Detection method for *Lycium barbarum* L. ripe fruit regions used in the precision vibration harvesting

Naishuo Wei[1], Shiwei Wen[1], Guangrui Hu[2], Yunlei Fan[1], Yingkuan Wang[3,4*], Jun Chen[1*]
(1. *College of Mechanical and Electronic Engineering, Northwest A & F University, Yangling, Shaanxi 712100, China*;
2. *School of Design, Xi'an Technological University, Xi'an 710021, China*;
3. *Academy of Agricultural Planning and Engineering, Ministry of Agriculture and Rural Affairs, Beijing 100125, China*;
4. *Chinese Society of Agricultural Engineering, Beijing 100125, China*)

Abstract: Current *Lycium barbarum* L. vibration harvesting equipment exhibits low levels of intelligence and precision, often resulting in a trade-off between efficiency and fruit damage. This study proposed a ripe fruit region detection model, YOLO-RFR, specifically for precision vibration harvesting of *L. barbarum*. First, the ADown downsampling module was introduced to replace part of the conventional convolution layers. Then, the C3k2-AP module, inspired by the asymmetric padding strategy, was designed to replace the C3k2 module. Additionally, the GCHead detection head was constructed using group convolution. Finally, the EMA-Slide Loss function was developed to optimize the classification performance by combining the slide weighting function with Exponential Moving Average (EMA). The experimental results showed that the model achieved precision, recall, and mAP of 93.7%, 92.0%, and 97.0%, respectively, representing improvements of 4.0%, 4.4%, and 2.6% over the baseline. The parameter, floating-point operations (FLOPs), and model size were 1.7 M, 4.1 G, and 3.8 MB, respectively, corresponding to decreases of 34.6%, 34.9%, and 30.9% compared with the baseline. To further validate its practical feasibility, the improved model was deployed on an NVIDIA Jetson AGX Xavier embedded device, achieving an inference speed of 163 fps with TensorRT acceleration. In conclusion, the YOLO-RFR model demonstrated excellent performance in detection accuracy, model lightweighting, and deployment on embedded devices, providing strong technical support for the precision vibration harvesting of *L. barbarum*.

Keywords: *L. barbarum*, precision vibration harvesting, target detection, embedded device deployment
DOI: 10.25165/j.ijabe.20261903.10021

Citation: Wei N S, Wen S W, Hu G R, Fan Y L, Wang Y K, Chen J. Detection method for *Lycium barbarum* L. ripe fruit regions used in the precision vibration harvesting. Int J Agric & Biol Eng, 2026; 19(3): 198–211.

1 Introduction

Lycium barbarum L. (*L. barbarum*) is an important economic crop in the northwest region of China, with Ningxia's *L. barbarum* being the most renowned[1]. Its fruits can be used to produce a variety of agricultural by-products with rich nutritional value, such as dried fruit, fruit juice, and fruit wine[2,3]. As an indeterminate inflorescence, continuous flowering and fruiting plant, it gradually ripens from July to October each year, with flowers and fruits coexisting on the same branch, making it necessary to selectively harvest the ripe fruits. The ripe fruits of *L. barbarum* are red, oval-shaped berries with thin skin and high moisture content. Untimely harvesting can lead to natural fruit drop, decay, damage, and browning or adhesion during drying[4,5], resulting in a distinct seasonal harvesting characteristic. Extensive research has been carried out by numerous scholars focusing on the mechanization of *L. barbarum* harvesting. Among these, efficient vibration harvesting has been widely applied. Depending on the operational scale, current harvesting equipment is generally categorized into two types: small portable and large cross-row self-propelled harvesters[6]. Zhang et al.[7] developed a handheld shaking harvester that induces oscillations in fruit-bearing branches to separate the fruits from the branches. The device achieved a green fruit mispicking rate of 5.72%, a fruit damage rate of 2.54%, and a harvesting efficiency 5.5 times higher than manual picking. Chen et al.[8] explored various reciprocating vibration harvesting methods for *L. barbarum* and applied excitation above the ripe fruit regions, achieving a 95.14% ripe fruit picking rate and a 2.98% fruit damage rate under the optimal vibration parameter combination. Mei et al.[9] designed a cross-row *L. barbarum* harvester by combining hedgerow planting agronomy. By selecting different array layout parameters according to the fruiting stages of summer fruits, a ripe fruit harvesting rate of 88.95%, a fruit damage rate of 3.64%, and a harvesting efficiency 26.91 times higher than manual picking were obtained. Small portable harvesters offer better harvesting performance but rely on manual operation, resulting in low operational efficiency. In contrast, large cross-row harvesters exhibit high harvesting capacity; however, the extensive vibration harvesting approach tends to cause significant mechanical damage to fresh *L. barbarum* fruits, often leading to a trade-off between harvesting efficiency and fruit damage. The vibration harvesting of *L. barbarum* primarily relies on applying excitation to fruit-bearing branches, generating inertia differences between ripe and unripe fruits. By considering

Received date: 2025-07-14 **Accepted date:** 2026-03-26
Biographies: Naishuo Wei, PhD candidate, research interest: image processing, Email: weinaishuo@nwafu.edu.cn; **Shiwei Wen**, MS candidate, research interest: image processing, Email: wsw142301@nwafu.edu.cn; **Guangrui Hu**, PhD, Lecturer, research interest: precision agriculture, Email: hugrui@xatu.edu.cn; **Yunlei Fan**, PhD candidate, research interest: agricultural engineering, Email: yunleifan@nwafu.edu.cn.
***Corresponding author: Yingkuan Wang**, PhD, Research Fellow, research interest: agricultural engineering, Email: wangyingkuan@163.com; **Jun Chen**, PhD, Professor, research interest: intelligent agricultural equipment. College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China. Tel: +86-13572191773, Email: chenjun_jdxy@nwsuaf.edu.cn.

the vibration transmission characteristics of the branches, applying excitation above the ripe fruit regions not only enables selective harvesting of ripe fruits while preserving unripe ones but also effectively reduces fruit damage[8]. To effectively balance vibration harvesting performance, operational efficiency, and fruit damage, the precise localization of vibration picking points above the ripe fruit regions has become a critical step for achieving precision vibration harvesting of *L. barbarum*. Precision vibration harvesting aims to intelligently perform "picking ripe while preserving unripe" by accurately applying excitation, thereby reducing fruit damage and improving efficiency. Therefore, the fast and accurate identification of *L. barbarum* ripe fruit regions using object detection methods serves as the key technology to assist vibration harvesting equipment in enhancing fruit harvesting completeness and minimizing damage. This is of great significance for promoting the intelligent and precise development of *L. barbarum* vibration harvesting equipment.

In recent years, with the rapid advancement of artificial intelligence technologies, deep learning-based object detection methods have been widely used in typical agricultural scenarios such as diverse fruit morphologies, complex natural backgrounds, and varying lighting conditions[10-13]. Common algorithms include R-CNN[14], Fast R-CNN[15], and Faster R-CNN[16], SSD[17], RetinaNet[18], CenterNet[19], and YOLO[20]. Among them, YOLO and its improved algorithms have been widely applied in intelligent agriculture scenarios such as automated harvesting, pest and disease detection, and yield prediction, due to their advantages of high detection accuracy and fast inference speed[21]. Gao et al.[22] realized apple detection for four categories, including no occlusion, leaf occlusion, branch occlusion, and fruit occlusion, based on Faster R-CNN. The experimental results showed that the average detection accuracy can reach 87.9%. Parvathi and Selvi[23] effectively distinguished between immature and mature coconuts in complex backgrounds by integrating the ResNet-50 feature extraction network with Faster R-CNN. Liu et al.[24] proposed an improved YOLOv8 lightweight apple detection model. The results showed that the improved model reduced the model parameters and FLOPs to 0.66 M and 2.29 G, respectively, while achieving an mAP@0.50:0.95 of 84.12% on a custom dataset. Qin et al.[25] employed an improved YOLOv8 model for individual *L. barbarum* ripe fruit detection, achieving a mean detection accuracy of 90.2%. Wang et al.[26] proposed an improved YOLOv8n-Pose model for the detection of individual *L. barbarum* fruits and the localization of picking points. The improved model contained only 3.04 M parameters, achieving 92.7% mAP for fruit detection and 86.4% mAP for picking point localization. Although the above studies have achieved significant progress in detection accuracy and model lightweighting, they primarily focus on single-fruit detection tasks. Their applicability remains limited in scenarios involving clustered fruits or intelligent harvesting operations requiring batch picking.

Many researchers have conducted in-depth studies on target detection algorithms for string-type and cluster-type fruits, specifically treating entire fruit clusters as unified detection targets. Li et al.[27] achieved a litchi detection accuracy of 87.43% by improving YOLOv3-tiny; the detection results were further partitioned into different harvestable regions in three-dimensional space to meet the requirements of batch harvesting using the *K*-means clustering algorithm. Chen et al.[28] proposed an improved multi-task deep convolutional neural network model based on YOLOv7, which integrated the detection of tomato clusters, individual tomatoes within the clusters, and the overall ripeness assessment of the clusters into a single framework. The improved model achieved an average precision of 86.6% across multiple tasks, with an average inference time of 4.9 ms, effectively balancing detection accuracy and real-time performance. Chen et al.[29] proposed an improved YOLOv5 model for real-time detection of ripe table grapes in complex orchard environments. The model was deployed on embedded devices and achieved an mAP@0.5 of 91.2%, with a 46% increase in detection speed and a per-image detection time of just 45 ms.

In summary, most existing fruit detection algorithms focus on medium-to-large individual fruits, such as apples, tomatoes, and grapes, or on fruit targets with obvious cluster features. These fruits typically have clear contours and distinct color differences, which are conducive to feature extraction and recognition using deep learning. However, for *L. barbarum*, which grows on tangled branches and is prone to occlusion, existing models still struggle to meet the practical requirements for precise detection of *L. barbarum* ripe fruit regions for precision vibration harvesting. This study proposed a detection model for *L. barbarum* ripe fruit regions and deployed it on the NVIDIA Jetson AGX Xavier embedded device. The main contributions may be as follows:

1) Images of *L. barbarum* under different lighting conditions in natural environments were collected to establish an image dataset for the "Ningqi No. 7" cultivar. Both offline and online data augmentation methods were employed to further enhance the dataset diversity.

