Lightweight Detection of Grape Inflorescences and Fruitlets using an Improved YOLOv8 Model*

Hu Guoyu, Lin Zhe, Wang Haining, Jiang Dexuan
(School of Mechanical Engineering (School of Intelligent Manufacturing and Modern Industry),
Xinjiang University, Urumqi Xinjiang 830017, China)

Abstract: Globally, grape cultivation spans vast areas and achieves substantial yields, making grapes and related industries vital economic pillars for many nations. In grape production, efficient and precise management during key growth stages is essential for enhancing both yield and quality. In view of the problems that during the grape inflorescences and young fruits stage, the targets are small in size, easily obscured by branches and leaves, and highly similar in color to the background, resulting in poor recognition performance of existing detection methods in complex natural environments, which in turn restricts the application of precision spraying technology. This paper establishes a dedicated dataset for grape inflorescences and young fruits in Xinjiang and proposes an improved lightweight detection model, YOLOv8-FCD. The model incorporates a PConv-based C2f_Faster module to reduce parameter count and computational complexity, replaces the original upsampling method with the CARAFE module to enhance feature extraction capability, and introduces the Detect_SEAM detection head to improve recognition accuracy under occlusion and small-target conditions. Experimental results show that the YOLOv8-FCD model achieves a detection precision (*P*) of 93.7% and a recall (*R*) of 87.3%, with a mean average precision (*mAP*) of 94.6%. Compared to the original YOLOv8n model, *P* improved by 8.2%, *mAP* increased by 2.6%, and the model size is reduced to 85.71% of the original. This model provides effective technical support for the identification of grape inflorescences and young fruits in intelligent spraying for plant protection.

Key words: image processing; deep learning; object detection; grape; YOLOv8

DOI: 10.13568/j.cnki.651094.651316.2025.07.21.0003

CLC Number: S232.3;S24 **Document Code**: A **Article ID**: 2096-7675(2026)02-0129-015

引文格式: 胡国玉,林哲,王海宁,江德轩. 基于改进YOLOv8模型的轻量化葡萄花穗及幼果检测模型[J]. 新疆大学学报(自然科学版中英文),2026,43(2):129-143.

英文引文格式: Hu Guoyu,Lin Zhe,Wang Haining,Jiang Dexuan. Lightweight detection of grape inflorescences and fruitlets using an improved YOLOv8 model[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English),2026,43(2):129-143.

基于改进YOLOv8模型的轻量化葡萄花穗及幼果检测模型

胡国玉, 林 哲, 王海宁, 江德轩
(新疆大学 机械工程学院(智能制造现代产业学院),新疆 乌鲁木齐 830017)

摘 要: 在全球范围内,葡萄种植面积广阔、产量丰富,葡萄及相关产业已成为许多国家重要的经济支柱. 在葡萄生产中,如何在其关键生长阶段实现高效精准的管理,对提升果实产量和品质至关重要. 针对葡萄花穗与幼果期目标尺寸

* **Received Date:** 2025-07-21; **Revised Date:** 2025-12-28; **Accepted Date:** 2025-12-30.

Foundation Item: The Major Science and Technology Special Project of Xinjiang Uygur Autonomous Region of China "Research, development, and integrated promotion of technologies for enhancing quality and efficiency across the entire industrial chain of Xinjiang honeydew melons" (2024A02007).

Biography: Hu Guoyu (1977—), female, professor, doctoral supervisor, research fields: research on agricultural robots and intelligent agricultural machinery equipment, E-mail: xjhuguoyu@xju.edu.cn.

小、易受枝叶遮挡、颜色与背景相似度高,致使现有检测方法在复杂自然环境下识别效果不佳,进而制约精准施药技术应用的问题,本文在新疆建立了葡萄花穗与幼果的专用数据集,并提出一种改进的轻量化检测模型YOLOv8-FCD. 该模型引入基于PConv的C2f_Faster模块以降低参数量与计算复杂度,将原始上采样方法替换为CARAFE模块,增强特征提取能力,并设计Detect_SEAM检测头,提升模型在遮挡与小目标场景下的识别精度. 实验结果表明,YOLOv8-FCD模型的检测精度(P)为93.7%,召回率(R)为87.3%,平均精度均值(mAP)达到94.6%. 与原始YOLOv8n模型相比,P提升8.2%,mAP提高2.6%,模型体积缩减至原来的85.71%. 该模型可为葡萄植保智能化喷雾中的花穗与幼果识别提供有效的技术支撑.

关键词:图像处理;深度学习;目标检测;葡萄;YOLOv8

0 Introduction

The Chinese grape industry has become intensive and large-scale, with the country ranking as the world's second-largest producer and leading producer of fresh table grapes[1]. Plant protection operations are a critical component in the grape cultivation process, significantly influencing grape growth and quality. The application of plant growth regulators during the inflorescence and juvenile stages is a crucial horticultural requirement in the cultivation and management process of fresh table grapes. It can effectively stimulate the production of large-grained, seedless grapes, thereby enhancing both the yield and quality of fresh table grapes, ultimately bolstering their economic viability[2–5].

While plant protection operations for grapes have largely achieved mechanization, issues such as severe pesticide waste, uneven deposition, and low efficiency in traditional application machinery are becoming increasingly prominent. The application of plant growth regulators during the inflorescence and juvenile stages typically depends on farmers using handheld backpack sprayers or electric sprayers. Inaccurate pesticide application strategy had a bad effect on grape quality and could not achieve pest control, greatly increasing labor costs and safety risks[6–7]. In modern agricultural production, intelligent robot-assisted plant protection operations offer significant advantages over traditional manual labor. The identification of grape inflorescences and fruitlets with a lightweight model can provide essential technical support and intelligent solutions for precise spraying of grape growth.

