Apple flower phenotype detection method based on YOLO-FL and application of intelligent flower thinning robot

Ang Gao[1,2], Yonghui Du[1,2], Yuqiang Li[1,2], Yuepeng Song[1*], Longlong Ren[2*]

(1. *College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai'an 827101, China*;
2. *Shandong Agricultural Equipment Intelligent Engineering Laboratory, Tai'an 827101, China*)

Abstract: In intelligent flower thinning robot applications, accurate and efficient apple flower detection is the key to realizing automated fruit tree thinning operations. However, complex orchard environments and diverse flower characteristics pose many challenges to apple blossom detection, such as shading, light variations, flower densities, and so on. To address these challenges, this study proposes an improved model based on the YOLO target detection framework which is named the YOLO-FL apple flower detection model. The model enhances the feature extraction capability by optimizing the Backbone part with EC3DFM structure, while introducing MFEM structure in the Neck part to improve the feature fusion effect. In addition, the ABRLoss loss function is used to optimize the prediction results of the prediction frame, and it also adds the SimAM attention mechanism to the middle two detection heads in the Neck part, which further improves the detection performance of the model. The experimental results respectively show that YOLO-FL achieves 74.63%, 73.82%, and 79.97% accuracy, recall, and mean average precision on the test set, which shows significant improvement over the benchmark model. Meanwhile, the model size was only 4693 kB, demonstrating high efficiency and storage advantages. After deploying the YOLO-FL model to the intelligent flower thinning robot, the frame rate of the test image was 40.7 FPS, the average missed detection rate was 7.26%, the false detection rate was 6.89%, and the model was able to efficiently complete the apple flower detection in the complex orchard environment. This study provides an effective solution and technical support for the application of image recognition technology in intelligent flower thinning robots.

Keywords: apple flower, YOLO-FL, deep learning, intelligent flower thinning robot
DOI: 10.25165/j.ijabe.20251803.9110

Citation: Gao A, Du Y H, Li Y Q, Song Y P, Ren L L. Apple flower phenotype detection method based on YOLO-FL and application of intelligent flower thinning robot. Int J Agric & Biol Eng, 2025; 18(3): 236–246.

1 Introduction

Scientific management of apple trees is important to ensure high yield and quality fruit[1-3]. In the application of intelligent flower thinning robots, exact and highly efficient apple flower detection is the core link to realize automated flower thinning operations. As a key indicator in the apple growth process, the number and density of apple flowers directly affect the final yield and quality of the fruit. By precisely identifying and monitoring apple flowers, the intelligent flower thinning robot can obtain information about the distribution of flowers, so as to carry out precise flower thinning operations, optimize the use of resources, and improve management efficiency. However, many factors such as complex orchard environments, morphological diversity of apple flowers, light variations, and flower densities pose a number of challenges for automatic detection of apple flowers, such as problems of shading, uneven light, and flower overlap. These factors make the existing automatic detection methods deficient in accuracy and real-time performance in high-density, diverse, and complex backgrounds, limiting the effectiveness of intelligent flower thinning robots. To cope with the above challenges, apple flower recognition approaches that are based on computer vision and deep learning have been gradually becoming a research hotspot.

Recent studies have applied computer vision algorithms and deep learning models to the recognition of apple flowers[4,5]. Chen et al. proposed an apple flower recognition and detection method based on the YOLOv5 deep learning model, which was enhanced through data augmentation techniques. This method effectively detects the growth status of apple flowers and quantifies flowering intensity. The average detection accuracy of the improved model for apple flower growth state was 2.53% with a detection time of 13 ms per image[6]. Additionally, Rakesh Mohan Datt et al. proposed a convolutional neural network (CNN) model for the identification of eight different periods of the apple crop, i.e., fruit ripening and senescence, shoot budding, flowering, leaf development, dormancy initiation, fruit development, bud development, and inflorescence emergence. The dataset was extended to 7000 photographs using data enhancement techniques, and the chosen recognition model was the Inception-v3 model. The microscopic F1 score calculated by the proposed model was 0.98[7].

Zhou et al.[8] investigated distant imagery of apple trees in bloom, captured within a natural setting, as their research focus. They implemented the slice-assisted hyper-inference (SAHI) algorithm to augment the dataset and incorporated the Swin Transformer-tiny to modify the YOLOX Backbone network, thereby establishing the S-YOLO model for apple flower detection. Their findings demonstrate that S-YOLO surpassed YOLOX-s in terms of detection precision across all four floral stages.

Received date: 2024-06-02 **Accepted date:** 2025-03-10
Biographies: **Ang Gao**, PhD candidate, research interest: agricultural informatization and intelligent control technology, Email: 2019110103@sdau.edu.cn; **Yonghui Du**, Master, research interest: intelligent agricultural equipment, Email: 2023120708@sdau.edu.cn; **Yuqiang Li**, Master, research interest: intelligent agricultural equipment, Email: liyq0817@163.com.
***Corresponding author: Yuepeng Song**, PhD, Professor, research interest: intelligent agricultural machinery equipment, Email: uptonsong@163.com; **Longlong Ren**, PhD, Lecturer, research interest: intelligent orchard agricultural machinery equipment, Email: renlonglong@sdau.edu.cn.

Specifically, S-YOLO exhibited improvements of 10.00%, 9.10%, 13.10%, and 7.20% over YOLOX-s in mean average precision across all categories (mAPALL), small objects (mAPS), medium objects (mAPM), and large objects (mAPL), respectively. These outcomes underscore the efficacy of the transformer-based S-YOLO approach in monitoring apple flower growth stages, enabling applications such as flower counting, percentage analysis, precise timing of blossoming, and quantitative assessment of bloom intensity.

Despite the successful application of various deep learning models for apple flower detection, which achieved favorable comprehensive evaluation metrics through structural optimizations, the deployment of intelligent thinning robots in real orchards still demanded higher levels of accuracy and processing speed for target image recognition[9]. To address the challenges faced by intelligent thinning robots in apple flower detection within complex orchard environments, this study puts forward an improved model which is based on the YOLO object detection framework: the YOLO-FL apple blossom detection model.

This model was trained and tested on a custom-built apple flower dataset captured in natural environments. It optimized the Backbone of the YOLO framework by incorporating the EC3DFM structure to enhance feature extraction capabilities. Additionally, the Neck component was augmented with the MFEM structure, and the ABRLoss function was employed. To further improve the detection performance, the SimAM attention mechanism was integrated into the two intermediate detection heads. The experimental results showed that the YOLO-FL model was better than the baseline model in terms of accuracy, recall, and mean average precision (mAP), while maintaining a compact model size and high processing speed. Therefore, the YOLO-FL model provided rapid and accurate apple flower recognition capability for intelligent thinning robots, offering strong technical support for the advancement of precision agriculture and smart farming management. The main contributions of this study are as follows:

1) Development of an apple flower dataset for complex orchard environments: A diverse set of apple flower images, captured under various natural environmental conditions, including multiple angles, different times of day, and varying flower densities, was collected and annotated. This dataset provides high-quality data support for the training and testing of the proposed model.

2) Development of the YOLO-FL apple flower detection model: An improved model, YOLO-FL, was developed based on YOLO object detection framework. The model integrated EC3DFM and MFEM structures to optimize feature extraction and fusion capabilities. Additionally,ABRLoss function and SimAM attention mechanism were introduced, dramatically improving the accuracy and efficiency of the inspection process.

3) Efficient deployment of the YOLO-FL model on intelligent flower thinning robots: The YOLO-FL model was efficiently deployed in real orchard environments, achieving a processing speed of 40.7 FPS. The model demonstrated a detection performance with false-negative and false-positive rates of 7.26% and 6.89%, respectively, significantly improving the effectiveness of automated thinning operations.

2 Materials and methods

To achieve rapid and high-precision apple flower detection for the flower thinning robot, this study proposed the YOLO-FL apple flower detection model. It was deployed on a rugged embedded industrial control computer (Nuvo-8003) and tested in the field on the flower thinning robot. The overall research framework is shown in Figure 1.