2) YOLO-RFR model was proposed by integrating ADown, C3k2-AP, GCHead detection head, and EMA-Slide Loss function into YOLOv11n.

3) The YOLO-RFR model was deployed on an NVIDIA Jetson AGX Xavier embedded device to evaluate its detection performance under the limited computing power of embedded devices, providing technical support for precision vibration harvesting of *L. barbarum*.

2 Materials and methods

2.1 Image data acquisition

The experiment was carried out at an *L. barbarum* plantation base located in Guyuan City (106°15′25″E, 36°00′36″N), Ningxia Hui Autonomous Region, China (Figure 1a). Images of *L. barbarum* during the ripening period were collected from July to August in both 2023 and 2024, focusing on the "Ningqi No. 7" cultivar. In this base, *L. barbarum* trees are planted in a double hedge planting mode (Figure 1b), with a row spacing of 3 m, a plant spacing of 1 m, and an age of 4-5 years. The branches are naturally drooping through artificial pruning to reduce the intertwining of branches. After pruning, the average plant height is 1.8 m, and the crown diameter is 1.2 m. The image acquisition devices include smartphones and depth cameras, with image resolutions of 3024×3024 pixels and 1920×1080 pixels, respectively, and are cropped to a 1:1 ratio. In order to simulate the actual working perspective of the visual sensor in the cross-row *L. barbarum* harvester, the images were taken facing the upright *L. barbarum* trees at a distance of approximately 1.0-1.5 m. A total of 1427 images under both front-light and back-light conditions at different time periods (from 8:00 a.m. to 11:00 a.m. and from 2:00 p.m. to 5:00 p.m.) were collected and saved in PNG format, as shown in Figure 1c.

2.2 Dataset construction

2.2.1 Dataset annotation

Before model training, it is essential to annotate the dataset images, as high-quality annotations play a crucial role in improving model performance. For fruit detection in precision vibration harvesting of *L. barbarum*, mechanical damage caused by direct contact with fruits can be effectively reduced by directly applying excitation above the ripe fruits on the fruit-bearing branches during vibration harvesting. The precise localization of vibration picking points thus relies on the fast and precise detection of ripe fruit regions. Therefore, this study targets the ripe fruit regions on fruit-bearing branches for annotation.

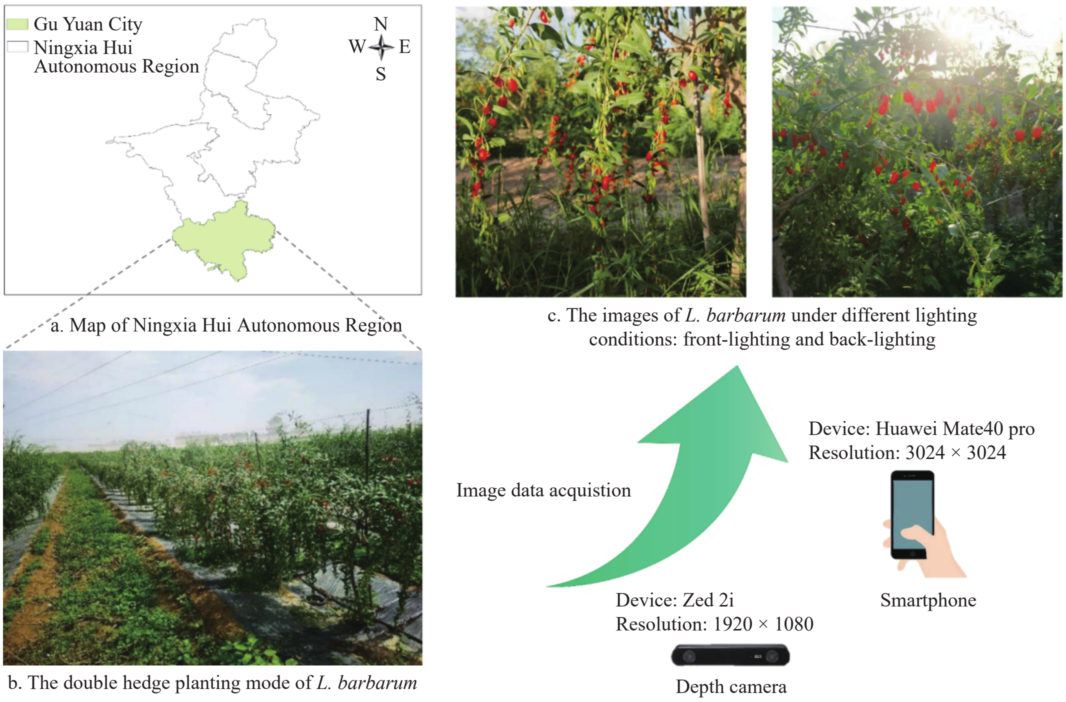


Figure 1 Image data acquisition area of this study

YOLOv11 can adapt to images of different sizes and convert them into the image resolution needed by the network, but this leads to longer training time and reduced training efficiency. To address this issue, the dataset images of *L. barbarum* were uniformly resized to 640×640 pixels before training the model to ensure compatibility with the YOLOv11 network and improve the training speed and efficiency. The Labelimg software was used to annotate the ripe fruit regions with rectangular bounding boxes, labeled as “goji”, as shown in Figure 2. The corresponding label file was generated to record the position and class information of the target boxes. After the annotation was completed, the label file was converted into the TXT format required by YOLOv11.

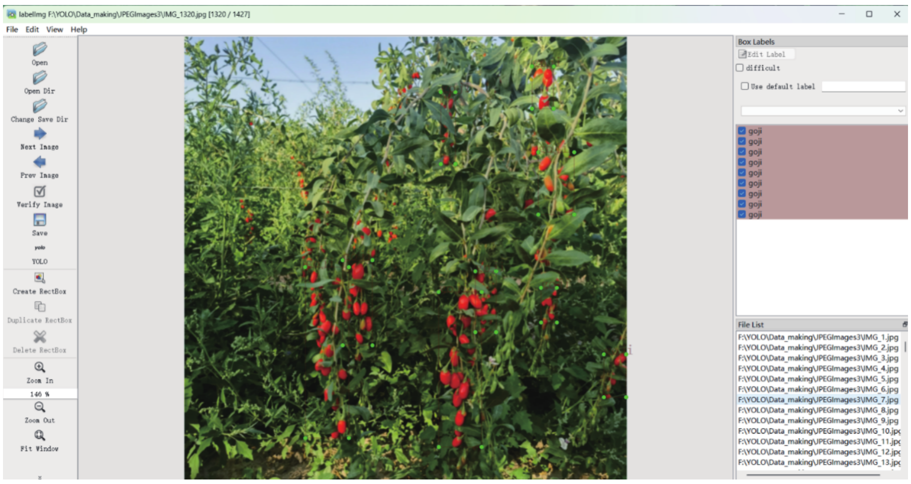
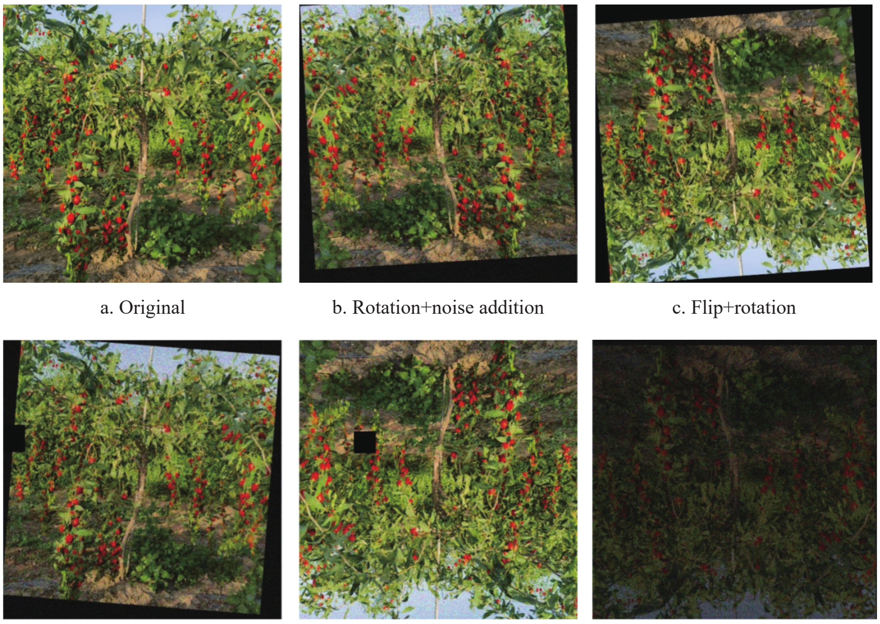


Figure 2 Examples of dataset annotation

2.2.2 Data augmentation and division

To enhance the robustness of the model and reduce the risk of overfitting, data augmentation techniques such as random flip, rotation, translation, brightness adjustment, noise addition, and cutout were applied to the original dataset, as presented in Figure 3. Each image underwent five augmentation operations, with at least one augmentation method applied per operation, expanding the dataset to 8562 images. Subsequently, the enhanced dataset and the corresponding label file were divided into a training set, a validation set, and a test set at a ratio of 7:2:1. The training set comprised 6850 images, the validation set comprised 1712 images, and the test set comprised 857 images. The division results are listed in Table 1.



a. Original b. Rotation+noise addition c. Flip+rotation

d. Rotation+noise addition+cutout e. Flip+noise addition+cutout f. Flip+translation+brightness adjustment+noise addition

Figure 3 Data augmentation processing

Table 1 Results of *L. barbarum* dataset division

Categories	Number of images			Number of labels		
	Front-light	Backlight	Total	Front-light	Backlight	Total
Training set	3867	2126	5993	38 647	17 714	56 361
Validation set	1149	563	1712	11 319	4533	15 852
Test set	540	317	857	5180	2695	7875
Total	5556	3006	8562	55 146	24 942	80 088