In recent years, target detection algorithms based on convolutional neural networks (CNN) have shown excellent performance[8–9]. The deep learning algorithms currently widely used for fruit and crop detection mainly fall into two categories. One is the two-stage algorithm based on candidate regions represented by region-based convolutional neural networks (R-CNN)[10], Faster R-CNN[11–12], which divides the detection process into two stages, focusing on the accuracy of target detection. The other is the one-stage algorithm based on regression represented by single shot multiBox detector (SSD)[13], you only look once (YOLO)[14–16], which converts the target detection task into an end-to-end regression problem, thus having better performance in real-time detection speed and can achieve rapid deployment on the edge side[17]. In order to detect tomatoes under the influence of various environmental factors, Gao et al.[18] proposed an improved model based on the original YOLOv5s. In their approach, they added a convolutional block attention module (CBAM) to the feature extraction network. Furthermore, they introduced an improved soft non-maximum suppression (Soft-NMS) method in the prediction part of the YOLOv5s network model. The overall performance of the improved model surpassed that of the original YOLOv5. Du et al.[19] proposed an improved Mask R-CNN model based on the ResNeXt network and integrated path enhancement to improve the detection and segmentation performance of fresh grape inflorescences and pedicels. They proposed a collection of logical algorithms to locate the gripping points of fresh grape inflorescences, effectively solving the problem of difficult detection of small targets such as fresh grape inflorescences and pedicels. Sun et al.[20] improved the YOLOv5s detection model for rapid and accurate identification of grape targets in complex orchard environments. They substituted the lightweight backbone network (MobileNetv3) and integrated the coordinate attention module (CA) into the backbone structure. Additionally, they incorporated the reparameterized VGG-style network

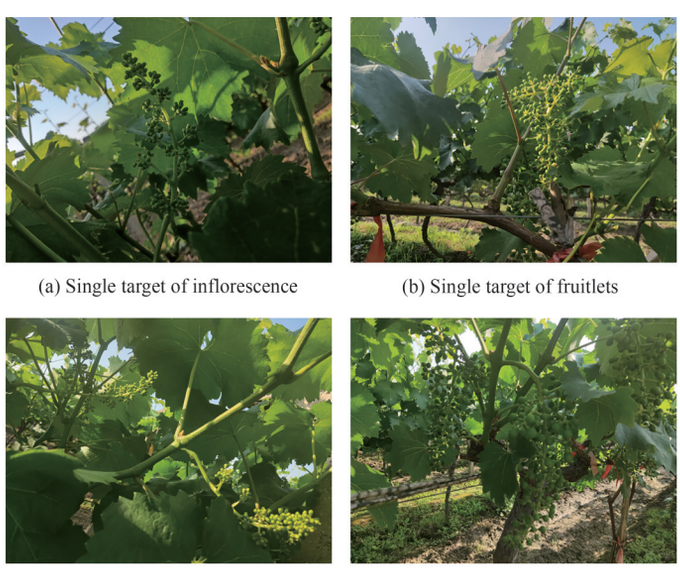
(RepVGG) block into the neck network to improve detection accuracy by merging multi-branch features. Moreover, they employed structural reparameterization of the RepVGG block to accelerate model inference speed, thereby enhancing both detection accuracy and speed for grapes. Wang et al.[21] improved the YOLOv8n detection model to enhance the recognition accuracy of camellia oleifera fruit under conditions of severe occlusion, near-scene color, and the coexistence of small targets. They employed MPDIOU as the loss function for YOLOv8n, incorporated a small target detection layer into the network, and utilized SCConv as the feature extraction network. Consequently, the improved COF-YOLOv8n network showed an increase in P, R, and mAP by 3.2%, 4.8%, and 2.4%, respectively, compared to the original YOLOv8n.

In the complex grape cultivation environment, the contours of grape inflorescences and fruitlets are irregular. Their colors are very similar to those of the surrounding leaves and branches, making detection challenging. Additionally, these targets are relatively small and exhibit significant differences in morphological characteristics and color compared to mature grapes. Existing methods are not suitable for detecting grape inflorescences and fruitlets. Therefore, to accurately and quickly detect grape inflorescences and fruitlets in a complex environment, this paper proposed an improved YOLOv8-FCD grape inflorescences and fruitlets detection model, which is based on the YOLOv8n algorithm. This provides a theoretical basis and technical support for the automation of grape plant protection.

1 Experimental Data

1.1 Image Acquisition

The image data for this paper was collected from a grape planting base in Huyi District, Xi'an City, Shaanxi Province, China, with geographical coordinates positioned at 33°46′-34°16′ N and 108°22′-108°46′ E. The data collection window extended from May 10, 2023, to May 24, 2023, with the shooting time ranging from 09:00 to 15:00. The shooting equipment was a Redmi K40 smartphone (Xiaomi M2012K11AC), with a collected image resolution of 4 000 × 3 000 pixels, horizontal and vertical resolution of 72 dpi, and an aperture value of f/1.79, with exposure time set to automatic. The images were taken under natural light, capturing various environmental conditions such as front light, backlight, close-up, long-distance, downward angle, and upward angle to improve data diversity and enhance the model's generalization ability. This approach yielded a total of 488 grape inflorescence images and 536 grape fruitlet images. Exemplary images are presented in Figure 1.