2.1 Image acquisition and pre-processing

The image data were collected at the apple experimental fields of Shandong Agricultural University and the Shandong Provincial Institute of Agricultural Fruit Trees in two phases. The first phase, conducted in early April 2022, involved the collection of 657 images of Fuji apple flowers using a Huawei Honor 20s smartphone. The second phase, conducted in early April 2023, involved the collection of 1288 images of the same apple variety using a Xiaomi 6X smartphone. These two data collection phases ensured coverage of diverse lighting conditions, various angles, and different flower densities, thereby enhancing the dataset's diversity and the model's generalization capability.

During the pre-processing stage, the collected images were first resized to 608×608 pixels for normalization. Subsequently, several data augmentation techniques, including random cropping, rotation, brightness adjustment, and horizontal flipping, were applied to expand the dataset. Finally, the images were accurately labeled using the labelImg tool to construct a comprehensive apple flower dataset in a natural background, as shown in Figure 2.

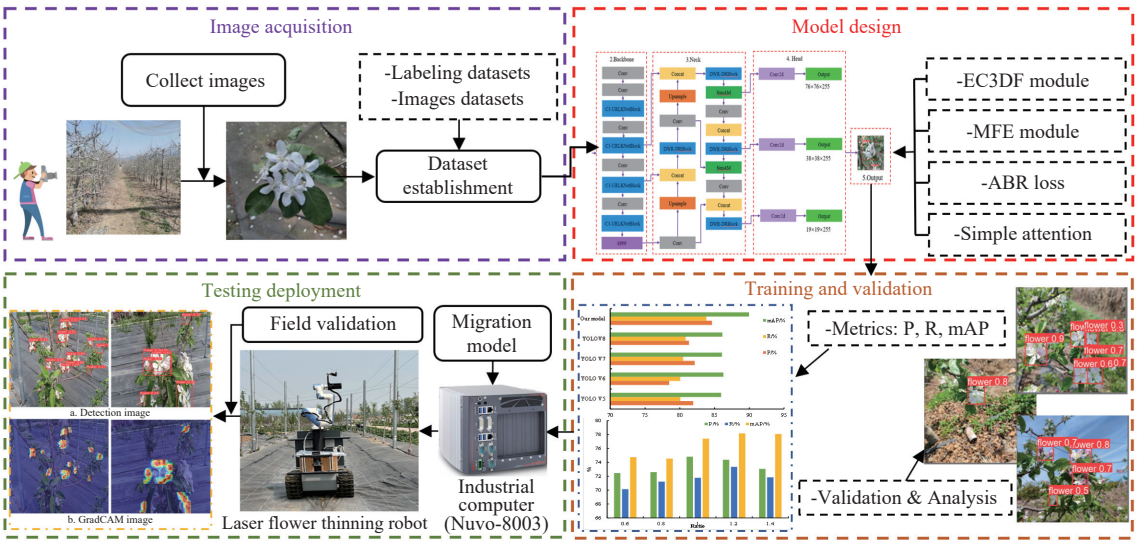


Figure 1 Overall research route of this study

Figure 2 Partial image of the dataset

2.2 Development of the YOLO-FL: A fast and lightweight apple flower detection model

Given that the apple flower images were captured in natural orchard environments, and considering that the flower thinning robot must also operate in similar real-world conditions, the complex working environment demanded higher performance for apple flower detection. To address these challenges, this study proposes the YOLO-FL apple flower detection model, based on the YOLO framework. A schematic of the model architecture is shown in Figure 3.

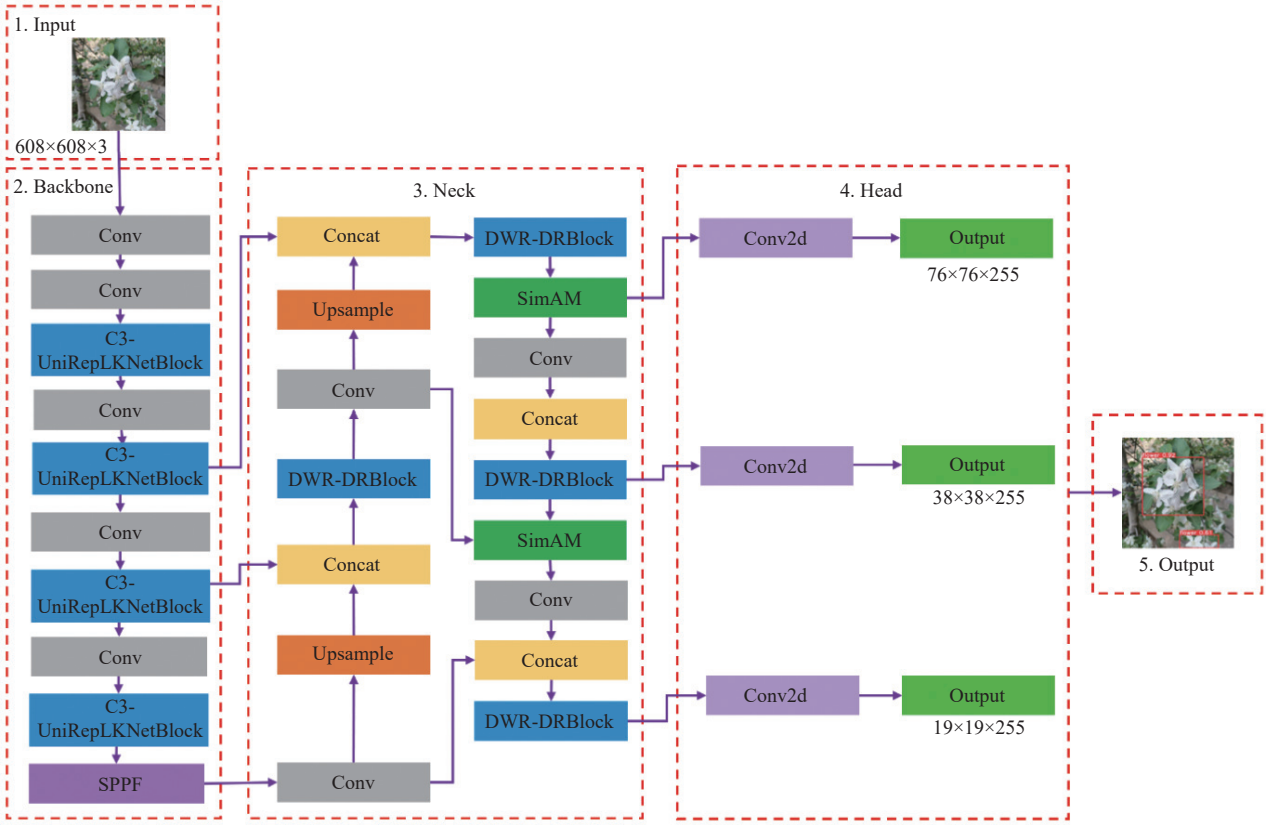


Figure 3 Structure of YOLO-FL apple flower detection model

In the YOLO-FL model, both the Backbone and Neck components of the original YOLO framework were restructured. Additionally, the loss function was redesigned, and an attention mechanism was incorporated to enhance detection performance. These modifications were aimed at improving the model's robustness and efficiency in detecting apple flowers in dynamic and cluttered natural environments.

1) EC3DFM (Enhanced C3 Deep Feature Module)

In YOLOv5, the C3 module implemented the CSP (Cross Stage Partial) Bottleneck structure through a specific combination of convolutional layers, which includes two 1×1 convolutional layers, multiple Bottleneck modules, channel concatenation, and a final 1×1 convolutional layer (as shown in Figure 4a). This structure facilitated accurate object detection and localization by combining different levels of C3 modules. However, the C3 module still had limitations in extracting deep features, especially when handling complex scenes, where its feature representation capacity may be insufficient[10-12].

To address this bottleneck, we propose the Enhanced C3 Deep Feature Module (EC3DFM), which is an optimized deep fusion of the C3 module and the UniRepLKNetBlock module (as shown in Figure 4d). UniRepLKNet is a novel large convolution kernel

neural network (ConvNet) architecture, illustrated in Figures 4b and 4c. This module introduces four key design principles for convolutional networks, including the use of efficient SE (Squeeze-and-Excitation) modules to enhance model depth, the application of extended reparameterization techniques to optimize large convolution kernels, and the replacement of additional large convolutional layers with 3×3 convolutions during deep expansion[13]. Additionally, the Lark Block in UniRepLKNet significantly expanded the receptive field and enhanced feature extraction depth by combining dilated reparameterization blocks, SE modules, Feedforward Networks (FFN), and Batch Normalization (BN) layers[14].