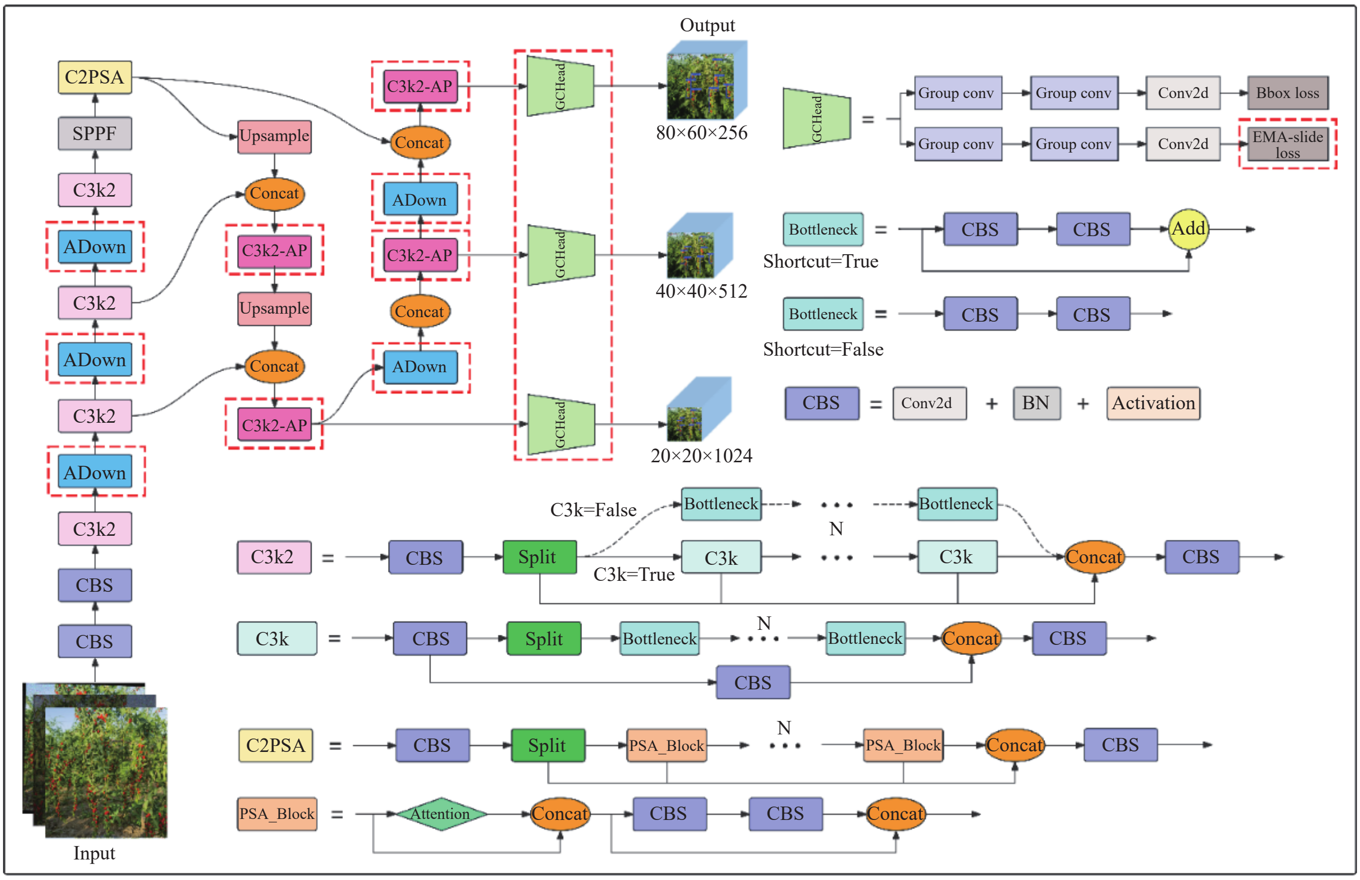
2.3 YOLOv11 baseline model

YOLOv11[30] is the latest iteration of the target detection algorithm developed by the Ultralytics team. According to the depth and width of the model, it is categorized into five variants: YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x. As the model size and computational volume increase, the accuracy of the model rises. In this study, the model needs to be deployed into the embedded device for the *L. barbarum* vibration harvesting operation in field conditions. Therefore, YOLOv11n with the smallest scale is selected as the baseline model to complete the detection of *L. barbarum* ripe fruit regions.

The YOLOv11 model consists of four parts: input, feature extraction backbone network, feature fusion neck network, and detection head. The input terminal transmits the dataset images after image preprocessing and Mosaic online data enhancement to the network. The backbone network is based on the CSPDarkNet architecture for bottom-up feature extraction and introduces C3k2, SPPF, and C2PSA modules to optimize the computational efficiency and feature extraction ability while maintaining lightweight. The C3k2 module adopts a parallel convolution branch structure, where the main branch extracts high-order features through multiple bottleneck layers (C3/Bottleneck), while the shortcut branch preserves the original features through ordinary convolution layers. Finally, the two are spliced to achieve cross-stage feature fusion. The structure of the bottleneck layer is determined by the parameter C3k. If C3k is set to True, the C3 module structure is used; if C3k is set to False, the basic Bottleneck module structure is used. Typically, shallow networks use the Bottleneck structure (C3k=False) and deep networks use the C3 module structure (C3k=True) for efficiency. The Spatial Pyramid Pooling-Fast (SPPF) module processes input features through cascaded max-pooling layers. The pooled outputs are concatenated with the original input features and fused via convolutional layers. This architecture efficiently expands the network's receptive field while capturing multi-scale contextual information, thereby enhancing robustness to scale variations of objects at different resolutions. The new C2PSA module is integrated into the last layer of the backbone network. This module introduces the pyramid split attention (PSA) mechanism based on the C2f architecture. By enhancing the model's ability to extract spatial information from feature maps, it further improves target detection performance in complex scenes. The neck network adopts the path aggregation network (PANet) structure, which performs top-down and bottom-up feature fusion on feature maps of different scales output from the backbone network to achieve cross-layer feature interaction. The detection head follows the previous decoupled head structure, which separates the predictive regression and classification tasks to accelerate the model convergence speed. At the same time, two depthwise separable convolution (DWSConv) modules are added to the classification branch to reduce model parameters and computation, thus further improving the model performance. In addition, YOLOv11 adopts DFL Loss+CIoU Loss as the predictive regression loss and BCE Loss as the classification loss to optimize the confidence, location, and class information of the model's predictive bounding box.

2.4 The proposed YOLO-RFR model

In complex orchard environments, the soft and messy hanging branches of *L. barbarum*, along with its small, densely distributed and easily occluded fruits, pose significant challenges for accurate detection. To address these challenges, this study proposed YOLO-RFR, a ripe fruit region detection model for precision vibration harvesting of *L. barbarum*, whose network structure is shown in Figure 4. The main improvements of this study are as follows:



Note: The red dashed boxes in the figure indicate the improvements proposed in this study. Specifically, ADown is used for downsampling and feature enhancement in the backbone and neck networks; C3k2-AP is used to improve multi-directional feature extraction; GCHead is used to enhance feature representation and spatial information modeling in the detection head; and EMA-Slide Loss is used to optimize hard-sample learning and improve model convergence stability.

Figure 4 YOLO-RFR network structure diagram

1) The Adown downsampling module was used to partially replace conventional convolutional layers in the backbone and neck network. The module significantly reduced model parameters by decreasing the spatial dimensions of the feature map and improved the detection accuracy of dense small targets.

2) The C3k2-AP module was designed to replace the original C3k2 module in the neck network based on the idea of asymmetric filling in pinwheel-shaped convolution (PConv). This module can effectively expand the receptive field with the least parameter increase, and improve the model's ability to extract spatial features in multiple directions.

3) Group convolution was employed to replace conventional convolution and depthwise separable convolution (DWSConv) in the original detection head, resulting in the construction of an improved GCHead detection head, which fully captured the target spatial information and improved the detection accuracy of *L. barbarum* ripe fruit regions.

4) The exponential moving average slide loss (EMA-Slide Loss) function was constructed to optimize the original classification loss function by introducing the Slide weighting function and combining it with the exponential moving average (EMA) technique. The function improved the model's focus on hard samples by dynamically adjusting the function threshold and effectively suppressing the fluctuations and noise in the parameter updating process, thereby enhancing the convergence stability of the model.

2.4.1 ADown downsampling module

The ADown downsampling module was initially applied to the backbone and neck network of YOLOv9[31], and its structure is shown in Figure 5. In this module, the input feature map is first processed by an average-pooling operation and then divided into two equal parts along the channel dimension. A part is directly processed by a conventional convolution, while the other part is sequentially passed through a max-pooling layer followed by a conventional convolutional layer, and the outputs of both parts are concatenated and fused to obtain the final result. Therefore, the ADown module was introduced to replace the conventional convolutional layer following the C3k2 module in the YOLOv11 network for downsampling, which enables a better balance between detection accuracy and model complexity.

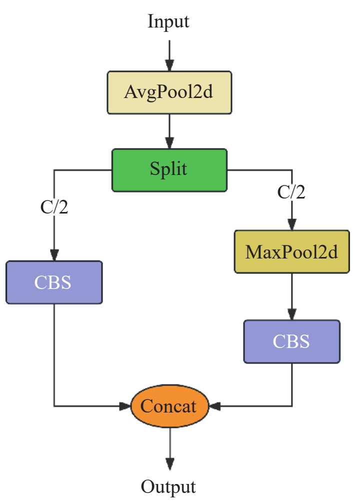


Figure 5 Structure of ADown module

2.4.2 C3k2-AP module

Yang et al.[32] proposed PConv (Pinwheel-shaped convolution) for the infrared small target detection and segmentation (IRSTDS) task. This convolution uses asymmetric padding to create horizontal and vertical convolution kernels for different regions of the image, which can better adapt to the pixel Gaussian spatial distribution features of small targets and expand the receptive field with minimal parameter increase while enhancing the low-level feature extraction.

