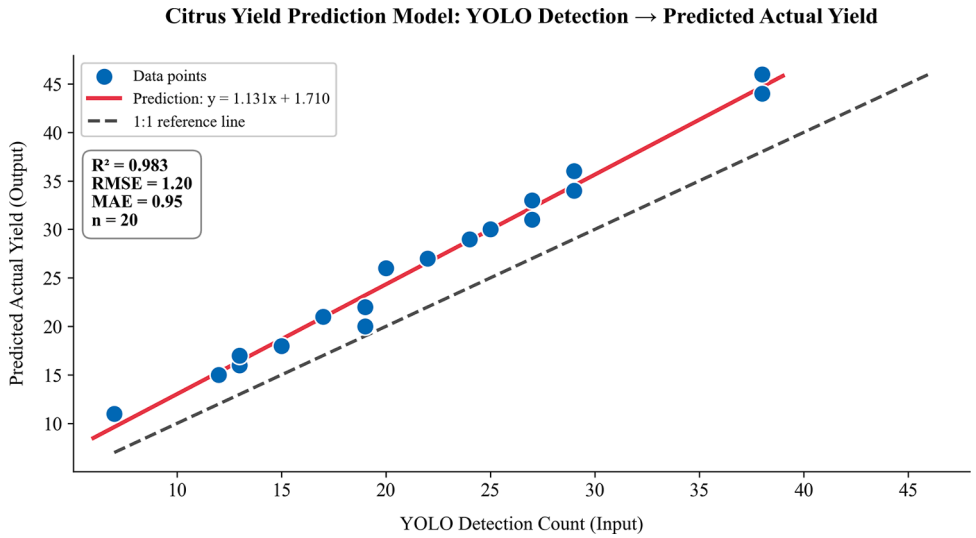

Fig. 15. Linear fit plot of estimated versus actual yield.

Fig. 16. Visualization of estimation results.

3) Different citrus varieties exhibit varying degrees of difference in visual traits such as fruit size, colour and shape [50]. To evaluate the model's generalization capability, future research may incorporate additional citrus varieties as experimental samples to validate the model's applicability across different citrus cultivars. In practical applications, when constrained by limited resources and unable to obtain sufficient samples of target varieties, transfer learning-based methods can be employed. This approach trains models using data from existing varieties and then transfers these models to other varieties not directly studied, thereby enhancing the model's generalization capability and adaptability under conditions of limited data [51].

5. Conclusions

Smart agriculture relies heavily on object detection technology, which provides great convenience for fruit picking, yield prediction and disease control through detection algorithms deployed on edge devices. Currently, the existing fruit detection algorithms perform optimally, but the model size needs to be optimized. While many optimization methods can improve detection accuracy, they often introduce more parameters and FLOPs, posing a challenge to model deployment on edge devices. Researchers are developing an algorithm with sufficient detection capability for difficult samples that is also lightweight enough for deployment on edge devices.

To address the aforementioned issue, this study proposed a lightweight citrus detection model named YOLO-FCAP based on YOLOv8n. Despite only occupying 1.95 MB of storage space, YOLO-FCAP maintains excellent detection capabilities and effectively meets the accuracy and real-time requirements. Furthermore, deployment in the vision system of edge devices is straightforward. This study also established a linear regression equation between the predicted and actual yields of the YOLO-FCAP model based on some of the samples to infer the citrus yields of other samples in the orchard, which provides technical support for realizing intelligent yield estimation in citrus orchards.

Ethical statement

Not applicable: This manuscript does not include human or animal research.

CRediT authorship contribution statement

Tiwei Zeng: Writing – review & editing, Supervision, Software, Resources, Project administration. **Jintao Tong:** Writing – review & editing, Visualization, Validation, Formal analysis. **Xudong Sun:** Validation, Resources, Investigation, Funding acquisition. **Jiacheng Liu:** Resources, Investigation, Data curation. **Xiangguo He:** Resources, Conceptualization. **Zhenzhen Guan:** Resources, Project administration. **Lingfeng Liu:** Supervision, Methodology. **Nan Jiang:** Resources, Funding acquisition. **Tao wan:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Jiangxi Province Early Vocational Young Scientific and Technological Talents Cultivation Program (20244BCE52165), the Jiangxi Province Natural Science Foundation (20242BAB25361, 20242BAB25066), the Jiangxi Province Double Thousand Plan of Jiangxi Province (JXSQ2023201010), the Jiangxi Province International Science and Technology Cooperation Project (2025), Jiangxi Provincial Department of Education Scientific Research Project(2025) and the Jiangxi Province Key Laboratory of Advanced Network Computing (2024SSY03071).