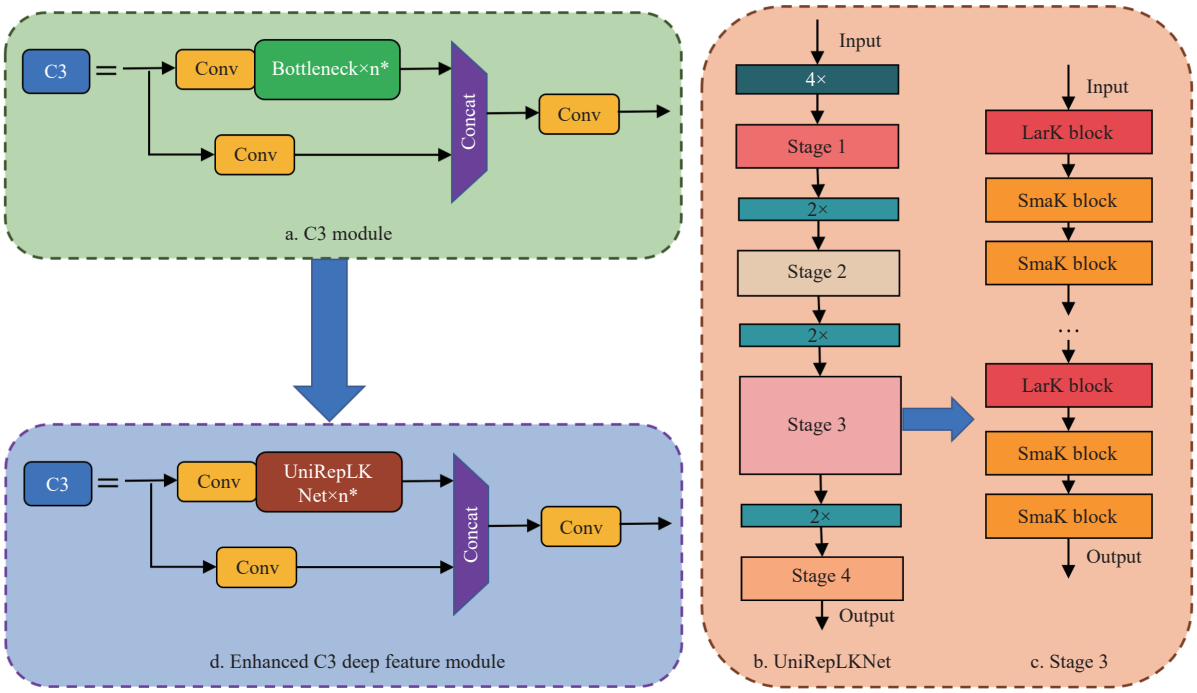


Figure 4 Structure of the EC3DFM

In the Backbone network of this study, the EC3DFM was employed for feature extraction. The innovative aspects of EC3DFM are primarily reflected in the following areas:

(1) Multi-Scale Feature Enhancement: The C3 module utilized a branching structure and channel concatenation to achieve cross-layer feature fusion, thereby improving the model's ability to capture targets at multiple scales.

(2) Deep Feature Abstraction Optimization: The UniRepLKNetBlock, incorporating dilated convolutions and SE modules, effectively expanded the receptive field of the model, thereby enhancing the expressive power of deep features.

(3) Lightweight Structure and Performance Balance: EC3DFM not only enhanced deep feature extraction capabilities but also maintained manageable computational complexity, thereby enabling its application in environments with limited computational resources.

2) Multi-Scale Feature Enhancement Module (MFEM)

The DWRSeg module optimized the network's receptive field requirements at different stages. As depicted in Figure 5a, this module adopted a three-branch architecture, where each branch employs depthwise separable filters with diverse dilation factors to gather features across different scales. This configuration allowed for the concurrent acquisition of both intricate local textures and expansive global contexts, ultimately fostering a richer and more comprehensive feature representation[15]. Unlike traditional residual modules, where branches share an initial convolution, DWRSeg assigns independent convolution operations to each branch. This "regional residual" design enhanced the module's flexibility and adaptability, allowing each branch to focus on receptive fields of different scales based on the specific task requirements. By adopting this approach, the network is capable of adaptively tailoring its feature extraction strategy in accordance with the input data's attributes and the task's requirements, resulting in notable enhancements in model performance across various contexts. In contrast to other residual blocks, DWRSeg efficiently mitigates the module's non-linearity, thereby lowering computational complexity and substantially boosting operational efficiency[16].

The dilated reparameterization block in UniRepLKNet further expands the receptive field, enhancing its ability to capture global information. This module incorporates multiple convolutional layers with varying dilation rates (as shown in Figure 6a, with K=9 as an example). By integrating convolution kernels of varying sizes and dilation rates, the network's capability for global feature extraction is substantially enhanced, leading to a deeper and more nuanced feature representation. Furthermore, the seamless integration of batch normalization within the convolutional layers minimizes both the parameter footprint and computational overhead of the model. This design both increased deployment efficiency and provided a novel approach for performance optimization in resource-constrained environments.

To further enhance feature extraction capabilities, a Multi-Scale Feature Enhancement Module (MFEM) was proposed, as shown in Figure 5b. This module integrated the characteristics of DWRSeg and the dilated reparameterization block, employing a four-branch structure. The first branch utilized a convolution with a kernel size of 3, while the second and third branches used dilated reparameterization block (DRB) convolutions with a kernel size of 5. The fourth branch employed a DRB convolution with a kernel size of 7. By using convolution kernels of varying sizes and dilation rates, MFEM extracted features with different receptive fields,

enabling more comprehensive feature representation. During deployment, the integration of batch normalization with convolutional layers facilitated the optimization of both model footprint and computational demands, rendering MFEM exceptionally well-suited for real-time implementations in environments with limited resources.