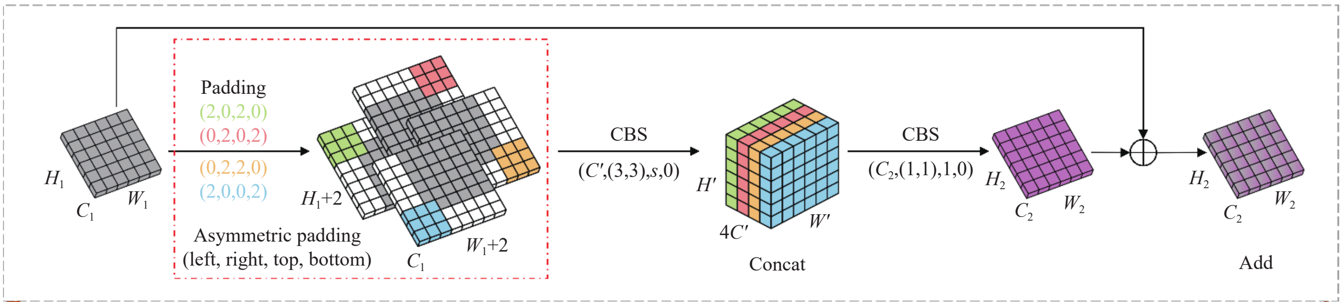
AP-Bottleneck[32] is constructed based on the asymmetric padding concept in PConv. Before convolution, the input feature map is not padded uniformly on all four sides; instead, pixels are added only along a specific side or direction. Specifically, four parallel branches are constructed in this study, which apply asymmetric padding to the input feature map in the left, right, top, and bottom directions, respectively. This enables the convolution kernels to obtain differentiated receptive fields in different directions, thereby allowing more targeted extraction of directional spatial features. It uses asymmetric padding with four different directions for parallel convolution, extracts features, and concatenates them in the channel dimension to enhance the receptive field expansion effect and multi-scale information extraction ability. In addition, the module adopts the residual connection structure to concatenate and fuse the output results with the input feature map, which facilitates the interaction of feature information. Therefore, the AP-Bottleneck structure can effectively improve the detection accuracy of ripe fruit regions on *L. barbarum* fruit-bearing branches in complex orchard environments.

As shown in Figure 6a, AP-Bottleneck first performs parallel convolution on the input feature map, and the detailed process is as follows:

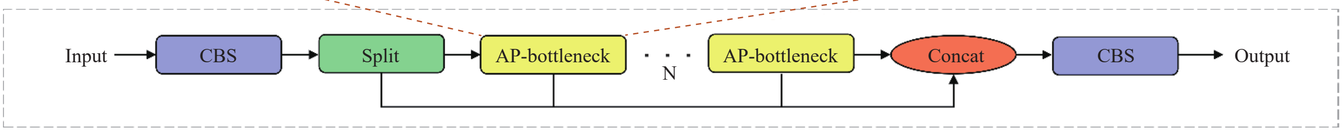
$$\begin{cases} X_1^{(H',W',C')} = SiLU(BN(X_{P(2,0,2,0)}^{(H_1,W_1,C_1)} \otimes W^{(3,3,C')})), \\ X_2^{(H',W',C')} = SiLU(BN(X_{P(0,2,0,2)}^{(H_1,W_1,C_1)} \otimes W^{(3,3,C')})), \\ X_3^{(H',W',C')} = SiLU(BN(X_{P(0,2,2,0)}^{(H_1,W_1,C_1)} \otimes W^{(3,3,C')})), \\ X_4^{(H',W',C')} = SiLU(BN(X_{P(2,0,0,2)}^{(H_1,W_1,C_1)} \otimes W^{(3,3,C')})). \end{cases} \tag{1}$$

$$H' = \frac{H_1}{s} + 1, W' = \frac{W_1}{s} + 1, C' = \frac{C_2}{4}, \tag{2}$$

where, H_1, W_1, *and* C_1 denote the height, width, and number of channels of the input feature map, respectively, while H', W', and C' denote the height, width, and number of channels of the convolutional output feature map of the layer, respectively. $P(n_1,n_2,n_3,n_4)$ denotes the number of padded pixels applied to the left, right, top, and bottom sides of the input feature map, respectively. Specifically, $P(2,0,2,0)$ indicates that 2 pixels are padded on the left side and top side of the input feature map, respectively. $\otimes$ represents the convolution operation and adds batch normalization (BN) and sigmoid linear unit (SiLU) activation function after each convolution, which helps to improve the training speed and accuracy of the model, and $W^{(3,3,C')}$ denotes a 3×3 convolution kernel with an output channel of C'. C_2 denotes the number of channels in the final output feature map of the AP-Bottleneck module, and s denotes the convolution stride, where s equals 1.



a. The AP-Bottleneck block extracts multi-directional spatial features through parallel convolutions with asymmetric padding in four directions and then achieves feature fusion via channel concatenation and residual connection



b. The C3k2-AP module embeds the AP-Bottleneck into the C3k2 structure to enhance the model's perception and extraction of multi-directional features in complex orchard environments

Note: The red-boxed region indicates the asymmetric padding process. The terms left, right, top, and bottom in parentheses denote the order of the padding parameters. Different padding patterns and their corresponding extracted feature maps are indicated using the same colors.

Figure 6 C3k2-AP module network structure

In order to obtain richer multi-directional feature information and enhance the network's direction-aware ability, especially for the morphological feature recognition of *L. barbarum* ripe fruit regions in different directions under natural conditions, this study concatenates the multi-directional feature maps extracted by the first convolution layer in the channel dimension, as formulated in Equation (3).

$$X'^{(H',W',C_2)} = Cat(X_1^{(H',W',C')}, ..., X_4^{(H',W',C')}). \tag{3}$$

A 1×1 convolutional kernel $W^{(1,1,C_2)}$ rearranges and integrates the concatenated output results to achieve feature fusion and simultaneously adjusts the output dimensions to the preset values H_2, W_2, and C_2, as shown in Equation (4). This approach not only reduces unnecessary computational overhead, but also enhances the feature representation capability of the model and improves the detection performance of *L. barbarum* ripe fruit regions under varying lighting conditions, shading, and directional changes.

$$X^{(H_2,W_2,C_2)} = SiLU(BN(X'^{(H',W',4C')}) \otimes W^{(1,1,C_2)}) \tag{4}$$

When the number of input channels C_1 is equal to the number of output channels C_2, the residual connection is introduced to add the upper-layer output feature map and the input feature map to fuse the newly extracted direction-aware features and the key feature information in the original input to enhance the robustness and stability of the model to different direction features. In addition, this operation can effectively alleviate the problem of gradient disappearance and improve the training effect of the deep network. Its output $Y^{(H_2,W_2,C_2)}$ is calculated as follows:

$$Y^{(H_2,W_2,C_2)} = X^{(H_1,W_1,C_1)} \oplus X^{(H_2,W_2,C_2)} \tag{5}$$

In natural environments, the regions of *L. barbarum* ripe fruit are prone to occlusion, posing greater challenges to target detection. In addition, textures, shadows, and hanging fruit features in natural scenes have significant directionality. Therefore, this study proposes a C3k2-AP module by integrating the asymmetric padding mechanism of AP-Bottleneck into the C3k2 module architecture, as shown in Figure 6b. It replaces the bottleneck layer in the C3k2 module with the AP-Bottleneck structure to enhance the ability of the C3k2-AP module to perceive spatial features in different directions, thereby improving the model's ability to detect *L. barbarum* ripe fruit regions in complex orchard environments.

2.4.3 GCHead detection head

YOLOv11 uses depthwise separable convolution (DWSConv) in the classification branch of the detection head to significantly reduce parameters and computational complexity. However, DSConv extracts features only for each input channel separately, resulting in the loss of spatial feature information between different channels. To enhance the spatial information interaction ability between channels and improve the detection performance of *L. barbarum* ripe fruit regions, this study introduced group convolution[33] to replace the ordinary convolution and DWSConv in the YOLOv11 detection head to construct the GCHead detection head. The structure of group convolution is shown in Figure 7.

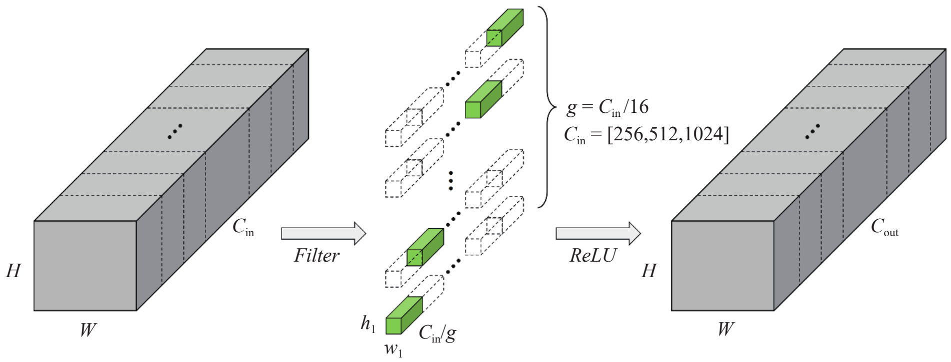


Figure 7 Structure of group convolution

In group convolution, the group number g simultaneously affects computational complexity and feature extraction capability. As g increases, the computational complexity decreases, but the number of channels contained in each group is reduced, weakening cross-channel information interaction and potentially leading to a decline in feature representation ability. Conversely, if g is too small, the lightweight advantage becomes less significant. Therefore, a proper choice of the group number g is crucial for balancing computational complexity and detection performance.

Considering the lightweight requirement of the model and its detection performance, this study adopts a strategy of dynamically adjusting the group number according to the input channel number C_{in}, as shown in Equation (6), to ensure that each group convolution retains an appropriate number of channels for feature learning. For the three input scales of the detection head in this study, C_{in} = [256, 512, 1024], and the corresponding group numbers g are 16, 32, and 64, respectively.

$$g = C_{in}/16 \tag{6}$$

2.4.4 EMA-Slide Loss function

Slide Loss[34] is a loss function used to solve the sample imbalance problem in deep learning, especially to improve the model's ability to handle difficult samples. The samples are classified into easy and hard samples based on the intersection over union (IoU) value of the prediction and ground truth bounding box. The average IoU value of all bounding boxes is used as the threshold value μ. The samples with an IoU smaller than μ are regarded as easy samples, while those with an IoU greater than μ are considered hard samples. To address the issue of higher loss values for samples near the boundary μ that are difficult to detect, the Slide weighting function is introduced. It applies sliding weights to samples near the boundary, aiming to balance the learning of easy and hard samples by assigning higher weights to the hard samples, so as to improve the model's stability and generalization ability. The specific form of the Slide weighting function is as shown in Equation (7), where x denotes the IoU value of any bounding box and μ denotes the weight threshold.

$$f(x) = \begin{cases} 1, & x \leq \mu - 0.1 \\ e^{1-\mu}, & \mu - 0.1 < x < \mu \\ e^{1-x}, & x \geq \mu \end{cases} \tag{7}$$

However, when addressing the sample imbalance problem, Slide Loss only relies on the instantaneous threshold value while neglecting the threshold at the previous moment. During parameter updates, this may cause significant fluctuations, thereby slowing down the convergence speed of model training. To mitigate this limitation, this study constructed EMA-Slide Loss[35-38] by integrating the exponential moving average (EMA) technique into the Slide Loss framework, as illustrated in Figure 8. This method dynamically adjusts the boundary thresholds based on prior data and reasonably assigns the loss weights according to the updated threshold to improve the robustness and convergence speed of the model. The EMA-Slide Loss can be expressed as Equations (8)-(10).