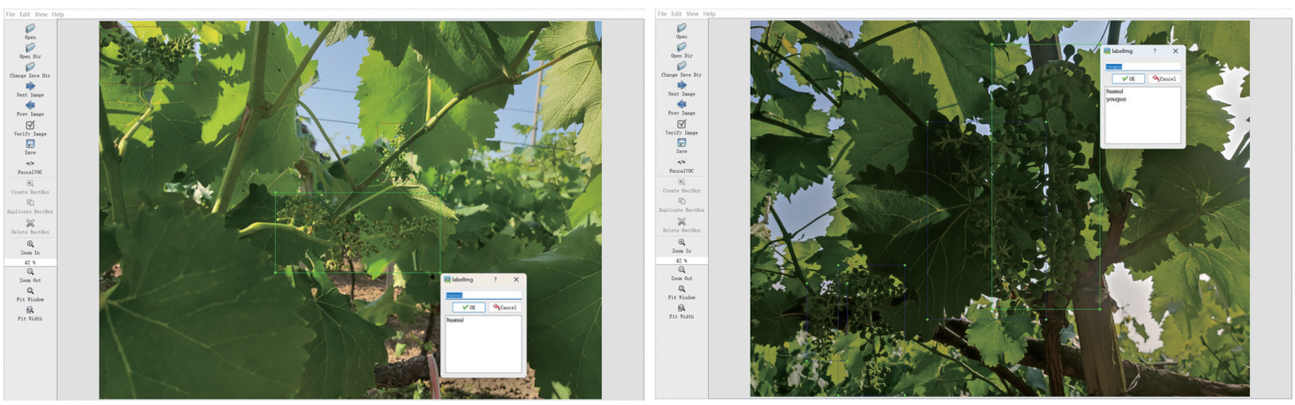
(a) Single target of inflorescence (b) Single target of fruitlets

(c) Multiple targets of inflorescence (d) Multiple targets of fruitlets

Figure 1 Examples of grape inflorescence and fruitlet collection images

1.2 Dataset Construction

This experiment involved researchers with expertise in grape morphology using the Labelimg tool to annotate rectangular bounding boxes around target regions in the original images, setting the label for grape inflorescence as "huasui" and the label for fruitlets as "youguo", and generating annotation files in the PASCAL VOC (pattern analysis, statistical modelling and computational learning visual object classes) format. The ground truth box, referred to as the marker box, is utilized to annotate defect information within the image. The label file records the coordinate information of the ground truth box. Figure 2 depicts an image instance from the dataset that has been labeled using Labelimg.



(a) Ground-truths of grape inflorescences (b) Ground-truths of grape fruitlets

Figure 2 Annotation diagram

In light of the complexity of the orchard environment, to enhance the robustness of the model and to avoid overfitting, Python scripts were used to perform data augmentation on the annotated images, including random rotation (±45°), mirror transformation, and adding Gaussian noise (mean 0, variance 0.01). Two new images were generated for each original image. After generating new images, the research team conducted a rapid visual inspection of all augmented samples to ensure that all images used for training were of high quality and effective. The final dataset amounted to a total of 3 072 images, including 1 464 grape inflorescence images and 1 608 grape fruitlet images. Finally, the entire dataset was partitioned into a training set and a validation set, and the VOC format XML files were transformed into YOLO format TXT files to serve as model input for training. The dataset distribution is shown in Table 1, and the expanded images are shown in Figure 3.

Table 1 Distribution of grape inflorescences and fruitlets dataset

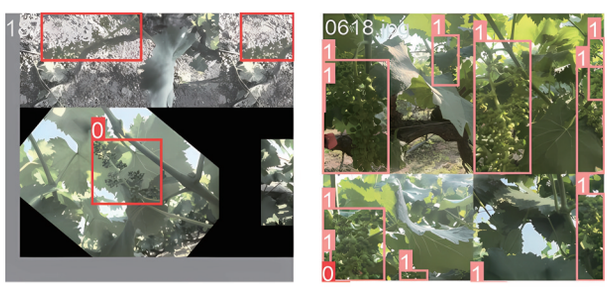
Category	Number of photos	Label	Number of labels	Total number of labels
Training set	2 764	huasui	2 796	7 573
		youguo	4 777	
Validation set	308	huasui	312	842
		youguo	530	

The data augmentation methods of YOLOv8 include mosaic, adaptive anchor box calculation, and adaptive target scaling. Mosaic data augmentation involves randomly cropping, flipping, scaling, and changing the color range of four images, and then stitching them together onto a single image as training data. The advantage of this approach is that it enriches the background of the images, and the stitching of the four images effectively increases the batch size. This method can effectively expand the dataset, improve the network's performance and robustness, and reduce memory consumption. As illustrated in Figure 4, the mosaic data augmentation effect is significant.

(a) Original image (b) Random rotation + Padding (c) Mirror + Noise

Figure 3 Data augmentation examples for grape inflorescences and grape fruitlets

Note: The first row consists of multiple data augmentation examples for grape inflorescences, while the second row consists of multiple data augmentation examples for grape fruitlets



(a) Grape inflorescences (b) Grape fruitlets

Figure 4 Mosaic data augmentation effects

2 Grape Inflorescences and Fruitlets Target Detection Method

2.1 Baseline YOLOv8 Network Model

The YOLOv8 is a state-of-the-art (SOTA) target detection algorithm launched by Ultralytics in 2023, building upon the historical versions of the YOLO series and introducing new features and improvements to further enhance performance and flexibility. YOLOv8, based on scaling factors, provides models of different scales such as n/s/m/l/x to meet the needs of various deployment platforms and application scenarios. In the backbone network, YOLOv8 has replaced the C3 module from the previous YOLOv5 model with the C2f module, achieving further lightweighting while continuing to use the SPPF module from YOLOv5, and fine-tuning for different scale models significantly improves model performance. In the neck network, the 1 × 1 convolutional layer has been removed. The detection head has been replaced with the current mainstream decoupled head structure, separating the regression branch and the classification branch, reducing the model complexity. YOLOv8 uses VFL Loss as the classification loss (BCE Loss is used in actual training), and DFL Loss + CIOU Loss as the regression loss, improving the model's detection accuracy and precision for targets.