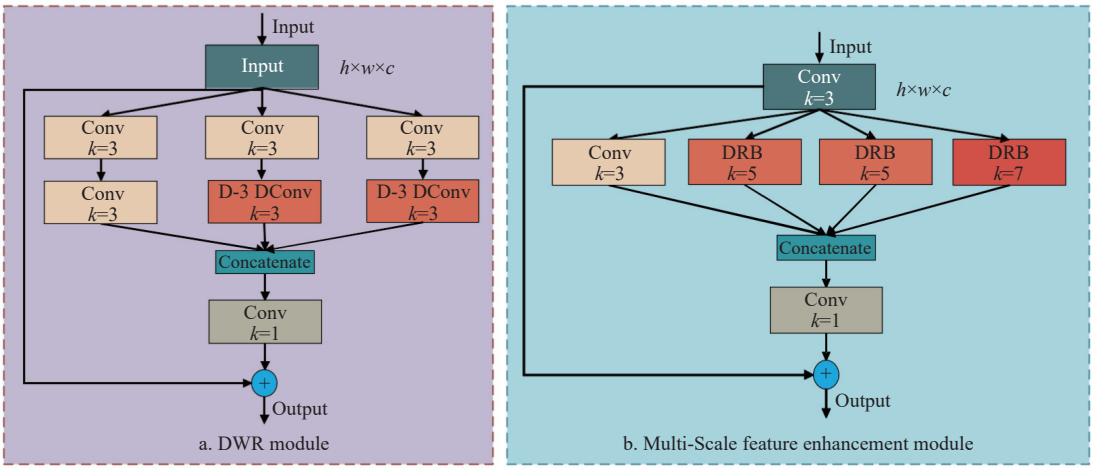


Figure 5 Structure of DWR and MFEM modules

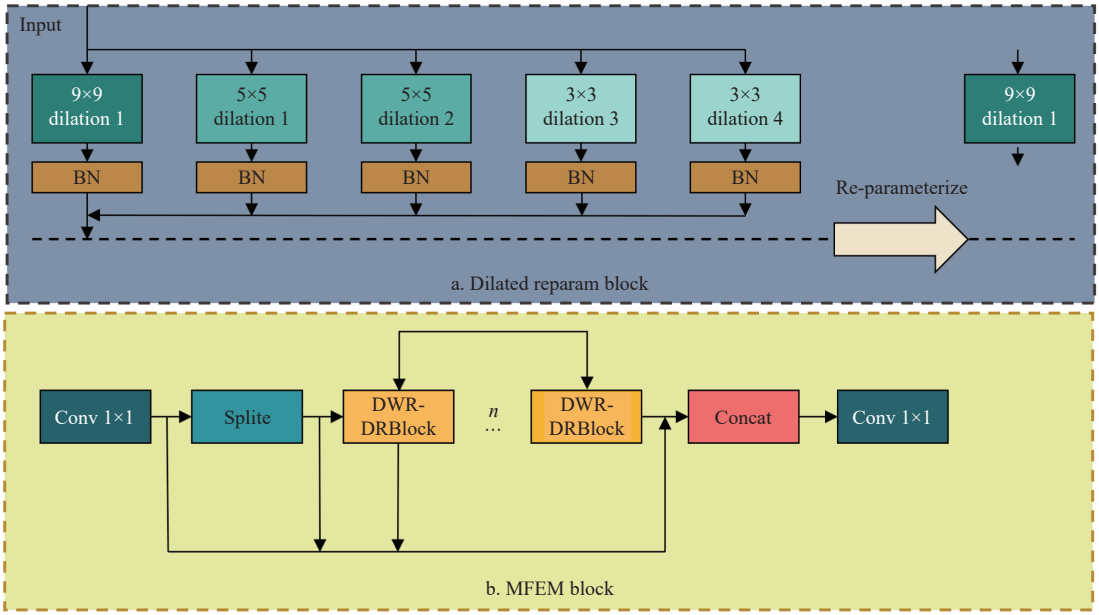


Figure 6 Structure of DRBlock and MFEM block modules

In the Neck network of this study, the Block within the C2f module was substituted with MFEM for the purpose of extracting features, as illustrated in Figure 6b. The innovations of this approach are primarily reflected in the following aspects:

(1) Independent convolutional branch design: A residual structure was introduced for each branch, assigning independent convolution operations that enhanced the flexibility and adaptability of the network.

(2) Combination of multi-scale convolutions and dilated reparameterization: The use of multiple convolutional layers with varying dilation rates significantly expanded the receptive field, thereby strengthening the network's ability to capture global information.

(3) Integration of batch normalization: This integration reduced both model size and computational cost, enhancing efficiency and adaptability in resource-constrained environments.

3) Adaptive Boundary Refinement Loss (ABRLoss)

In object detection, bounding box regression served an important role in enhancing model performance by optimizing the alignment of predicted and ground truth bounding boxes. However, conventional Intersection over Union (IoU) loss functions exhibited limitations in providing adequate gradient information under conditions of minimal overlap between predicted and ground truth boxes, leading to sluggish model convergence, particularly for instances characterized by low IoU scores.

To tackle this challenge, the Adaptive Boundary Refinement Loss (ABRLoss) was introduced. This loss function harnesses the strengths of both the Inner-IoU and MPDIoU methods, facilitating swift convergence for high IoU samples while augmenting the regression performance for those with low IoU values. Specifically, Inner-IoU focuses on refining the internal area of the bounding box, offering a refined measure of overlap, as elaborated in Equations (1)-(8). Compared to traditional IoU, Inner-IoU is capable of providing effective regression gradients even when the overlap between the bounding box and the ground truth box is small, thereby accelerating the convergence of high IoU samples[17,18]. On the other hand, MPDIoU (Minimum Point Distance IoU)[19] addresses the issue of optimization difficulties in the loss function when the

predicted and ground truth boxes share the same aspect ratio by introducing the concept of minimum point distance. It calculates the minimum Euclidean distance between the four corner points of the predicted and ground truth boxes. MPDIoU, as detailed in Equations (10)-(13), not only considers the overlapping area but also incorporates the distance between the center points and the width-height deviation, thereby achieving better generalization for bounding boxes of varying shapes and sizes[20,21].

The ABRLoss function [Equation (9)] further improved the accuracy of bounding box predictions and the overall model performance by combining the internal overlap optimization of Inner-IoU and the minimum point distance measure of MPDIoU. This loss function adaptively balances the gradient distribution of different IoU samples by adjusting the size ratio parameter of the auxiliary bounding box. When the ratio is less than 1, the loss function focuses on optimizing high IoU samples, thereby accelerating convergence. Conversely, when the ratio exceeds 1, the loss function expands the effective optimization range for low IoU samples, enhancing the model's ability to locate bounding boxes in the early stages.

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} \times \text{ratio}}{2},\ b_r^{gt} = x_c^{gt} + \frac{w^{gt} \times \text{ratio}}{2} \tag{1}$$

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} \times \text{ratio}}{2},\ b_b^{gt} = y_c^{gt} + \frac{h^{gt} \times \text{ratio}}{2} \tag{2}$$

$$b_l = x_c - \frac{w \times \text{ratio}}{2},\ b_r = x_c + \frac{w \times \text{ratio}}{2} \tag{3}$$

$$b_t = y_c - \frac{h \times \text{ratio}}{2},\ b_b = y_c + \frac{h \times \text{ratio}}{2} \tag{4}$$

$$\text{inter} = \left(\min\left(b_r^{gt}, b_r\right) - \max\left(b_l^{gt}, b_l\right)\right) \times \left(\min\left(b_b^{gt}, b_b\right) - \max\left(b_t^{gt}, b_t\right)\right) \tag{5}$$

$$\text{union} = (w^{gt} \cdot h^{gt}) \cdot (\text{ratio})^2 + (w \cdot h) \cdot (\text{ratio})^2 - \text{inter} \tag{6}$$

$$\text{IoU}^{\text{inner}} = \frac{\text{inter}}{\text{union}} \tag{7}$$

$$L_{\text{Inner-IoU}} = 1 - \text{IoU}^{\text{inner}} \tag{8}$$

$$L_{\text{ABRLoss}} = L_{\text{MPDIoU}} + \text{IoU} - \text{IoU}^{\text{inner}} \tag{9}$$

$$d_1^2 = \left(x_1^B - x_1^A\right)^2 + \left(y_1^B - y_1^A\right)^2 \tag{10}$$

$$d_2^2 = \left(x_2^B - x_2^A\right)^2 + \left(y_2^B - y_2^A\right)^2 \tag{11}$$

$$\text{MPDIOU} = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{12}$$

$$L_{\text{MPDIoU}} = 1 - \text{MPDIoU} = 1 - \text{IoU} + \frac{d_1^2}{h^2 + w^2} + \frac{d_2^2}{h^2 + w^2} \tag{13}$$

4) Simple Attention Module (SimAM) attention mechanism

The attention mechanism plays a critical role in enhancing the performance and robustness of computer vision models, particularly in object detection tasks. Common attention modules, such as CBAM[22], SE[23], CA[24], and ECA[25], each have unique characteristics in emphasizing spatial and channel attention. However, these modules are often associated with additional computational and memory overhead, making them difficult to deploy efficiently in resource-constrained embedded environments.

In this study, tailored to the apple flowers dataset and the computational constraints of the embedded device, an attention module was designed to operate efficiently within a lightweight model framework while enhancing the precision of apple flower detection. Consequently, the SimAM attention mechanism[26] was introduced, which operates without introducing additional parameters to the base network, instead augmenting the model's focus on target features through a streamlined adaptive weighting procedure.

SimAM employs a minimum energy function to assess the significance of individual neurons. This function determines the weights assigned to each neuron, serving as a metric for evaluating the degree of linear discriminability between the target neuron and its counterparts. Specifically, the minimum energy function for the k-th neuron is formulated as:

$$e_k^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t_k - \hat{u})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{14}$$

where, λ is regular term; t_k is k-th neuron of the input feature mAP on a single channel; $\hat{u}$ is mean value of all neurons on a single channel; $\hat{\sigma}^2$ is variance of all neurons on a single channel.