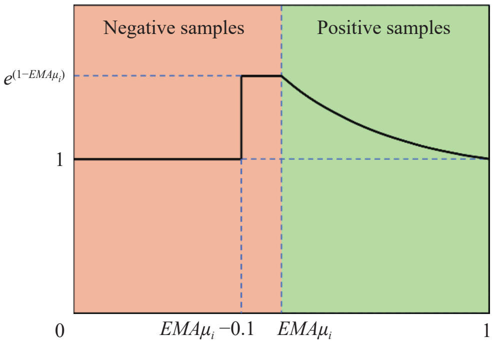


Figure 8 EMA-Slide Loss

$$EMA\mu_i = \alpha_i \cdot EMA\mu_{i-1} + (1-\alpha_i) \cdot \theta_i \tag{8}$$

$$\alpha_i = \lambda \cdot \left(1 - e^{-\frac{i}{tau}}\right) \tag{9}$$

$$f'(x) = \begin{cases} 1, & x \le EMA\mu_i - 0.1 \\ e^{1-EMA\mu_i}, & EMA\mu_i - 0.1 < x < EMA\mu_i \\ e^{1-x}, & x \ge EMA\mu_i \end{cases} \tag{10}$$

In the task of ripe fruit regions detection in *L. barbarum*, the hanging branches are intricate and multi-layer distributed; the outermost naturally drooping ripe fruit regions have obvious features, which are relatively easy to detect. However, the inner branches are more difficult to detect due to the serious mutual occlusion, resulting in the uneven distribution of the samples. In addition, there are more noises in natural scenes, which puts forward higher requirements for the stability of the model. To this end, this study introduces the EMA-Slide Loss to optimize the classification loss function BCEWithLogits Loss in the original model, which makes the model pay more attention to the learning of hard samples, so as to improve the detection ability of the model for small targets and reduce the fluctuations and noises during training.

2.5 Model training

The test followed the principle of control variables, and the test hardware and software are consistent. The hardware configuration of the platform includes an Intel(R) Xeon(R) Silver 4210 CPU (2.20 GHz), an NVIDIA Quadro RTX 4000 GPU, and 64 GB RAM. The software environment is Windows 10 operating system. In this environment, CUDA 12.1, Python 3.10, and PyTorch 2.1.0 deep learning framework have been constructed, with PyCharm 2021 used as the programming platform. The model input size is 640×640, the training epoch is 300, the batch size is 16, and model parameters are updated using SGD stochastic gradient descent with an initial learning rate of 0.01, a momentum parameter of 0.923, and a weight decay factor of 0.0005.

2.6 Evaluation metrics

The performance of a target detection model is typically evaluated in terms of detection accuracy, inference speed, and model complexity. For detection accuracy, this study selects the metrics of Precision (*P*), Recall (*R*), and mean Average Precision (mAP) to effectively evaluate the model's performance, and their calculation methods are as shown in Equations (11)-(14).

$$P = \frac{TP}{TP+FP} \tag{11}$$

$$R = \frac{TP}{TP+FN} \tag{12}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{13}$$

$$AP = \int_0^1 P(R)dR \tag{14}$$

where, TP denotes the number of instances in which the actual label is *L. barbarum* and the predicted label is also *L. barbarum*; FP denotes the number of instances in which the actual label is not *L. barbarum* but the predicted label is *L. barbarum*; TN denotes the number of instances in which the actual label is not *L. barbarum* and the predicted label is not *L. barbarum*; and FN denotes the number of instances in which the actual label is *L. barbarum* but the predicted label is not *L. barbarum*. Precision represents the proportion of samples that are correctly detected among all samples that have a positive predicted label. Recall represents the proportion of samples that are correctly detected when the actual label is positive. AP represents the average precision for each category and can be obtained by integrating the Precision – Recall (*P*-*R*) curve as shown in Equation (13). mAP refers to the mean average precision of all classes at an IoU threshold of 0.5. In addition, the value of *n* represents the number of categories of detected objects in the object detection task. In this study, with only one class named "goji", *n* equals 1.

In terms of detection speed, frames per second (FPS) is used to evaluate the inference speed of the model on a given hardware, which is an important indicator of the real-time detection capability of the model. Finally, model parameter count (Parameters), floating-point operations (FLOPs), and model size are used to evaluate the model complexity.

3 Results and discussion

3.1 Ablation experiments

To further validate the effectiveness of the ADown downsampling module, the C3k2-AP module, the GCHead detection head, and EMA-Slide Loss function in enhancing the detection performance of *L. barbarum* ripe fruit regions, a series of ablation experiments were conducted in this study by progressively integrating each component to YOLOv11n baseline model. The results of the ablation experiments are listed in Table 2.

Table 2 Results of the ablation experiments

Model	*A*	*B*	*C*	*D*	P/ (%)↑	R/ (%)↑	mAP/ (%)↑	Parameters/ (M)↓	FLOPs/ (G)↓	Model size/ (MB)↓
1	×	×	×	×	89.7	87.6	94.4	2.6	6.3	5.5
2	√	×	×	×	92.2	90.6	96.2	2.1	5.3	4.6
3	×	√	×	×	91.6	89.4	95.6	2.5	6.3	5.3
4	×	×	√	×	90.5	88.6	95.1	2.3	5.0	4.9
5	×	×	×	√	89.7	88.5	94.4	2.6	6.3	5.6
6	√	√	×	×	93.3	90.9	96.8	2.0	5.3	4.4
7	√	√	√	×	93.7	91.6	97.0	1.7	4.1	3.8
8	√	√	√	√	**93.7**	**92.0**	**97.0**	**1.7**	**4.1**	**3.8**

Note: √ represents the addition of the module. *A* is ADown module, *B* is a C3k2-AP module, *C* is a GCHead detection head, and *D* is an EMA-Slide Loss function. ↑indicates an upward trend, and ↓indicates a downward trend. The Bold font represents the best results of the experiment.

The experimental results demonstrated that incorporating the ADown downsampling module, the C3k2-AP module, the GCHead detection head, and EMA-Slide Loss function into the YOLOv11n baseline model, as well as exploring the strategy of multi-module integration, led to varying degrees of improvement in both detection accuracy and model lightweighting. By replacing the standard convolutional layers following the C3k2 modules in both the backbone and neck of the YOLOv11n model with the ADown downsampling module, the improved model achieved a 15.9% reduction in FLOPs compared to the original YOLOv11n. Notably, the most significant lightweight advantages of this modification were reflected in a 19.2% decrease in parameters and a 16.4% reduction in model size. Meanwhile, the detection performance was also enhanced, with precision, recall, and mAP increasing by 2.5%, 3.0%, and 1.8%, respectively. These results demonstrated that the introduction of the ADown module not only effectively reduced the overall model complexity but also enhanced target detection accuracy. The constructed C3k2-AP module brought slight optimization in model lightweighting, particularly in terms of parameters and model size, but it yielded more significant gains in target detection accuracy. Compared to the YOLOv11n model, the modified model achieved improvements of 1.9%, 1.8%, and 1.2% in precision, recall, and mAP, respectively, which validated that the asymmetric padding mechanism enhanced the model's ability to perceive spatial features in different directions, thus improving its capability to detect *L. barbarum* ripe fruit regions in complex natural orchard environments. The introduction of the GCHead detection head led to improvements of 0.8%, 1%, and 0.7% in precision, recall, and mAP, respectively, compared to YOLOv11n. At the same time, the model complexity was significantly reduced, with a 20.6% decrease in FLOPs, representing the most notable lightweighting effect among all individual improvements. This demonstrated that group convolution could effectively enhance the interaction of spatial information across channels and that a well-designed grouping strategy better balanced computational complexity and feature extraction capability. EMA-Slide Loss dynamically adjusted the boundary thresholds based on historical samples and allocated loss weights accordingly, thereby enhancing the model's ability to detect hard samples. Without affecting model complexity, this function improved recall by 0.4%. Finally, by sequentially integrating the above four modules into the YOLOv11n model, the proposed YOLO-RFR model achieved progressive improvements in both detection accuracy and model lightweighting, ultimately attaining optimal values across various performance metrics. These results indicate strong synergistic effects among the modules, further validating the effectiveness of the proposed methods in the task of detecting *L. barbarum* ripe fruit regions.

In addition, to more intuitively evaluate the learning capability of the YOLO-RFR model in the detection of *L. barbarum* ripe fruit regions, this study employed the KPCA-CAM[39] method to visualize the heatmaps generated by the YOLOv11n and YOLO-RFR models. Three feature maps corresponding to the detection head were selected for visualization to investigate the models' focus on different image regions during the detection process. Figure 9 presents the visualization results for multiple input images.