The entire YOLOv8 model structure consists of four parts: The input end, the backbone network, the neck network, and the head network. Firstly, the input image undergoes preprocessing for the image data of the target to be detected at the input end. Then, it is input into the Darknet53 backbone network composed of convolutional layers, pooling layers, and residual connections to gradually extract features at different levels of abstraction. Secondly, the neck network integrates the feature maps extracted from the backbone network, deepening the network's hierarchical structure and increasing its nonlinearity to enhance the feature expression and generalization capabilities. Finally, the head network processes the final target detection task, and through multi-layer convolutional operations

and feature maps of different scales, it can better predict the location and category information of the target.

Since the YOLOv8n network is the smallest model in the YOLOv8 series and more easily meets the lightweight requirements, this paper chooses to make improvements based on this.

2.2 C2f_Faster Module

The C2f module utilized in YOLOv8 encompasses a greater number of bottleneck structures. While this facilitates the extraction of more features, it concomitantly results in an excessive redundancy of channel information. In the two-category detection task of grape inflorescences and fruitlets for this paper, the problem of redundant parameter volume and computational volume is more pronounced. Therefore, it is necessary to make lightweight improvements to the feature extraction network module of YOLOv8 to ensure that the detection model has high real-time performance on embedded devices.

In order to reduce the model's parameter volume and computational volume and optimize the feature extraction module, in this paper, we have introduced a C2f_Faster module based on PConv, and improved the detection model of this paper by replacing the bottleneck in C2f with the FasterBlock from FasterNet[22]. The PConv in FasterNet provides a more lightweight and efficient alternative, different from conventional convolution and depthwise convolution. The core idea of PConv is to apply conventional convolution on some channels while keeping the input unchanged on other channels. Its purpose is to reduce memory access and computational redundancy at the same time, which helps to reduce computational complexity and memory access, thereby improving the efficiency of CNN. Its working principle is shown in Figure 5.

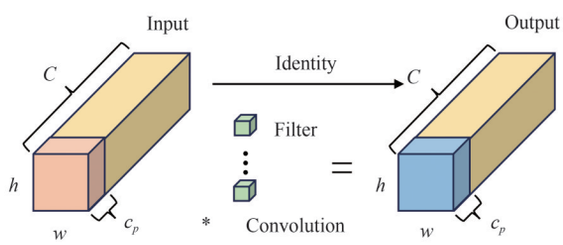


Figure 5 Structure of the PConv

The *FLOPs* equation (1), *MAC* equation (2), and *r* equation (3) for the PConv module are as follows:

$$FLOPs = h \times w \times k^2 \times c_p^2, \tag{1}$$

$$MAC = h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p, \tag{2}$$

$$r = \frac{c_p}{c}, \tag{3}$$

where h and w represent the length and width of the feature graph, respectively, k represents the size of the convolution kernel, c_p represents the number of channels participating in the convolution, and c represents the number of input channels.

Due to the fact that the number of channels involved in the convolution is only c_p, with $c - c_p$ channels not participating in the computation, and typically the rate of channels involved in the convolution is 1/4, the memory access of the PConv convolution module is only 1/4 of the conventional convolution, and the *FLOPs* are only 1/16 of the conventional convolution. Among them, c_p channels participate in the extraction of spatial feature information, while the remaining channels remain unchanged, ensuring that the subsequent feature channel information is not lost, while effectively reducing the model's computational volume and memory access. For continuous or regular memory access, the first or last continuous channel is regarded as a representative of the entire feature maps for calculation. It is assumed that the number of channels in the feature map input and output is the same without losing generality. In this study, we have employed the PConv to construct the FasterBlock, and by utilizing the mechanism of class inheritance, the bottleneck within the C2f module has been replaced with the FasterBlock from the

FasterNet (Figure 6). As the feature map passes through the FasterBlock, the PConv reduces the number of channels in similar feature maps, thereby decreasing the number of parameters involved in the computation, as well as the computational redundancy and memory access. The output feature map is then followed by two 1 × 1 convolutional kernels to ensure effective feature extraction. This design makes the C2f_Faster more lightweight during feature extraction, further improving the model's detection efficiency.

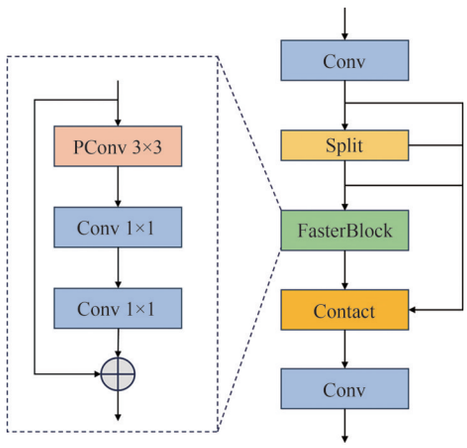


Figure 6 Structure of the C2f_Faster module

2.3 CARAFE Up-Sampling

In the actual grape garden environment, the large background noise hinders the model's ability to accurately distinguish useful target information during the up-sampling process. This results in poor feature map quality and an increase in useless interference information, which in turn affects the model's final detection accuracy. In the YOLOv8 target detection network model, the feature pyramid structure employs a nearest neighbor interpolation up-sampling method. This method only determines the up-sampling kernel based on the spatial position of the pixels, focusing solely on local features. It lacks a mechanism to fully utilize feature information and has a limited receptive field, failing to accurately reflect the image's global features. To enhance the expressive power of the output target feature information, this paper introduces a lightweight up-sampling operator, content-aware reassembly of features (CARAFE)[23], into the feature fusion network of the YOLOv8n model. This addresses the issue of traditional up-sampling operators ignoring semantic information in feature maps and their limited receptive field.