Smaller values of e_k^* indicate lower energy, and neuron K is more differentiated from peripheral neurons and more important for visual processing. Therefore, the importance of each neuron is the inverse of e_k^*. The output feature mAP was calculated as:

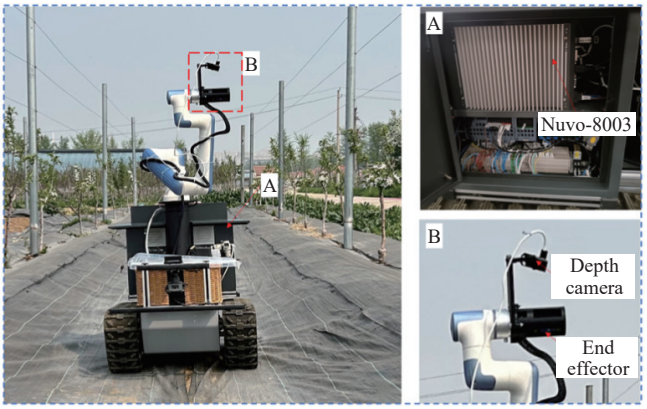
$$\tilde{X} = \text{sigmoid}\left(\frac{1}{E}\right) X \tag{15}$$

To enhance the detection capabilities of the model presented in this paper, the SimAM module was incorporated into the middle two layers of the Neck stage. This integration effectively bolstered the model's responsiveness and discriminatory power for apple flower features, without incurring any additional computational overhead.

2.3 YOLO-FL model deployment

1) Flower thinning robot system construction

To evaluate the practical application of the YOLO-FL, it was successfully deployed on a robot specifically designed for flower thinning in orchards. The hardware configuration of this robot system is shown in Figure 7, which included a tracked chassis, a six-axis collaborative robotic arm, LiDAR, depth cameras, and other high-precision sensors. Additionally, a laser-based flower thinning device was mounted on the end of the robotic arm, enabling precise removal of apple flowers through laser targeting.



Note: This figure illustrates the key components of the flower thinning robot, including the tracked chassis, six-axis robotic arm, LiDAR, end effector, and depth camera, all working in conjunction with the YOLO-FL model for apple flower detection and thinning operations.

Figure 7 Schematic of flower thinning robot system

The robot's control system is based on the Neousys Nuvo-8003 rugged embedded industrial PC from Neousys Technology. This industrial PC runs on the Ubuntu 20.04.6 operating system and a GeForce RTX 2060 GPU, providing sufficient computational power to support real-time image processing and object detection using the YOLO-FL model, ensuring the efficient operation of the vision system in dynamic working environments.

2) Model transfer

In order to enhance the generalization and adaptability of the YOLO-FL model for apple flower detection across diverse orchard environments, this study implemented a transfer learning approach. As shown in Figure 8, the YOLO-FL model pre-trained on the dataset served as the initial weights, which were then transferred to the specific task of apple flower detection within the context of flower thinning robots. Subsequently, the model underwent further refinement through fine-tuning to match the unique characteristics of the new dataset.

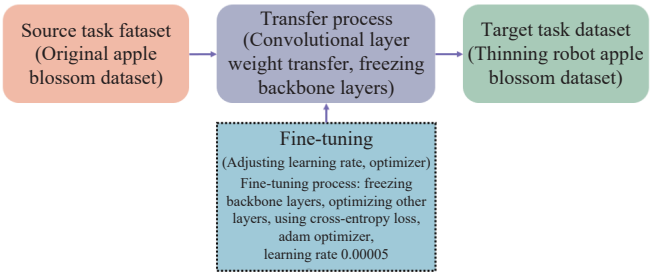


Figure 8 Flowchart of model migration

During the transfer process, the convolutional layer weights learned from the source task were retained and transferred to the target task. To prevent overfitting to the features of the original task while enhancing the learning of apple flower features, the model's Backbone layer was frozen, and the remaining layers were fine-tuned. Fine-tuning was performed using a dataset of 200 apple flower images from the orchard environment of the flower thinning robot.

3 Results and discussion

3.1 Experiment environment

The device used in this experiment for training and testing the model is Lenovo R9000p 2021, RTX 3060 graphics card, and Windows 11 Home Chinese version 2021. Python version was 3.10.10, Pytorch version was 1.13.1, and Cuda version was 11.7[27].

3.2 Evaluative metrics

This paper adopted precision, recall, and mAP as the core evaluation metrics[28] in order to comprehensively evaluate the performance of the trained model in the apple flower detection task. The specific calculation methods and meanings of these metrics are detailed as follows:

$$Precision = \frac{T_P}{T_P + F_P} \times 100\% \quad (16)$$

$$Recall = \frac{T_P}{T_P + F_N} \times 100\% \quad (17)$$

$$\text{mAP} = \frac{1}{K}\sum_{i=1}^{K} \text{AP}(i) \quad (18)$$

where, *Precision* is precision, %; *Recall* is recall rate, %; mAP is mean average precision, %; mAP is a comprehensive evaluation metric that combines precision and recall by calculating the average of the areas under the precision-recall (*P*-*R*) curves. mAP can more comprehensively reflect the model's performance at different recall levels. AP is the area under the *P*-*R* curves; T_P is the number of true positives, the number of apple flowers correctly predicted; T_N is true negative, the number of predicted non-apple flower samples that are indeed non-apple flowers; F_P is false positive, the number of non-apple flowers incorrectly predicted as apple flowers; F_N is false negative, the number of apple flowers incorrectly predicted as non-apple flowers.

3.3 Model component comparison and optimization validation experiments

3.3.1 Architecture improvement validation experiments

In order to verify the effectiveness of the improvement of the detection model in this paper, different modules are used for the comparison test of Bottleneck part and Neck part, as listed in Table 1. In which, the serial numbers are defined as 1: Exclusively employed the C3 module in both the backbone and neck parts. Serial number 2: Integrated EC3DFM in the backbone and C3 in the neck. Serial number 3: Deployed C3 in the backbone and MFEM in the neck. Serial number 4: Applied EC3DFM in the backbone and MFEM in the neck. Serial number 5: Utilized MFEM in the backbone and EC3DFM in the neck.

Among these configurations, the combination of EC3DFM in the Backbone and MFEM in the Neck achieved the peak *p*-value of 74.4% and an mAP value of 77.54% in apple flower detection tasks. Compared to the baseline model, which solely used C3 in both parts, this combination led to improvements in *P*, *R*, and mAP by 2.48%, 1.31%, and 1.61%, respectively. Notably, serial number 3 attained the highest *R*-value of 71.52%, which was slightly higher (by 0.14%) than serial number 4. After comprehensive evaluation of all metrics, this paper adopts the combination of EC3DFM in the Backbone and MFEM in the Neck to construct the apple flower detection model.

Table 1 Verification results of different combinations of EC3DFM, MFEM modules

Serial number	Model formulation	C3	EC3DFM	MFEM	*P*/%	*R*/%	*mAP*/%
1	Backbone	√	×	×	71.92	70.07	75.93
	Neck	√	×	×			
2	Backbone	×	√	×	71.44	73.91	76.92
	Neck	√	×	×			
3	Backbone	√	×	×	72.83	71.52	76.55
	Neck	×	×	√			
4	Backbone	×	√	×	74.40	71.38	77.54
	Neck	×	×	√			
5	Backbone	×	×	√	72.54	69.57	75.96
	Neck	×	√	×			

3.3.2 ABRLoss and SimAM mechanism validation experiments

In this paper, ABRLoss was incorporated into the loss function, while the SimAM mechanism was employed in the attention mechanism. To assess their individual and combined effects, experiments were conducted utilizing CIoU, MPDIoU, and Inner-CIoU, and performance with and without the SimAM mechanism was evaluated. The results of these experiments are presented in Table 2.

Examining Table 2,when using MPDIoU alone as the loss function, the model showed some improvement in precision and recall compared to CIoU, but a slight decrease in mAP/%. When using Inner-CIoU at the same time there was an improvement in recall and mAP/%, but a slight decrease in precision. When using

ABRLoss loss function there was a slight decrease of 0.05 percentage points in precision compared to CIoU and an increase of 1.96% in recall and 0.62% in mAP, indicating that the performance of different loss function realizations varies in this dataset.