Data availability

Data will be made available on request.

References

[1] S. Suri, A. Singh, P.K. Nema, Current applications of citrus fruit processing waste: a scientific outlook, Appl. Food. Res. 2 (2022) 100050, https://doi.org/10.1016/j.afres.2022.100050.

[2] J. ZHANG, J. ZHANG, Y. SHAN, C. GUO, L. HE, L. ZHANG, W. LING, Y. LIANG, B. ZHONG, Effect of harvest time on the chemical composition and antioxidant capacity of Gannan navel orange (Citrus sinensis L. Osbeck 'Newhall') juice, J. Integr. Agric. 21 (2022) 261–272, https://doi.org/10.1016/S2095-3119(20)63395-0.

[3] M. Huang, C. Lai, Y. Liang, Q. Xiong, C. Chen, Z. Ju, Y. Jiang, J. Zhang, Improving the functional components and biological activities of navel orange juice through fermentation with an autochthonous strain lactiplantibacillus paraplantarum M23, Food. Bioproducts. Process. 149 (2025) 249–260, https://doi.org/10.1016/j.fbp.2024.11.027.

[4] B. Zhou, K. Wu, M. Chen, Detection of Gannan navel orange ripeness in natural environment based on YOLOv5-NMM, Agronomy 14 (2024) 910, https://doi.org/10.3390/agronomy14050910.

[5] Q. Liu, J. Lv, C. Zhang, MAE-YOLOv8-based small object detection of green crisp plum in real complex orchard environments, Comput. Electron. Agric. 226 (2024) 109458, https://doi.org/10.1016/j.compag.2024.109458.

[6] S.N. Khan, A.N. Khan, A. Tariq, L. Lu, N.A. Malik, M. Umair, W.A. Hatamleh, F. H. Zawaideh, County-level corn yield prediction using supervised machine learning, Eur. J. Remote. Sens. 56 (2023), https://doi.org/10.1080/22797254.2023.2253985.

[7] B.R. Sapkota, G.S. Baath, K.C. Flynn, K. Adhikari, C. Hajda, D.R. Smith, Machine learning algorithms for maize yield prediction with multispectral imagery: assessing robustness across varied growing environments, Sci. Remote. Sens. 12 (2025) 100267, https://doi.org/10.1016/j.srs.2025.100267.

[8] G. Zhu, C. Zhao, L. Zhou, Z. Li, H. Zhu, Winter wheat yield prediction at a county scale using time series variation features of remote sensing spectra and machine learning, Eur. J. Agronomy. 170 (2025) 127751, https://doi.org/10.1016/j.eja.2025.127751.

[9] F. Imtiaz, A.A. Farooque, G.S. Randhawa, X. Wang, T.J. Esau, S.E. Hashemi Garmdareh, B. Acharya, Optimizing potato yield mapping and prediction: integrating satellite-based remote sensing and machine learning for sustainable agriculture, Comput. Electron. Agric. 237 (2025) 110636, https://doi.org/10.1016/j.compag.2025.110636.

[10] F. Xiao, H. Wang, Y. Xu, R. Zhang, Fruit detection and recognition based on deep learning for automatic harvesting: an overview and review, Agronomy 13 (2023) 1625, https://doi.org/10.3390/agronomy13061625.

[11] J. Lin, Y. Zhao, S. Wang, Y. Tang, YOLO-DA: an efficient YOLO-based detector for remote sensing object detection, IEEE Geosci. Remote. Sens. Lett. 20 (2023) 1–5, https://doi.org/10.1109/LGRS.2023.3303896.

[12] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587. https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html.

[13] R. Girshick, Ross, Fast R-CNN, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448. https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html.

[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, 2015. https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

[15] F. Gao, L. Fu, X. Zhang, Y. Majeed, R. Li, M. Karkee, Q. Zhang, Multi-class fruit-on-plant detection for apple in SNAP system using faster R-CNN, Comput. Electron. Agric. 176 (2020) 105634, https://doi.org/10.1016/j.compag.2020.105634.

[16] P. Siricharoen, W. Yomsatieankul, T. Bunsri, Fruit maturity grading framework for small dataset using single image multi-object sampling and mask R-CNN, Smart. Agric. Technol. 3 (2023) 100130, https://doi.org/10.1016/j.atech.2022.100130.

[17] H. Qian, H. Wang, S. Feng, S. Yan, FESSD:SSD target detection based on feature fusion and feature enhancement, J. Real. Time. Image. Process. 20 (2023) 2, https://doi.org/10.1007/s11554-023-01258-y.