CARAFE is a lightweight up-sampling operator. Unlike deconvolutional up-sampling, which uses the same fixed convolutional kernel for up-sampling in the feature maps, CARAFE supports instance-specific content-aware processing and can dynamically generate adaptive kernels. Firstly, the input $H \times W \times C$ feature image is compressed to an $H \times W \times C_m$ feature map through a 1 × 1 convolution with the number of channels reduced to C_m, and then a content encoder with a convolutional kernel size of $k_{up} \times k_{up}$ is used to generate a reassembly kernel, resulting in a feature map of size $\sigma^2 \times k_{up}^2$. Secondly, through the pixel shuffle method, the height, width, and number of channels of the feature map are reorganized in turn to obtain an up-sampling kernel of size $\sigma H \times \sigma W \times k_{up}^2$, followed by softmax normalization processing. Finally, the feature image is input into the reassembly module, and the features on each layer of the feature map are multiplied by the predicted up-sampling kernel to obtain a feature map of size $\sigma H \times \sigma W \times C$. The structure of the CARAFE up-sampling network is shown in Figure 7.

CARAFE facilitates the reassembly of features within a predefined locale, each centered on a specific location, by employing weights that are generated with content-aware intelligence. For each locale, there are multiple sets of such weights dedicated to the up-sampling process, which are then rearranged to form a spatial block, thereby accomplishing the up-sampling of features. Consequently, the substitution of the conventional nearest neighbor interpolation up-sampling module in the YOLOv8n model with the CARAFE up-sampling technique en-

ables a heightened attentiveness to the distribution of features across the entire feature map. This approach augments the model's capacity to discern salient features during the up-sampling phase and bolsters the network's proficiency in feature extraction.

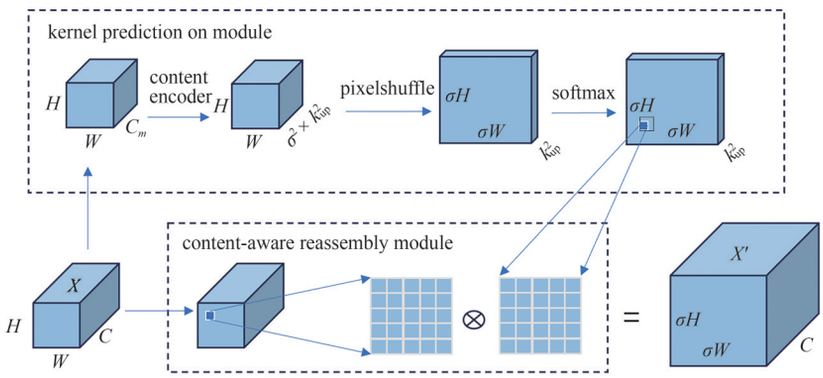


Figure 7 Structure of the CARAFE up-sampling network

2.4 Detect_SEAM Module

Grape inflorescences and fruitlets are relatively small compared to mature fruits, and there are situations where grape leaves, branches, and other foreign objects block the detection targets. The model detection is more difficult in complex and multi-occlusion orchard scenes, which affects the final model detection accuracy. Therefore, in this study, we combine the separated and enhancement attention module (SEAM)[24] with the detection head to redesign it into the Detect_SEAM detection module, as shown in Figure 8. The Detect_SEAM detection module can enhance the model's occlusion detection ability, achieve multi-scale target detection, and emphasize the detection target area in the image, while weakening the background area.

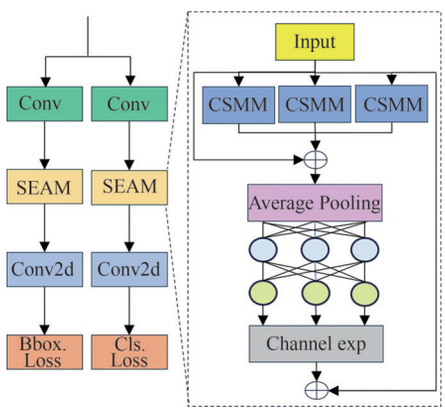


Figure 8 Structure of the Detect_SEAM module

Depthwise separable convolution is a channel-separated convolutional operation, capable of discerning the importance of individual channels and reducing parameter volume. However, this approach tends to overlook the relationships between inter-channel information. To compensate for the loss of inter-channel information, the outputs from different depthwise convolutions are combined through a 1×1 convolution, followed by the integration of channel-wise information through two fully connected layers, thereby strengthening the connections between channels within the network. The SEAM, an advanced form of depthwise separable convolution with residual connections, is capable of learning the relationship between occluded and non-occluded targets as inferred from the preceding stage, compensating for feature loss in occlusion scenarios. Subsequently, the output produced by the fully connected layer is processed by an exponential function, extending the value range from $[0,1]$ to $[1,e]$. Such exponential normalization provides a monotonic mapping, enabling the results to better avoid positional errors. Finally,

the output of the SEAM module is multiplied by an attention mechanism with the original features, enabling the model to more effectively process the feature information of occluded targets.