Table 2 ABRLoss letter validation results

Model	CIoU	MPDIoU	Inner-CIoU	ABRLoss	SimAM	*P*/%	*R*/%	*mAP*/%
YOLO-FL	√	×	×	×	×	74.40	71.38	77.54
YOLO-FL	×	√	×	×	×	74.81	71.77	77.39
YOLO-FL	×	×	√	×	×	72.68	73.35	78.23
YOLO-FL	×	×	×	√	×	74.35	73.34	78.16
YOLO-FL	×	×	×	√	√	74.63	73.82	79.97

Upon incorporating the SimAM attention mechanism in conjunction with ABRLoss, the model's precision, recall, and mAP were augmented by 0.28%, 0.48%, and 1.81%, respectively. This demonstrated that the SimAM attention mechanism enhanced the model's overall performance by amplifying its focus on critical features within the image. The synergistic impact of the ABRLoss function and SimAM attention mechanism within the YOLO-FL framework confirms their effectiveness as potent strategies for elevating the performance of the target detection model and refining its detection accuracy and stability.

3.4 ABRLoss optimization ratio analysis experiments

In the model presented in this paper, the ABRLoss was utilized, and the optimization ratio within this loss function exerted a significant influence on the model's performance. In order to deeply investigate the effect of different optimization ratios on the model performance, this section was tested by adjusting only the ratio parameter in the ABRLoss function with other settings of the YOLO-FL unchanged, and the results are shown in Figure 9.

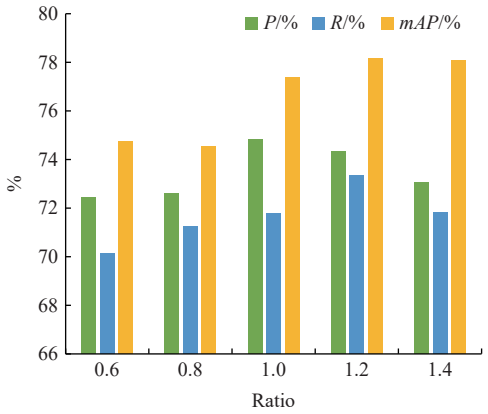


Figure 9 Verification results of different ratios of ABRLoss functions

Figure 9 shows the test results for different ratios: 0.6, 0.8, 1.0, 1.2, and 1.4. The model achieved the highest performance when the ratio parameter was set to 1.2, with *P* at 74.35%, *R* at 73.34%, and mAP at 78.16%, outperforming all other subgroups. The model's performance decreased as the ratio parameter deviated from 1.2. The lowest performance was observed when the ratio was 0.6, with *P* at 72.45% and *R* at 70.14%. Similarly, the mAP value was the lowest among all models when the ratio was 0.8, at 74.53%. The experimental results indicate that selecting the appropriate ratio parameter is crucial for optimizing the ABRLoss function, as it has a direct bearing on the model's detection capabilities and overall efficacy. For the lower IoU samples of this paper's dataset, a larger auxiliary bounding box was found to be appropriate, while the detection performance of the YOLO-FL was optimal when the ratio parameter was 1.2. This further validated the reliability of the ABRLoss function proposed in this paper.

3.5 Discussion of results for different attention mechanism functions

This section examined the performance of four distinct attention mechanisms—CBAM, SE, EMA[29], and SimAM—within the YOLOv5 model for apple flower detection, specifically under identical experimental conditions. The four attention mechanisms were added to the same middle two layers of outputs of the Head and tested under the same experimental environment. The results are listed in Table 3. Table 3 shows that the benchmark YOLOv5 model achieved an accuracy of 71.92%, a recall of 70.07%, and a mean average precision of 75.93%. The introduction of CBAM improved the accuracy, but the recall and mean averages of precision decreased slightly. The introduction of SE improved the recall, but the accuracy decreased. The introduction of EMA resulted in a substantial decrease in both the recall and accuracy by 1.4%. This study demonstrated that the introduction of the SimAM attention mechanism had a positive impact on the model's accuracy, recall, and mean average precision. Specifically, the model's accuracy increased to 73.09%, recall increased to 70.9%, and mean average precision increased to 76.12%. It is crucial to observe that the impact of various attention mechanisms on the model varied significantly. Notably, one particular attention mechanism exhibited a more balanced performance across the evaluative indicators compared to the other three. Furthermore, the incorporation of this mechanism did not entail an increase in the model's size, thereby highlighting its practical viability.

Table 3 Validation results of different attention mechanisms

Model	CBAM	SE	EMA	SimAM	*P*/%	*R*/%	*mAP*/%	Model size/KB
YOLOv5	×	×	×	×	71.92	70.07	75.93	5163
YOLOv5	√	×	×	×	72.4	69.28	74.68	5206
YOLOv5	×	√	×	×	69.68	71.05	74.75	5166
YOLOv5	×	×	√	×	73.32	70.05	76.10	5174
YOLOv5	×	×	×	√	73.09	70.9	76.12	5164

3.6 Field test experiment of a flower thinning robot

After deploying the YOLO-FL apple flower detection model to the multifunctional orchard robot, in order to test its detection performance, it was tested in the dwarf and densely planted apple test garden of Shandong Agricultural University. Initially, Figure 10 displays the video detection segment of the imagery, demonstrating the model's capacity to efficiently identify apple flowers within the video stream. Additionally, to assess the model's practical performance more rigorously, 60 randomly selected groups of apple images were subjected to detection, achieving a frame rate of 40.7 FPS. The images were divided into two major categories according to the far target and the near target, and divided into non-congested, moderately congested, and very congested according to the number of apple flowers. The number of apple blossoms, the number of missed detections, and the number of false detections in the images were counted, and the results are shown in Table 4. Some of the detected images and the corresponding heat mAPs generated by GradCAM are shown in Figure 11.