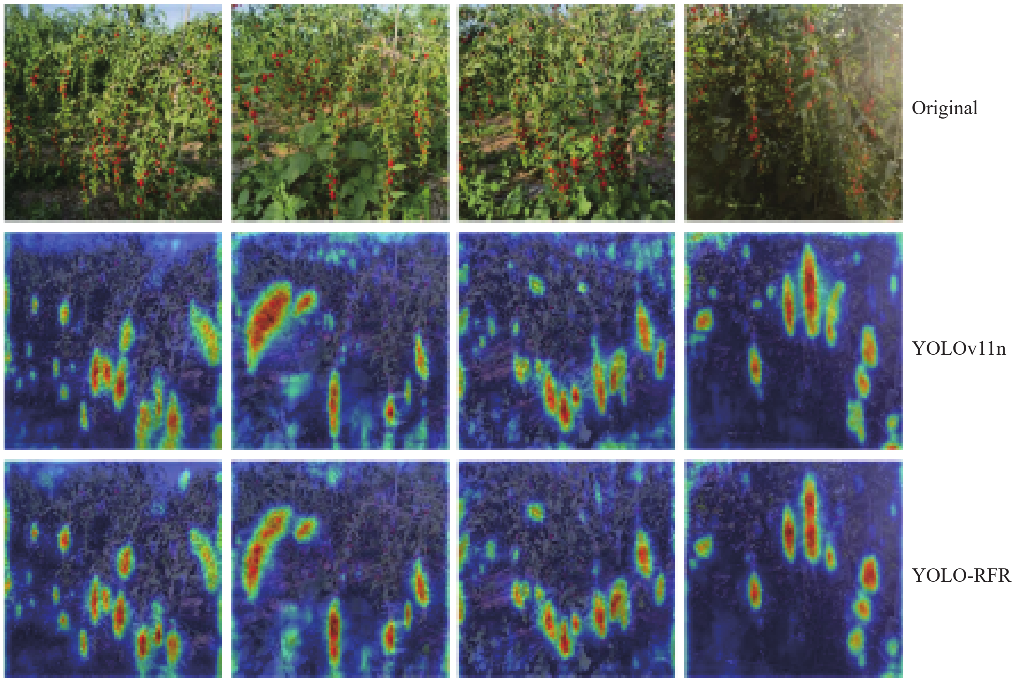


Figure 9 Heatmaps of different models based on KPCA-CAM

Compared to YOLOv11n, the YOLO-RFR model exhibited response regions in the heatmaps that were more closely aligned with the spatial distribution of the target fruits, indicating that it was able to more accurately focus on *L. barbarum* ripe fruit regions while suppressing attention to surrounding non-target areas such as leaves and branches. As a result, the model demonstrated enhanced robustness and key feature extraction capability in complex environments, further validating its detection performance for ripe fruit regions and providing effective support for the precise localization of vibration-picking points.

3.2 Comparison experiments of different target detection models

Based on the YOLOv11n model, this study proposed a novel lightweight object detection model named YOLO-RFR through a series of architectural improvements. To validate the advantages of the proposed model in detecting *L. barbarum* ripe fruit regions, comparison experiments were conducted against several state-of-the-art lightweight object detection models, including YOLOv7-tiny[40], YOLOv8n[41], YOLOv9t[31], YOLOv10n[42], as well as the baseline model YOLOv11n.

The results of the comparison experiments are listed in Table 3. It was observed that the YOLO-RFR model exhibited the best overall performance in both target detection accuracy and model lightweighting. In terms of target detection accuracy, the YOLO-RFR model achieved an mAP of 97.0%, which was 1.8%, 4.4%, 2.8%, 1.1%, and 2.6% higher than that of YOLOv7-tiny, YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n, respectively. In addition, the YOLO-RFR model also demonstrated lower complexity compared to the other models, with 1.7 M, 4.1 G, and 3.8 MB of parameters, FLOPs, and model size, respectively, which is 34.6%, 34.9%, and 30.9% less than the baseline model YOLOv11n, respectively.

Table 3 Results of the comparison experiments

Model	*P*/(%)↑	*R*/(%)↑	mAP/(%)↑	Parameters/(M)↓	FLOPs/(G)↓	Model size/(MB)↓
YOLOv7-tiny	90.9	90.3	95.2	6.0	13.2	12.3
YOLOv8n	88.1	85.5	92.6	3.0	8.1	6.3
YOLOv9t	85.5	82.6	90.3	2.0	7.6	4.7
YOLOv10n	87.8	85.1	92.7	2.7	8.2	5.8
YOLOv11n	89.7	87.6	94.4	2.6	6.3	5.5
YOLO-RFR	**93.7**	**92.0**	**97.0**	**1.7**	**4.1**	**3.8**

Note: The Bold font represents the best results of the experiment.

The above results indicate that YOLO-RFR achieves higher detection accuracy while maintaining a lightweight model structure. This advantage is attributable not only to the effective reduction in the number of parameters and computational complexity, but also to the good compatibility between the proposed improved modules and the morphological characteristics of ripe *L. barbarum* fruit regions. In complex orchard environments, ripe fruit regions are typically characterized by high target density, small scale, variable spatial orientations, and frequent occlusion by branches and leaves. These characteristics impose higher requirements on the model's ability to preserve small-target features, extract multi-directional features, and enhance cross-channel information interaction. To address these challenges, the ADown module improves the retention of critical information from dense small targets during downsampling while reducing parameters and computation; the C3k2-AP module enhances the perception and extraction of spatial features in different directions through asymmetric padding and parallel convolution, making it better suited to the variable orientations and pronounced local occlusions of ripe fruit regions; the GCHead module strengthens cross-channel interaction and feature representation in the detection head through group convolution, thereby improving the extraction of key spatial information while controlling complexity. In addition, EMA-Slide Loss further increases the model's focus on hard samples, thereby improving training stability and classification optimization. Therefore, YOLO-RFR is able to improve detection accuracy while significantly reducing the number of parameters, indicating that the improved model proposed in this study is well adapted to the morphological characteristics of ripe fruit regions in complex orchard environments.

Figure 10 shows the Precision-Recall (P-R) curves of different models on the test set, where the horizontal axis represents recall, and the vertical axis represents precision. These curves further reflected the trade-off between precision and recall under varying confidence thresholds. It was observed that the area of the P-R curve bounded by the axis of the YOLO-RFR model was significantly larger than that of the other models. Moreover, the balance point where precision equaled recall (P = R) was closer to the ideal point (1,1), indicating that the YOLO-RFR model achieved the best detection performance among all the compared models.

The detection performance of the above six detection models was evaluated on the test set under different lighting conditions, including front-lighting and back-lighting. Several representative images were randomly selected for visual comparison, as shown in Figure 11. In the figure, the yellow ellipse indicates *L. barbarum* ripe fruit regions that the model failed to detect, while the red ellipse indicates incorrectly detected regions. These annotations provide a more intuitive assessment of each model's effectiveness in detecting *L. barbarum* ripe fruit regions. It can be observed that, except for YOLO-RFR, the other models exhibited varying degrees of missed detection under both front-lighting and back-lighting conditions, which may be attributed to occlusion by branches and leaves, resulting in unclear target features. In addition, YOLOv7-tiny, YOLOv8n, and YOLOv10n also demonstrated certain false detection issues, mainly involving incorrect identification of small distant targets and repeated detection of the same target. This may be related to weakened spatial distribution features caused by densely clustered fruits and disordered branch growth. Overall, the YOLO-RFR model demonstrated robust detection performance for *L. barbarum* ripe fruit regions under various lighting conditions. Moreover, its lightweight network architecture enabled efficient deployment on embedded devices, providing strong technical support for precision vibration harvesting of *L. barbarum*.

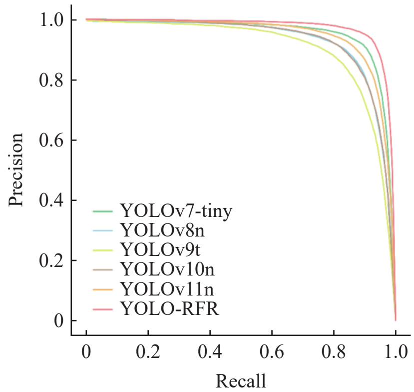


Figure 10 P-R curves of different models

3.3 Deployment on embedded device

To evaluate the deployment performance of the YOLO-RFR model on an embedded device, this study deployed it on the NVIDIA Jetson AGX Xavier platform and utilized TensorRT for inference acceleration. The detailed deployment environment configuration is listed in Table 4.

The model was first converted from PyTorch format to an FP16-precision ONNX file and then optimized and compiled using the TensorRT toolchain to generate a serialized inference engine file (.engine) to improve inference speed. During deployment, FP16 precision mode was enabled to fully exploit the hardware acceleration capabilities of Jetson AGX Xavier platform for half-precision computation, thereby further enhancing inference speed and reducing computational resource consumption. The model inference was executed in maximum performance mode with the batch size set to 1. As shown in Table 5, when the input resolution was 640×640, the YOLO-RFR model achieved an inference speed of 163 FPS on Jetson AGX Xavier, meeting the real-time detection requirements of low-latency industrial applications. At the same time, compared with the 41 FPS before TensorRT acceleration, the model inference speed increased by 3.98 times, indicating that the improved model structure not only enhances detection accuracy but also maintains high acceleration performance.

3.4 Discussion

At present, research on target detection for *L. barbarum* is relatively limited, with most studies focusing on the detection of individual *L. barbarum* fruits and their pedicels. Wang et al.[26] achieved a fruit detection accuracy of 92.7% and a pedicel picking point localization accuracy of 86.4% for *L. barbarum* under natural conditions, effectively addressing the challenges of fruit detection and pedicel picking point localization in natural environments. The detection and localization algorithm proposed in their study is more suitable for fine-scale harvesting scenarios targeting individual fruits, such as laser cutting or intelligent picking robots. In contrast, the present study focuses on the practical requirements of precision vibration harvesting of *L. barbarum*, aiming to achieve fast and accurate detection of ripe fruit regions and to precisely localize vibration picking points within suitable harvesting zones, thus differing significantly from existing studies in terms of application scenarios. Therefore, this section reviews relevant studies on the detection of string-type and cluster-type fruits and compares their detection performance with that of the proposed YOLO-RFR model. Xiong et al. achieved a detection accuracy of 93.75% for litchi clusters in nighttime environments[43]. Li et al. attained a detection accuracy of 91.08% for fresh grapes under complex background conditions[44]. Zhao et al. proposed a grape bunch detection model that achieved 93.27% accuracy while reducing the model size to 21.3 MB, achieving over 10% improvement in model compactness compared to the baseline[45]. The YOLO-RFR model proposed in this study achieved 97% accuracy in the task of detecting *L. barbarum* ripe fruit regions, and the model size is only 3.8 MB, demonstrating both high precision and lightweight advantages, and its performance outperformed that of existing comparable models. The model not only accurately identified *L. barbarum* ripe fruit regions but also exhibited excellent real-time detection performance on Jetson AGX Xavier embedded device.