2.5 Improved YOLOv8-FCD Network Model

This study proposes an improved YOLOv8-FCD network model for the target detection of grape inflorescences and fruitlets in natural environments, based on YOLOv8n. The main improvements of this paper include: Introducing the C2f_Faster module based on PConv to reduce the model's parameter volume and computational volume. Utilizing CARAFE as a fusion network up-sampling module to enhance feature extraction capabilities. Introducing SEAM to redesign the detection head into the Detect_SEAM detection module to further improve detection accuracy. And finally introducing transfer learning to improve the model's fitting speed. The improved YOLOv8-FCD network model further enhances the recognition accuracy of grape inflorescences and fruitlets in natural environments while maintaining lightweight and fast detection speed. The structure of the improved YOLOv8-FCD is shown in Figure 9.

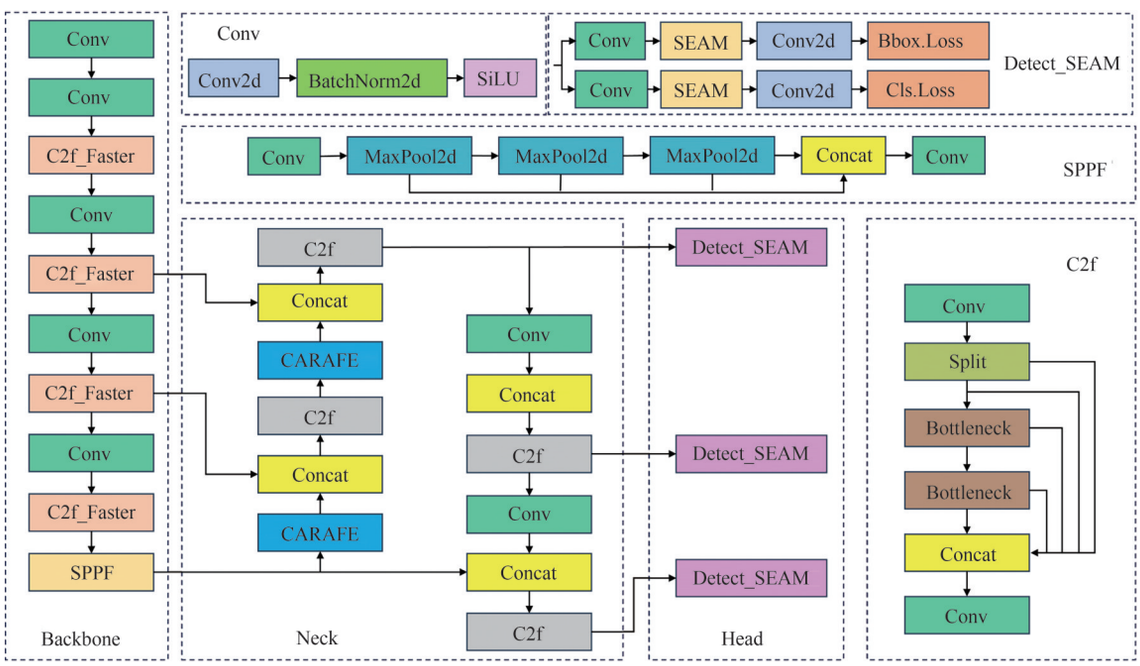


Figure 9 Structure of the improved YOLOv8-FCD network

2.6 Model Training

In this study, we use PyCharm to construct and improve the YOLOv8n network model. The hardware configuration of the platform used for experiments and training is a 13th Gen Intel(R) Core(TM) i9-13900HX processor at 2.20 GHz, an Nvidia GeForce RTX 4060 graphics card with 8 GB of video memory, running on a Windows 11 system, 64-bit, Python 3.9 version, PyTorch 1.12.0 version, and CUDA 11.3 version. The training parameters are shown in Table 2.

Table 2 Training parameters

Training parameter	Value
Initial learning rate	0.01
Optimizer	SGD
Momentum	0.937
Weight decay	0.000 5
Batch size	16
Epochs	100

3 Results and Analysis

3.1 Ablation Test

The results of the ablation study are shown in Table 3. It can be seen that after the introduction of the C2f_Faster module based on PConv to improve the detection model, the parameter volume and model size were reduced, and the model inference speed was increased. Replacing the up-sampling in the neck layer with CARAFE up-sampling improved the model's accuracy while reducing the parameter volume and computational load, making the detection model pay more attention to global feature information, more suitable for the recognition of grape inflorescences and fruitlets. After using the redesigned Detect_SEAM detection module to improve the model, the model's accuracy and detection capabilities were further enhanced, strengthening the model's occlusion-aware detection effects. By introducing transfer learning into the model, the weight parameters of the model trained on COCO128 data set can be transplanted into the new target learning model, which can improve the convergence speed and detection performance of the model.

Table 3 The results of the ablation test

Test No.	A	B	C	D	*P*/%	*R*/%	*mAP*/%	Weight/MB	*FPS*	*FLOPs*/G	Parameters
1	×	×	×	×	85.5	85.1	92.0	6.3	116.5	8.1	3 006 038
2	√	×	×	×	89.5	81.8	91.2	5.5	**129.8**	7.0	2 644 838
3	×	√	×	×	92.5	82.5	93.3	6.6	118.1	8.4	3 146 142
4	×	×	√	×	89.6	85.5	93.2	5.9	112.0	7.0	2 817 878
5	√	√	×	×	92.4	80.0	91.1	5.8	112.1	7.3	2 784 942
6	√	√	√	×	88.0	81.4	89.9	5.4	116.2	6.2	2 596 782
7	√	×	×	√	88.5	87.1	93.4	5.5	120.5	7.0	2 644 838
8	×	√	×	√	90.4	87.7	94.3	6.6	111.0	8.4	3 146 142
9	×	×	√	√	90.3	**88.4**	94.0	5.9	121.9	7.0	2 817 878
10	√	√	√	√	**93.7**	87.3	**94.6**	**5.4**	125.3	**6.2**	**2 596 782**