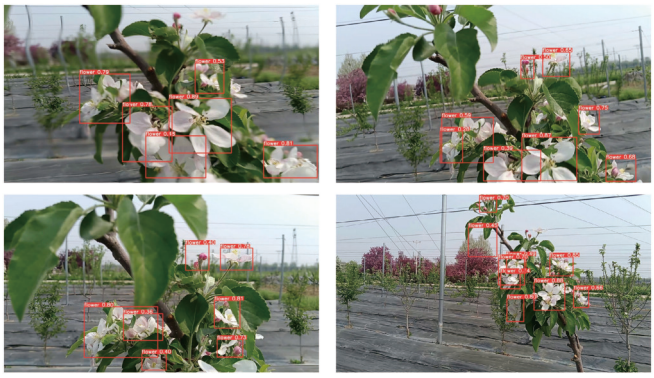


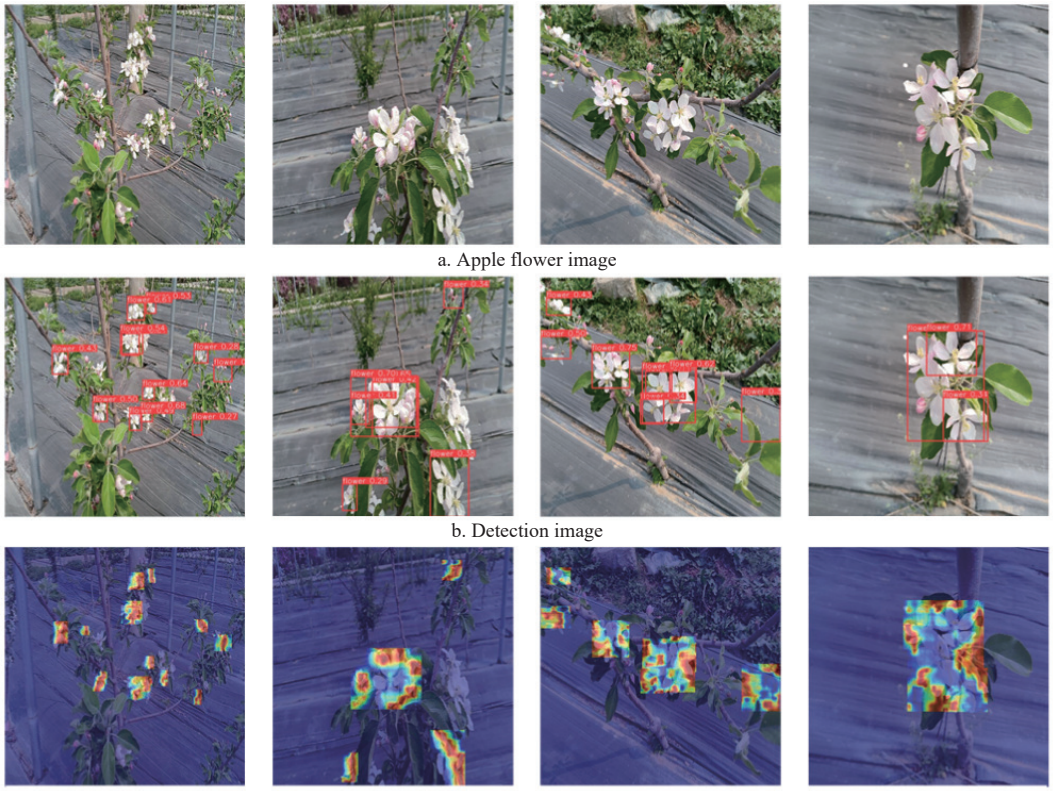
Figure 10 On-site video detection

The tabled data in Table 4 indicate that the model demonstrated superior overall detection performance, boasting an average un-detection rate of 7.26% and an average false detection rate of 6.89%. The detection performance under different distances and target densities was different. The overall detection effect of near targets was better than that of far targets, and the un-detection rate and false detection rate were relatively low in the non-clustered case, with the lowest un-detection rate at 3.57% and the lowest false detection rate at 1.85%. The un-detection rate and false detection rate were relatively high in the highly congested case. The leakage detection rate was 16.25% and the false detection rate was 14.69% in the case of a highly congested distant target. This shows that the model is more prone to un-detection and false detection under high target density in the distant situation.

Upon examination of the detection images and corresponding heat mAPs presented in Figure 11, it is clear that the model possessed the capability to detect apple flowers with a speed that aligns with the operational demands of this robotic vision system in practical field conditions. However, it is still evident that under intense illumination and a varying number of flowers, the precision of detection diminished, while the miss rate escalated. A thorough assessment of diverse detection performance metrics further substantiates that the proposed model in this paper fulfilled the target recognition needs of multifunctional robots in orchard environments.

Table 4 Experimental results of the YOLO-FL model deployed into a multifunctional robotic vision system

Classification		Total number	Detected/ numbers	Undetected/ numbers	False detection/ numbers	Un-detection rate/%	False detection rate/%
Far target	Non-congested	84	76	3	5	3.57	5.95
	Moderately congested	177	157	11	9	6.21	5.08
	Very congested	320	221	52	47	16.25	14.69
Near target	Non-congested	54	51	2	1	3.70	1.85
	Moderately congested	153	140	8	5	5.23	3.27
	Very congested	267	216	23	28	8.61	10.49



a. Apple flower image

b. Detection image

c. GradCAM image

Figure 11 Field image detection

4 Discussion

4.1 Comprehensive performance comparison analysis of YOLO-FL apple flower detection models

This subsection undertakes a comparative analysis of the efficacy of several cutting-edge target detection models for the apple flower recognition task, encompassing YOLOv5[30], YOLOv6[31], YOLOv7[32], YOLOv8[33], and the refined model introduced in this study. The outcomes presented in Table 5 reveal that, in the realm of precision, the refined model introduced herein demonstrates a notable superiority compared to its contemporary counterparts. The accuracy rate of this paper's model reached

74.63%, which was improved by 2.71% compared with the 71.92% of the benchmark model, YOLOv5. Additionally, compared with YOLOv6, YOLOv7, and YOLOv8, this paper's model had different degrees of improvement. Secondly, in terms of recall rate, this paper's model also showed excellent performance. The recall rate reached 73.82%, which was improved by 3.7% compared to 70.07% in YOLOv5. In terms of mAP, this paper's model also achieved the optimal result of 79.97%. In contrast to other models, the model introduced in this study demonstrated a superior balance in overall performance, consistently delivering high detection accuracy across various scenarios. Additionally, in terms of model size, this study's model exhibited a compact design of 4693 KB, marking a substantial decrease of approximately 60.8% and 44.8% in comparison to YOLOv7's 11977 KB and YOLOv6's 8502 KB, respectively. Consequently, through a rigorous evaluation of diverse state-of-the-art models on the apple flower recognition task, the strengths of the proposed model in accuracy, recall, mAP, and model compactness were further highlighted.

Table 5 Test results of different models

Model	*P*/%	*R*/%	*mAP*/%	Model size/KB
YOLOv5	71.92	70.07	75.93	5163
YOLOv6	68.47	70.05	76.27	8502
YOLOv7	72.14	70.46	76.07	11 977
YOLOv8	71.32	70.84	76.11	6115
YOLO-FL	74.63	73.82	79.97	4693

4.2 Practical applicability analysis of the YOLO-FL apple flower detection model

To assess the efficacy of the YOLO-FL model in practical orchard applications, it was integrated into a laser-based flower thinning robot and subjected to field trials. The YOLO-FL model's exceptional detection performance in complex orchard environments was evident from the results presented in Table 4, which were corroborated by the visual representation in Figure 11. In terms of detection speed, YOLO-FL achieved a high rate of 40.7 FPS, meeting real-time requirements. Moreover, in scenarios involving nearby targets and non-clustered conditions, the model showed lower un-detection and false detection rates, further confirming its reliability in practical applications.

However, it was also observed that under conditions of long distances and high flower density, the detection accuracy of YOLO-FL declined, with both un-detection rate and false detection rate increasing. This was attributed to the smaller size of flowers in images captured at greater distances compared to those in the training dataset, as well as the greater disparity between the two. Additionally, high flower density led to increased occlusion and overlap, further complicating detection.

Overall, the YOLO-FL model proposed in this study was successfully deployed on the laser-based flower thinning robot, enabling effective apple flower detection. This validated the model's detection performance and deployment capabilities on embedded devices, providing reliable visual detection support for the flower thinning robot.

This finding implies that future research endeavors should prioritize refining the model to achieve enhanced adaptability to increasingly complex environmental conditions. Potential avenues include the incorporation of sophisticated image preprocessing techniques to bolster image contrast and clarity, as well as the utilization of innovative data augmentation methodologies to bolster the model's resilience and enhance its ability to generalize effectively.

4.3 Limitations analysis of the YOLO-FL apple flower detection model

The YOLO-FL developed in this study was shown to effectively detect apple flowers, achieving high detection accuracy while improving detection speed. However, in complex orchard environments, the model encountered challenges with un-detection and false detection rates under certain conditions, such as long distances, high flower density, and varying lighting conditions. As shown in Table 4, under scenarios with long distances and extreme flower clustering, the un-detection rate reached 16.25%, and the false detection rate reached 14.69%. These limitations were primarily caused by the adverse effects of complex environmental factors, which reduced detection performance.

In summary, although the overall performance of the model was significantly improved, enhancing its robustness and generalization ability for apple flower detection under complex conditions remains a priority for future optimization efforts.

5 Conclusions

This study introduced an enhanced apple flower detection model, YOLO-FL, which built upon the YOLO object detection framework. The model exhibited notable advancements in detection accuracy without compromising on speed, thereby offering a robust and dependable approach for apple flower recognition tasks, characterized by both efficiency and reliability.

In the model development process, a comprehensive data collection and preprocessing procedure was conducted on the apple flower dataset, establishing a solid foundation for subsequent training and validation. The Backbone was optimized by incorporating the EC3DFM architecture, which enhanced the model's feature extraction capabilities. The MFEM structure was introduced in the Neck section, effectively improving feature fusion. Furthermore, the ABRLoss function was employed to refine the predictions of bounding boxes, thereby increasing localization accuracy. To further boost detection performance, the SimAM attention mechanism was integrated into the two middle detection heads in the Neck section.