[18] L. Xu, Y. Wang, X. Shi, Z. Tang, X. Chen, Y. Wang, Z. Zou, P. Huang, B. Liu, N. Yang, Z. Lu, Y. He, Y. Zhao, Real-time and accurate detection of citrus in complex scenes based on HPL-YOLOv4, Comput. Electron. Agric. 205 (2023) 107590, https://doi.org/10.1016/j.compag.2022.107590.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot MultiBox detector, in: 2016: pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.

[20] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html.

[21] Q. An, K. Wang, Z. Li, C. Song, X. Tang, J. Song, Real-time monitoring method of strawberry fruit growth State based on YOLO improved model, IEEE Access 10 (2022) 124363–124372, https://doi.org/10.1109/ACCESS.2022.3220234.

[22] J. Chen, H. Liu, Y. Zhang, D. Zhang, H. Ouyang, X. Chen, A multiscale lightweight and efficient model based on YOLOv7: applied to Citrus orchard, Plants 11 (2022) 3260, https://doi.org/10.3390/plants11233260.

[23] J. Lu, P. Chen, C. Yu, Y. Lan, L. Yu, R. Yang, H. Niu, H. Chang, J. Yuan, L. Wang, Lightweight green citrus fruit detection method for practical environmental applications, Comput. Electron. Agric. 215 (2023) 108205, https://doi.org/10.1016/j.compag.2023.108205.

[24] B. Gu, C. Wen, X. Liu, Y. Hou, Y. Hu, H. Su, Improved YOLOv7-tiny complex environment citrus detection based on lightweighting, Agronomy 13 (2023) 2667, https://doi.org/10.3390/agronomy13112667.

[25] Y. Liao, L. Li, H. Xiao, F. Xu, B. Shan, H. Yin, YOLO-MECD: citrus detection algorithm based on YOLOv11, Agronomy 15 (2025) 687, https://doi.org/10.3390/agronomy15030687.

[26] Z. Fan, D. Lu, M. Liu, Z. Liu, Q. Dong, H. Zou, H. Hao, Y. Su, YOLO-PDGT: a lightweight and efficient algorithm for unripe pomegranate detection and counting, Measurement 254 (2025) 117852, https://doi.org/10.1016/j.measurement.2025.117852.

[27] H. Cheng, F. Wan, G. Lei, L. Xu, GCS-YOLO: a lightweight strawberry disease detection algorithm based on improved YOLOv8, in: 2023 5th International Conference on Frontiers Technology of Information and Computer (ICFTIC), 2023, pp. 1071–1076, https://doi.org/10.1109/ICFTIC59930.2023.10455857. IEEE.

[28] Z. Liu, R.M. Rasika D Abeyrathna, R. Mulya Sampurno, V. Massaki Nakaguchi, T. Ahamed, Faster-YOLO-AP: a lightweight apple detection algorithm based on improved YOLOv8 with a new efficient PDWConv in orchard, Comput. Electron. Agric. 223 (2024) 109118, https://doi.org/10.1016/j.compag.2024.109118.

[29] H. Li, P. Xiong, J. An, L. Wang, Pyramid attention Network for semantic segmentation, (2018). http://arxiv.org/abs/1805.10180.

[30] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: faster and better learning for bounding box regression, Proc. AAAI Conf. Artif. Intell. 34 (2020) 12993–13000, https://doi.org/10.1609/aaai.v34i07.6999.

[31] G. Liu, A. Dundar, K.J. Shih, T.-C. Wang, F.A. Reda, K. Sapra, Z. Yu, X. Yang, A. Tao, B. Catanzaro, Partial convolution for padding, inpainting, and image synthesis, IEEE. Trans. Pattern. Anal. Mach. Intell. (2022) 1–15, https://doi.org/10.1109/TPAMI.2022.3209702.

[32] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, 2015, pp. 448–456, pmlr, https://proceedings.mlr.press/v37/ioffe15.html.

[33] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Networks 107 (2018) 3–11, https://doi.org/10.1016/j.neunet.2017.12.012.

[34] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13713–13722. https://openaccess.thecvf.com/content/CVPR2021/html/Hou_Coordinate_Attention_for_Efficient_Mobile_Network_Design_CVPR_2021_paper.html.

[35] Q.-L. Zhang, Y.-B. Yang, SA-Net: shuffle attention for deep convolutional neural networks, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 2235–2239, https://doi.org/10.1109/ICASSP39728.2021.9414568. IEEE.