Note: All tests use YOLOv8n as the baseline network. A represents the C2f_Faster module, B represents the CARAFE module, C represents the Detect_SEAM module, D represents transfer learning, × means not using this module, √ means using this module

As shown in Table 3, Test 2, after replacing the original C2f module in the YOLOv8n network with the C2f_Faster module, the model weight was reduced to 87.30% of the baseline network, and the *P* increased by 4.0%, while *R* and *mAP* value decreased. The reason is that only a portion of the channels perform convolution operations. While reducing the model volume and computational redundancy, some features that the remaining channels may contain are lost, leading to a slight decrease in *R* and *mAP* value. In Test 3, after using CARAFE up-sampling, the *P* increased by 7.0%, and the *mAP* value increased by 1.3%. However, the *R* decreased by 2.6%. This indicated that CARAFE up-sampling could enhance the model's recognition accuracy. In Test 4, after replacing the model detection head with the redesigned Detect_SEAM, the *P* and *mAP* value increased by 4.1% and 1.2%, respectively. And the model weight was reduced to 93.65% of the baseline network. This indicated that the Detect_SEAM detection module can effectively improve the model's recognition accuracy and reduce the model volume, enhancing the model's detection capabilities under occluded scenes. In Test 5, after using C2f_Faster and CARAFE module, compared with Test 2, the *P* increased by 2.9%. The C2f_Faster module reduces computational redundancy through PConv, but inevitably sacrifices some channel information; CARAFE up-sampling relies on global contextual information. When combined, the loss of early-stage feature information may leave CARAFE with insufficient basis for feature reconstruction, particularly for grape clusters with complex backgrounds and small targets. This can result in the non-recall of some positive samples, leading to decreased recall rates. In Test 6, the model *R* increased by 1.4% on the basis of Test 5 by adding the Detect_SEAM detection module, and the model

weight and *FLOPs* value were reduced to 85.71% and 76.54% of the baseline network, respectively. Tests 7, 8, and 9 added C2f_Faster, CARAFE, and Detect_SEAM respectively on the basis of adding transfer learning, among which the *R* increased by 5.3%, 5.2%, and 2.9%, respectively, compared with Tests 2, 3, and 4, and the *mAP* value increased by 2.2%, 1.0%, and 0.8%, respectively. Combined with Test 10, which added transfer learning on the basis of Test 6, the *P*, *R*, and *mAP* value increased by 5.7%, 5.9%, and 4.7%, respectively. This indicated that transfer learning could achieve optimal training performance, improving indicators such as *R* and *mAP* without sacrificing the model's performance, and thus enhancing its detection accuracy and generalization ability. Compared with the baseline network, after improvement, the model weight was reduced to 85.71% of the baseline network, the model *P* increased by 8.2%, the *R* increased by 2.2%, and the *mAP* increased by 2.6%.

3.2 Comparative Experiment of Detection Models

To explore the superiority of the improved algorithm in this study, a comparative experiment was conducted between the improved YOLOv8-FCD model based on YOLOv8n and other mainstream object detection network models such as SSD, Faster R-CNN, and the YOLO series. We used the *AP* of 50% *IoU* for evaluating *AP*1 and *AP*2. *AP*1 represents the *AP* value of grape inflorescences, and *AP*2 represents the *AP* value of grape fruitlets. The results are shown in Table 4.

Table 4 Comparison of the performance of different detection models

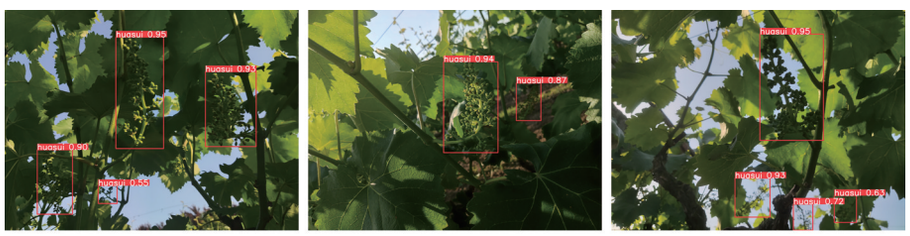
Model	*AP*1/%	*AP*2/%	*mAP*/%	*F*1/%	*R*/%	*P*/%	Weight/MB
SSD	64.5	80.2	72.4	69	55.0	84.1	91.1
Faster R-CNN	66.6	81.8	74.2	61	66.0	55.7	108.0
YOLOv3	42.5	72.7	57.6	39	27.2	92.4	235.0
YOLOv5s	89.9	90.3	89.9	81	71.7	91.4	27.1
YOLOX-nano	35.5	59.6	47.6	55	38.9	87.9	3.7
YOLOv7-tiny	81.1	83.1	82.1	65	49.7	92.1	23.1
YOLOv9c	**94.5**	94.3	94.4	89	86.7	92.0	102.8
YOLOv10n	86.4	92.0	89.2	84	77.9	90.3	5.8
YOLOv8-FCD	93.8	**95.3**	**94.6**	**90**	**87.3**	**93.7**	**5.4**