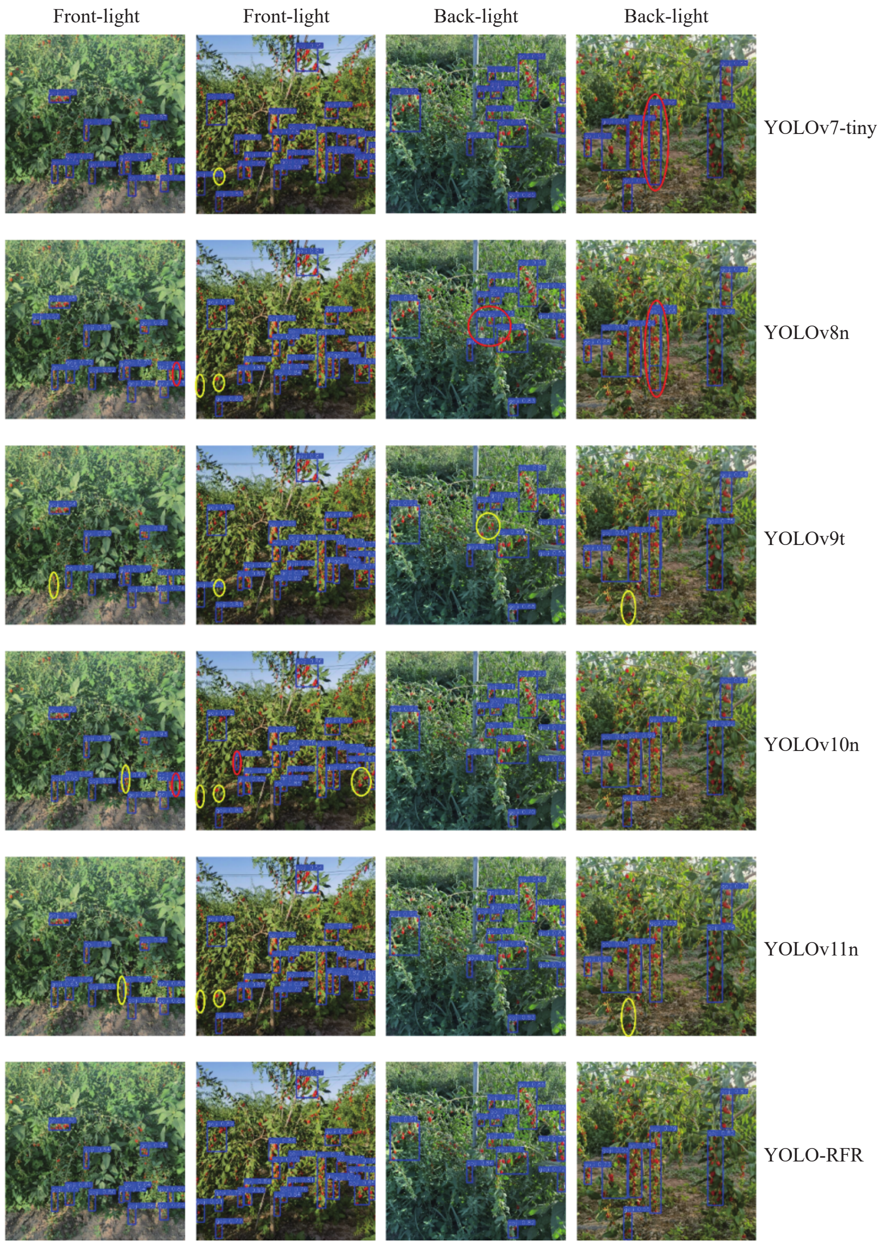


Figure 11 Comparison of detection performance of different models

Although the YOLO-RFR model achieved significant progress in both detection accuracy and lightweight design, there are still certain limitations in practical harvesting operations which need to be discussed and optimized in depth. In this study, images of the *L. barbarum* cultivar “Ningqi No. 7” were collected in Guyuan City, Ningxia Hui Autonomous Region, to construct a custom dataset for detecting *L. barbarum* ripe fruit regions. Due to the complex structure of the *L. barbarum* branches and frequent fruit occlusion, as well as variations in planting patterns, varieties, and others, the target detection task may bring great challenges. Therefore, the generalization ability of the improved model still requires further validation. Although the model developed in this study achieved good detection performance under both front-lighting and backlighting conditions, more complex situations may still arise in actual field environments, such as strong backlighting, overlapping shadows from branches and leaves, or plant swaying, which may affect the stability of detection to some extent. Future research should continue to expand the diversity of the *L. barbarum* dataset by including samples from more regions, cultivars, and planting patterns, while also increasing the coverage of samples collected under complex lighting and dynamic scene conditions. In addition, further studies may be conducted from the perspectives of multi-source information utilization and continuous-scene feature modeling to improve the model's adaptability, generalization ability, and robustness in complex orchard environments.

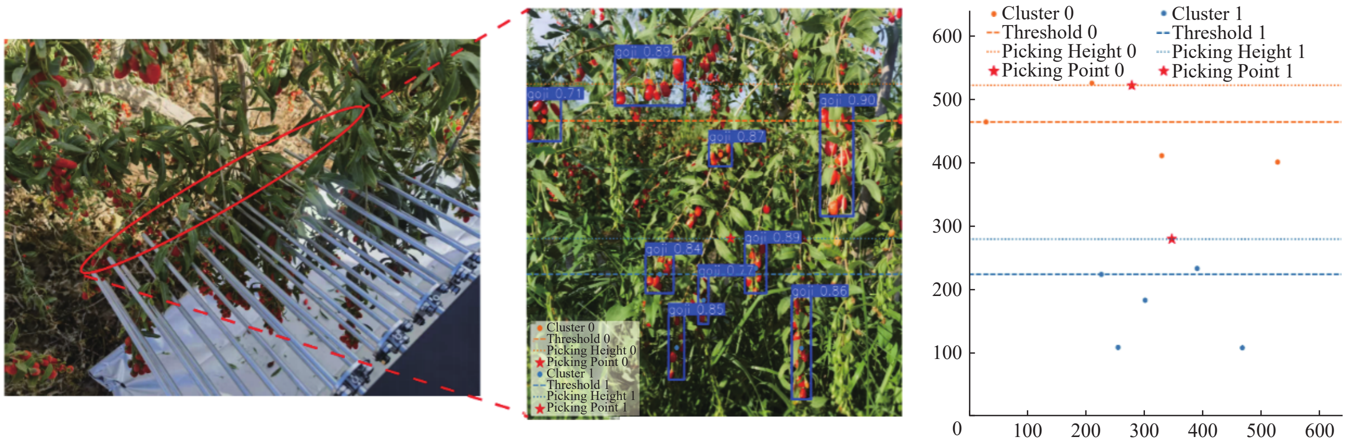
Existing vibration harvesting equipment still faces a significant trade-off between operational efficiency and fruit damage rate, and an effective balance between the two has not yet been achieved. To simultaneously improve harvesting efficiency and fruit quality, future work will deeply integrate the ripe fruit region detection method proposed in this study with the multi-point vibration harvesting device independently developed by our team. As shown in Figure 12, multiple torsional vibration picking heads are arranged in an array along the horizontal direction at the vibrating end-effector, enabling the same vibration excitation to be applied above ripe fruit regions with similar height distributions, thereby achieving precise vibration harvesting of *L. barbarum*.

Table 4 Deployment environment of NVIDIA Jetson AGX Xavier

Deployment environment	Version
Embedded device	Jetson AGX Xavier
Operating system	Ubuntu 20.04
JetPack	5.1.2
TensorRT	8.5.2.2
CUDA	11.4
python	3.8
torch	2.1.0
torchvision	0.16.0

Table 5 Embedded devices deployment results

Model	FPS (Embedded device)	FPS (TensorRT)
YOLOv11	39	-
YOLO-RFR	41	163



Note: Cluster represents the different harvesting regions segmented by clustering, with numbers indicating the actual operation sequence. Threshold denotes the bounding boxes of fruit targets that meet the 80% fruit density criterion within each harvesting region. Picking Height refers to the vertical position where the vibration excitation is applied, while Picking Point indicates the optimal vibration picking point for each harvesting region.

Figure 12 Vibration picking point localization method

Specifically, the *L. barbarum* ripe fruit regions were detected, and then *K*-means clustering analysis was carried out according to the height distribution characteristics of the regions. The one-dimensional height coordinate of the center point of each ripe fruit region is used as the input feature, and Euclidean distance is adopted as the metric to divide the canopy of each plant into harvesting sub-regions that can serve as vibration harvesting units. Considering that the transmission range of vibration excitation along branches is limited, the vibration picking points are further precisely determined based on the density distribution of ripe fruit regions. Within each harvesting sub-region, the height at which the number of ripe fruit region targets reaches 80% of the total targets in that region is taken as the threshold for determining the vibration picking height (Figure 12). Combined with the horizontal geometric center of the target harvesting sub-region and the depth information obtained by the ZED stereo camera, the coordinates of the target harvesting sub-region are mapped from image space to three-dimensional space, and the three-dimensional coordinates of the vibration picking point are then solved and mapped to the target position coordinates corresponding to the geometric center of the arrayed vibration rod region. The operation control system drives the platform to guide the vibration device to accurately insert into fruit-bearing branches within the canopy and apply vibration excitation above ripe fruit regions with similar height distributions. The harvesting process was conducted hierarchically from top to bottom to reduce the number of operations per plant, thereby balancing the harvesting efficiency and fruit quality of *L. barbarum* vibration harvesting and promoting the development of intelligent and precise *L. barbarum* harvesting.

4 Conclusions

To address the low level of intelligence and precision in the vibration harvesting of *L. barbarum*, this study proposed the YOLO-RFR model and deployed it on NVIDIA Jetson AGX Xavier embedded device. The model was designed to achieve fast and accurate detection of *L. barbarum* ripe fruit regions, thereby providing technical support for the precise localization of vibration picking points and promoting the development of precision vibration harvesting for *L. barbarum*. The main conclusions are as follows:

1) The ADown downsampling module was incorporated to replace certain conventional convolutional layers in the backbone and neck networks. This effectively reduced model complexity while improving its ability to detect dense small targets. Inspired by the asymmetric padding concept in PConv, a novel C3k2-AP module was designed to replace the original C3k2 module in the neck network. This expanded the receptive field and enhanced the model's ability to perceive spatial features in multiple directions, thereby better adapting to the multi-directional growth patterns of fruits on *L. barbarum* fruit-bearing branches. In addition, group convolution was employed to optimize the original detection head to construct the GCHead detection head, which strengthened the spatial information interaction ability of the model between channels. Finally, by introducing the Slide weighting function and combining it with the exponential moving average (EMA) technique, the EMA-Slide Loss function was proposed to optimize the original classification loss function. This not only improved convergence stability but also increased the model's attention to hard samples, thereby enhancing its capability in detecting difficult small targets.