On the test set, the model achieved a precision (*P*) of 74.63%, recall (*R*) of 73.82%, and mean average precision (*mAP*) of 79.97%, with a compact model size of 4693 KB. These results represent improvements of 2.71%, 3.75%, and 4.04%, respectively, compared to the YOLOv5 model, while the model size was reduced by 470 KB, demonstrating notable advantages in efficiency and storage.

The YOLO-FL was subsequently deployed on a multifunctional orchard robot. The system delivered an image with 40.7 FPS, with an average missed detection rate of 7.26% and a false detection rate of 6.89%, meeting the operational requirements of the robot's visual detection system. These findings indicate that the model demonstrates high performance and efficiency in apple flower recognition tasks.

This work not only augments the precision and efficacy of apple flower recognition but also contributes invaluable technical insights to applications within orchard management, agricultural production, and plant research. Prospective endeavors will concentrate on refining the model's performance and seamlessly integrating it with a laser-based flower thinning robot to facilitate intelligent thinning operations.

Acknowledgements

This study was supported the Fruit Industry Modern Agricultural Technology System in Shandong Province (SDAIT-06-

12) and Innovation Team Fund for Key R&D Program of Shandong Province, China (2024TZXD038, 2024TZXD045).

[References]

[1] Bi S H, Li X, Shen T, Xu Y, Ma L Y. Apple classification based on evidence theory and multiple models. Transactions of the CSAE, 2022; 38(13): 141–149. (in Chinese)

[2] Chen Z, Yu L, Liu W, Zhang J, Wang N, Chen X S. Research progress of fruit color development in apple (Malus domestica Borkh). Plant Physiology and Biochemistry, 2021; 162: 267–279.

[3] Hussain M, He L, Schupp J, Lyons D, Heinemann P. Green fruit segmentation and orientation estimation for robotic green fruit thinning of apples. Computers and Electronics in Agriculture, 2023; 207: 107734.

[4] Wang D, He D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. Biosystems Engineering, 2021; 210(6): 271–281.

[5] Shang Y Y, Xu X S, Jiao Y T, Wang Z, Hua Z X, Song H B. Using lightweight deep learning algorithm for real-time detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 2023; 207: 107765.

[6] Chen G F, Chen Z Y, Wang Y L, Fan G Q, Li H Q. Research on detection method of apple flower based on data-enhanced deep learning. Journal of Chinese Agricultural Mechanization, 2022; 43(5): 148–155. (in Chinese)

[7] Datt R M, Kukreja V. Phenological stage recognition model for apple crops using transfer learning. 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022; pp.1537–1542. DOI: 10.1109/ICACITE53722.2022.9823711.

[8] Zhou X Z, Sun G X, Xu N M, Zhang X L, Cai J Q, Yuan Y P, et al. A method of modern standardized apple orchard flowering monitoring based on S-YOLO. Agriculture, 2023; 13(2): 380.

[9] Mu X Y, He L, Heinemann P, Schupp J, Karkee M. Mask R-CNN based apple flower detection and king flower identification for precision pollination. Smart Agricultural Technology, 2023; 4: 100151.

[10] Yang R J, Li W F, Shang X N, Zhu D P, Man X Y. KPE-YOLOv5: An improved small target detection algorithm based on YOLOv5. Electronics, 2023; 12(4): 817.

[11] Li Y D, Xue J X, Zhang M Y, Yin J Y, Liu Y, Qiao X D, et al. YOLOv5-ASFF: A multistage strawberry detection algorithm based on improved YOLOv5. Agronomy, 2023; 13(7): 1901.

[12] Shi H K, Xiao W F, Zhu S P, Li L B, Zhang J F. CA-YOLOv5: Detection model for healthy and diseased silkworms in mixed conditions based on improved YOLOv5. Int J Agric & Biol Eng, 2023; 16(6): 236–245.

[13] Ding X H, Zhang Y Y, Ge Y X, Zhao S J, Song L, Yue X Y, et al. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA, 2024. DOI: 10.1109/CVPR52733.2024.00527.

[14] Lin X, Yue J T, Chan K C K, Qi L, Ren C, Pan J S, et al. Multi-task image restoration guided by robust DINO features. arXiv: 2312.01677, 2023; DOI: 10.48550/arXiv.2312.01677.

[15] Liu X, Wang Y, Yu D, Yuan Z. YOLOv8-FDD: A real-time vehicle detection method based on improved YOLOv8. IEEE Access, 2024; 12: 136280–136296. DOI: 10.1109/ACCESS.2024.3453298.

[16] Zhang H, Xu C, Zhang S J. Inner-IoU: More effective intersection over union loss with auxiliary bounding box. arXiv: 2311.02877, 2023. DOI: 10.48550/arXiv.2311.02877.

[17] Zhang P R, Zhang G X, Yang K H. APNet: Accurate positioning deformable convolution for UAV image object detection. IEEE Latin America Transactions, 2024; 22: 304–311.

[18] Hu Z C, Wang Y, Wu J P, Xiong W L, Li B L. Improved lightweight rebar detection network based on YOLOv8s algorithm. Advances in Computer, Signals and Systems, 2023; 7: 107–117.

[19] Ma S L, Xu Y. MPDIoU: A loss for efficient and accurate bounding box regression. arXiv: 2307.07662, 2023. DOI: 10.48550/arXiv.2307.07662.

[20] He J, Zhang S, Yang C, Wang H, Gao J, Huang W, et al. Pest recognition in microstates state: an improvement of YOLOv7 based on Spatial and Channel Reconstruction Convolution for feature redundancy and vision transformer with Bi-Level Routing Attention. Front. Plant Sci, 2024; 15: 1327237.

[21] Chen B J, Zhang W H, Wu W B, Li Y R, Chen Z L, Li C L ID-YOLOv7: an efficient method for insulator defect detection in power distribution network. Front. Neurorobot, 2024; 17: 1331427. DOI: 10.3389/fnbot.2023.1331427.

[22] Woo S, Park J, Lee J, Kweon I. CBAM: Convolutional block attention module. ArXiv: 1807.06521v2, 2018. DOI: 10.48550/arXiv/abs/1807.06521.

[23] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2018. DOI: 10.1109/CVPR.2018.00745.

[24] Hou Q B, Zhou D Q, Feng J S. Coordinate attention for efficient mobile network design. IEEE/CF Conference on Computer Vision and Patern Recognition, Nashville, USA, 2021. DOI: 10.1109/CVPR46437.2021.01350.

[25] 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. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA, 2020; pp.11531–11539. DOI: 10.1109/CVPR42600.2020.01155.

[26] Yang L X, Zhang R Y, Li L D, Xie X H. SimAM: A simple, parameter-free attention module for convolutional neural networks. International Conference on Machine Learning, 2021. Available: https://api.semanticscholar.org/CorpusID:235825945. Accessed on [2024-12-20].

[27] Gao A, Ren H, Song Y P, Ren L L, Zhang Y, Han X. Construction and verification of machine vision algorithm model based on apple leaf disease images. Front Plant Sci, 2023; 14: 1246065.

[28] Jiang Y Y, Chen X Y, Li G M, Wang F, Ge H Y. Graph neural network and its research progress in field of image processing. Computer Engineering and Applications, 2023; 59(7): 15–30.

[29] Ouyang D L, He S, Zhang G Z, Luo M Z, Guo H Y, Zhan J, et al. Efficient multi-scale attention module with cross-spatial learning. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023; DOI: 10.1109/ICASSP49357.2023.10096516.

[30] Roy A M, Bhaduri J. DenseSPH-YOLOv5: An automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism. Adv. Eng. Informatics, 2023; 56: 102007.

[31] Li C Y, Li L L, Jiang H L, Weng K H, Geng Y F, Li L, et al. YOLOv6: A single-stage object detection framework for industrial applications. ArXiv: 2209.02976. DOI: 10.48550/arXiv.2209.02976.

[32] Wang C Y, Alexey Bochkovskiy, Liao H Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp.7464–7475. DOI: 10.1109/CVPR52729.2023.00721.

[33] Hussain M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines, 2023; 11(7): 677.