[36] D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, Z. Huang, Efficient Multi-scale attention module with cross-spatial learning, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5, https://doi.org/10.1109/ICASSP49357.2023.10096516. IEEE.

[37] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-net: efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11534–11542. https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_ECA-Net_Efficient_Channel_Attention_for_Deep_Convolutional_Neural_Networks_CVPR_2020_paper.html.

[38] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, S. Liu, Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system, Sensors 22 (2022) 576, https://doi.org/10.3390/s22020576.

[39] K. Yue, P. Zhang, L. Wang, Z. Guo, J. Zhang, Recognizing citrus in complex environment using improved YOLOv8n, Nongye Gongcheng Xuebao Trans. Chin. Soc. Agric. Eng. 40 (2024) 152–158, https://doi.org/10.11975/j.issn.1002-6819.202401118.

[40] G. Ang, T. Zhiwei, M. Wei, S. Yuepeng, R. Longlong, F. Yuliang, Q. Jianping, X. Lijia, Fruits hidden by green: an improved YOLOV8n for detection of young citrus in lush citrus trees, Front. Plant. Sci. 15 (2024), https://doi.org/10.3389/fpls.2024.1375118.

[41] Y. Liao, L. Li, H. Xiao, F. Xu, B. Shan, H. Yin, YOLO-MECD: citrus detection algorithm based on YOLOv11, Agronomy 15 (2025) 687, https://doi.org/10.3390/agronomy15030687.

[42] J.A. James, H.K. Manching, M.R. Mattia, K.D. Bowman, A.M. Hulse-Kemp, W.J. Beksi, CitDet, (2024). https://doi.org/10.18738/T8/QFVHQ5.

[43] D. Rapado-Rincón, E.J. van Henten, G. Kootstra, Development and evaluation of automated localisation and reconstruction of all fruits on tomato plants in a greenhouse based on multi-view perception and 3D multi-object tracking, Biosyst. Eng. 231 (2023) 78–91, https://doi.org/10.1016/j.biosystemseng.2023.06.003.

[44] H. Wu, X. Mo, S. Wen, K. Wu, Y. Ye, Y. Wang, Y. Zhang, DNE-YOLO: a method for apple fruit detection in diverse natural environments, J. King Saud Univ. Comput. Inf. Sci. 36 (2024) 102220, https://doi.org/10.1016/j.jksuci.2024.102220.

[45] O.E. Apolo-Apolo, J. Martínez-Guanter, G. Egea, P. Raja, M. Pérez-Ruiz, Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV, Eur. J. Agronomy. 115 (2020) 126030, https://doi.org/10.1016/j.eja.2020.126030.

[46] Y. Ampatzidis, V. Partel, B. Meyering, U. Albrecht, Citrus rootstock evaluation utilizing UAV-based remote sensing and artificial intelligence, Comput. Electron. Agric. 164 (2019) 104900, https://doi.org/10.1016/j.compag.2019.104900.

[47] Y. Zhu, F. Liu, Y. Zhao, Q. Gu, X. Zhang, Citrus yield estimation for individual trees integrating pruning intensity and image views, Eur. J. Agronomy. 161 (2024) 127349, https://doi.org/10.1016/j.eja.2024.127349.

[48] Y. Guo, W. Zhou, Y.H. Fu, F. Hao, X. Zhang, L. Xu, J. Liu, Y. He, SegNeXt-RCMSCA: an improved SegNeXt network for detecting winter wheat lodging from UAS RGB images, Smart. Agric. Technol. 12 (2025) 101230, https://doi.org/10.1016/j.atech.2025.101230.

[49] L. Ma, L. Zhao, Z. Wang, J. Zhang, G. Chen, Detection and counting of small target apples under complicated environments by using improved YOLOv7-tiny, Agronomy 13 (2023) 1419, https://doi.org/10.3390/agronomy13051419.

[50] F. Deng, J. Chen, L. Fu, J. Zhong, W. Qiaoi, J. Luo, J. Li, N. Li, Real-time citrus variety detection in orchards based on complex scenarios of improved YOLOv7, Front. Plant. Sci. 15 (2024), https://doi.org/10.3389/fpls.2024.1381694.

[51] N. Wu, F. Liu, F. Meng, M. Li, C. Zhang, Y. He, Rapid and accurate varieties classification of different crop seeds under sample-limited condition based on hyperspectral imaging and deep transfer learning, Front. Bioeng. Biotechnol. 9 (2021), https://doi.org/10.3389/fbioe.2021.696292.