2) The effectiveness of each improved module and the overall performance of the model were validated through ablation and comparison experiments. The results demonstrated that the YOLO-RFR model outperformed mainstream object detection algorithms in detection accuracy and model lightweighting. On the custom *L. barbarum* dataset, the model achieved 97% mAP with only 1.7 M parameters, 4.1 G FLOPs, and a model size of 3.8 MB, which were reduced by 34.6%, 34.9%, and 30.9%, respectively, compared to the baseline model YOLOv11n. To further validate the YOLO-RFR model's potential for practical applications, it was deployed on NVIDIA Jetson AGX Xavier embedded device and accelerated using TensorRT, achieving a real-time inference speed of 163 FPS. This meets the real-time requirements of intelligent harvesting systems and provides technical support for precision vibration harvesting of *L. barbarum*.

Acknowledgements

This study was funded by the National Natural Science Foundation of China (Grant No. 32272001) and the Major Science and Technology Program of Ningxia Hui Autonomous Region (Grant No. 2022BBF01002).

[References]

[1] Lu Y Y, Guo S, Zhang F, Yan H, Qian D W, Wang H Q, et al. Comparison of functional components and antioxidant activity of *Lycium barbarum* L. fruits from different regions in China. Molecules, 2019; 24: 2228.

[2] Yu J, Yan Y M, Zhang L T, Mi J, Yu L M, Zhang F F, et al. A comprehensive review of goji berry processing and utilization. Food Science and Nutrion, 2023; 11(12): 7445–7457.

[3] Yu J, Yan Y M, Zhang L T, Mi J, Yu L M, Zhang F F, et al. A comprehensive review of goji berry processing and utilization. Food Science & Nutrition, 2023; 11(12): 7445–7457.

[4] Xiang W J, Wang H W, Tian Y, Sun D W. Effects of salicylic acid combined with gas atmospheric control on postharvest quality and storage stability of wolfberries: Quality attributes and interaction evaluation. Journal of Food Process Engineering, 2021; 44(8): e13764.

[5] Liu Z L, Xie L, Zielinska M, Pan Z, Deng L Z, Zhang J S, et al. Improvement of drying efficiency and quality attributes of blueberries using innovative far-infrared radiation heating assisted pulsed vacuum drying (FIR-PVD). Innovative Food Science & Emerging Technologies, 2022; 77: 102948.

[6] Li Y J, Hu Z Q, Zhang Y P, Wang J, Xu J H. Research progress of technology and equipment for mechanized harvest of wolfberry. J Chin Agric Mech, 2024; 45: 16–21, 35. (in Chinese)

[7] Zhang W Q, Zhang M M, Zhang J X, Li W. Design and experiment of vibrating wolfberry harvester. Transactions of the CSAM, 2018; 49(7): 97–102. (in Chinese)

[8] Chen Q Y, Zhang S X, Wei N S, Fan Y L, Zhang W, Wang Z Y, et al. Optimizing the parameters for the vibration harvesting of *Lycium barbarum* L. under various excitation modes. Transactions of the CSAE, 2025; 41: 32–42. (in Chinese)

[9] Mei S, Tang D B, Shi Z G, Song Z Y, Tian Z C, Zhou R. Design and test of a Chinese wolfberry harvester using arrayed vibration units. Transactions of the CSAE, 2024; 40(23): 115–125.

[10] Wang Z H, Xun Y, Wang Y K, Yang Q H. Review of smart robots for fruit and vegetable picking in agriculture. Int J Agric Biol Eng, 2022; 15(1): 33–54.

[11] Zhou J G, Wang Y K, Chen J, Luo T Y, Hu G R, Jia J L, et al. Research hotspots and development trends of harvesting robots based on bibliometric analysis and knowledge graphs. Int J Agric Biol Eng, 2024; 17(6): 1–10.

[12] Wen S W, Ge Y H, Wang Y K, Wei N S, Zhou J G, Hu G R, et al. Efficient and comprehensive visual solution for a smart apple harvesting robot in complex settings via multi-class instance segmentation. Int J Agric Biol Eng, 2025; 18(4): 200–215.

[13] Wen S W, Zhang D Y, Wei N S, Ge Y H, Chen J, Huang T L, et al. An improved Yolov11n-based Genet for missing-seed detection and counting in an oblique hook-shaped spoon-type small precision seed metering device. INMATEH Agric Eng, 2025; 77(3): 490–501.

[14] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus: IEEE, 2014; pp.580–587. doi: 10.48550/arXiv.1311.2524.

[15] Girshick R. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago: IEEE, 2015; pp.1440–1448. doi: 10.1109/ICCV.2015.169.

[16] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016; 39(6): 1137–1149.

[17] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, et al. SSD: Single shot multiBox detector. In: Computer Vision – ECCV 2016, 2016; 9905: pp.21–37. DOI:10.1007/978-3-319-46448-0_2

[18] Lin T Y, Goyal P, Girshick R, He K M, Dollar P. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020; 42(2): 318–327.

[19] Duan K W, Bai S, Xie L X, Qi H G, Huang Q M, Tian Q. CenterNet: Keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer, Seoul: IEEE, 2019; pp.6568–6577. doi: 10.48550/arXiv.1904.08189.

[20] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas: IEEE, 2016; pp.779–788. doi: 10.1109/CVPR.2016.91.

[21] Liu Q, Lv J, Zhang C P. MAE-YOLOv8-based small object detection of green crisp plum in real complex orchard environments. Computers and Electronics in Agriculture, 2024; 226: 109458.

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

[23] Parvathi S, Selvi S T. Detection of maturity stages of coconuts in complex background using Faster R-CNN model. Biosystems Engineering, 2021; 202: 119–32.

[24] Liu Z F, Abeyrathna R M R D, Sampurno R M, Nakaguchi V M, Ahamed T. Faster-YOLO-AP: A lightweight apple detection algorithm based on improved YOLOv8 with a new efficient PDWConv in orchard. Computers and Electronics in Agriculture, 2024; 223: 109118.

[25] Qin W J, Kang F, Wang Y X, Chen C C Tong S Y. Target detection of *Lycium barbarum* fruit based on improved YOLOv8 algorithm. Mech Eng Autom, 2024: (6): 24–26, 30.

[26] Wang J N, Tan D Z, Sui L M, Guo J, Wang R W. Wolfberry recognition and picking-point localization technology in natural environments based on improved Yolov8n-Pose-LBD. Computers and Electronics in Agriculture, 2024; 227(Part 1): 109551.

[27] Li C, Lin J Q, Li B Y, Zhang S, Li J. Partition harvesting of a column-comb litchi harvester based on 3D clustering. Computers and Electronics in Agriculture, 2022; 197: 106975.

[28] Chen W B, Liu M C, Zhao C J, Li X X, Wang Y Q. MTD-YOLO: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. Computers and Electronics in Agriculture, 2024; 216: 108533.

[29] Chen J L, Chen H, Xu F, Lin M N, Zhang D, Zhang L B. Real-time detection of mature table grapes using ESP-YOLO network on embedded platforms. Biosystems Engineering, 2024; 246: 122–134.

[30] Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO, 2024. Available: https://github.com/ultralytics/ultralytics. Accessed on [2024-12-28].

[31] Wang C Y, Yeh I H, Liao H Y M. YOLOv9: Learning what you want to learn using programmable gradient information. In: Computer Vision – ECCV 2024, 2024; 15089: 1–21.

[32] Yang J N, Liu S L, Wu J J, Su X Y, Hai N, Huang X. Pinwheel-shaped convolution and scale-based dynamic loss for infrared small target detection. arXiv Preprint, 2024. doi: 10.48550/arXiv.2412.16986.

[33] Ioannou Y, Robertson D, Cipolla R, Criminisi A. Deep Roots: Improving CNN efficiency with hierarchical filter groups. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu: IEEE, 2016; pp.5977-5986. doi: 10.1109/CVPR.2017.633.

[34] Yu Z P, Huang H B, Chen W J, Su Y X, Liu Y H, Wang X Y. YOLO-FaceV2: A scale and occlusion aware face detector. Pattern Recognition, 2022; 155: 110714.

[35] Chen S T, Zhou F, Gao G, Ge X L, Wang R G. Unleashing breakthroughs in aluminum surface defect detection: Advancing precision with an optimized YOLOv8n model. Digit Signal Process, 2025; 160: 105029.

[36] Morales-Brotons D, Vogels T, Hendrikx H. Exponential moving average of weights in deep learning: Dynamics and benefits. arXiv Preprint, 2024. doi: 10.48550/arXiv.2411.18704.

[37] Wu J, Zhao F Y, Yao G T, Jin Z G. FGA-YOLO: A one-stage and high-precision detector designed for fine-grained aircraft recognition. Neurocomputing, 2025; 618: 129067.

[38] Yang Z Q, Xu K N, Zhao L B, Hu N, Wu J P. PWDE-YOLOv8n: An enhanced approach for surface corrosion detection in aircraft cabin sections. IEEE Transactions on Instrumentation and Measurement, 2025; 74: 1–22.

[39] Karmani S, Sivakaran T, Prasad G, Ali M, Yang W, Tang S. KPCA-CAM: Visual explainability of deep computer vision models using Kernel PCA. In: 2024 IEEE 26th International Workshop on Multimedia Signal Processing (MMSP), West Lafayette: IEEE, 2024; pp.1–5. doi: 10.1109/MMSP61759.2024.10743968.

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

[41] Yaseen M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. arXiv Preprint, 2024. doi: 10.48550/arXiv.2408.15857.

[42] Wang A, Chen H, Liu L H, Chen K, Lin Z J, Han J G, et al. YOLOv10: Real-time end-to-end object detection. arXiv Preprint, 2024. doi: 10.48550/arXiv.2405.14458.

[43] Xiong J T, Lin R, Liu Z, He Z L, Tang L Y, Yang Z G, et al. The recognition of litchi clusters and the calculation of picking point in a nocturnal natural environment. Biosystems Engineering, 2018; 166: 44–57.

[44] Li H P, Li C Y, Li G B, Chen L X. A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. Biosystems Engineering, 2021; 212: 347–359.

[45] Zhao R Z, Zhu Y C, Li Y H. An end-to-end lightweight model for grape and picking point simultaneous detection. Biosystems Engineering, 2022; 223(Part A): 174–188.