It can be seen from Table 4 that the YOLOv8-FCD model has a smaller model size compared to other models and outperforms them in terms of *P*, *R*, and *mAP*. Specifically, the *mAP* of YOLOv8-FCD is 22.20%, 20.40%, 37.00%, 4.70%, 47.00%, 12.50%, 0.20%, and 5.40% higher than that of SSD, Faster R-CNN, YOLOv3, YOLOv5s, YOLOX-nano, YOLOv7-tiny, YOLOv9c, and YOLOv10n, respectively. Concurrently, the model weight is reduced by 94.07%, 95.00%, 97.70%, 80.07%, 76.62%, 94.75%, and 6.90% compared to SSD, Faster R-CNN, YOLOv3, YOLOv5s, YOLOv7-tiny, YOLOv9c, and YOLOv10n, respectively. The two-stage network model, Faster R-CNN, exhibits a larger model size and lower recognition accuracy. The one-stage network models, such as SSD, YOLOv3, YOLOv5s, YOLOv7-tiny, YOLOv9c, and YOLOv10n, have a smaller model size and parameter volume than Faster R-CNN and achieve higher detection accuracy. The improved YOLOv8-FCD demonstrates superior performance in detecting grape inflorescences and fruitlets compared to other mainstream networks. The detection results are illustrated in Figure 10.

3.3 Model Feature Visualization

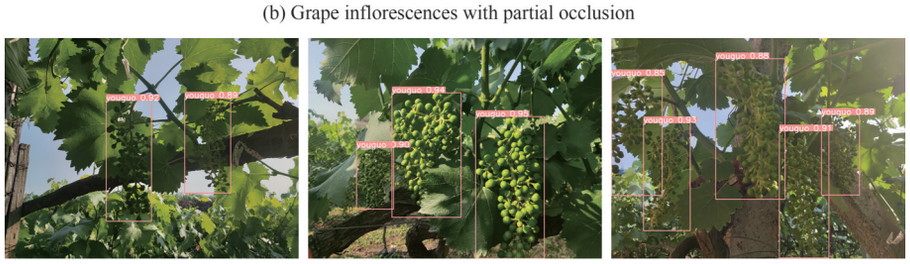
To more intuitively observe the improvement in the recognition capability of the improved model, Grad-CAM[25] is used to generate heatmaps, which can visually demonstrate the learning situation of the network for different targets. Grad-CAM utilizes the backward propagation of training weights, spatially averages the gradient matrix on a global scale, and after weighted activation of each channel of the feature layer, the heatmap is obtained.

The brightness of a certain area in the heatmap can show which parts have a greater impact on the model output. For the brightness depth of the specific heatmaps, we can see that the improved YOLOv8-FCD model pays more attention to the target characteristics. The model has better sensitivity to both trunk and fruit characteristics of the target. The heatmaps before and after the improvement of the detection model are shown in Figure 11.

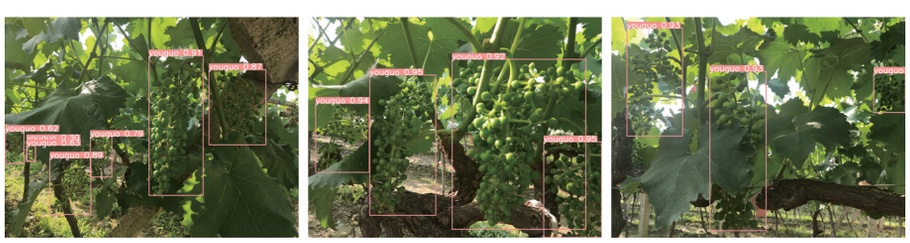


(a) Grape inflorescences without obstruction

(b) Grape inflorescences with partial occlusion



(c) Grape fruitlets without obstruction



(d) Grape fruitlets with partial occlusion

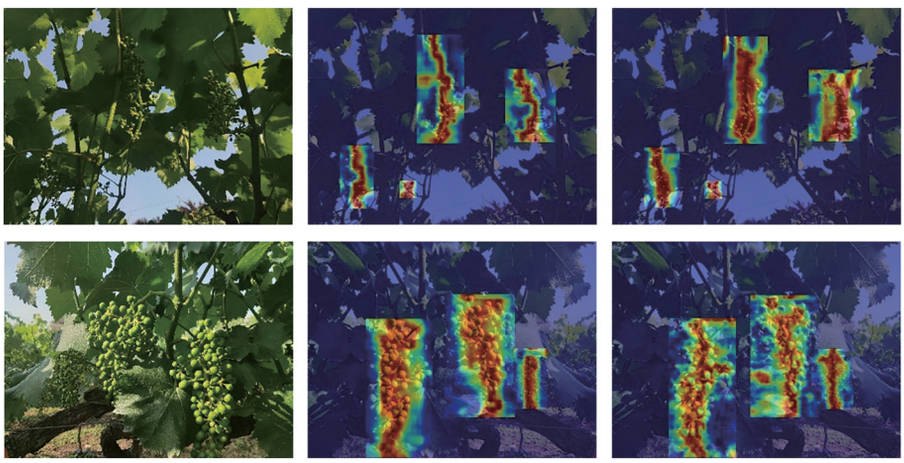
Figure 10 Detection effect diagram

3.4 Edge Device Deployment

In order to verify the actual detection effect and inference speed of the YOLOv8-FCD model, the model was deployed on the edge device and tested. Model migration deployment device uses Nvidia Jetson Nano, A57 quad-core ARM CPU, 128-core Maxwell GPU, running memory 4 GB, 64-bit LPDDR4. The software environment is Ubuntu 18.04, and the operating environment is configured with Jetpack 4.5, Python 3.6, Pytorch 1.8, and TensorRT 8.0.1.6.

At the same time, in order to improve the detection speed of the model, TensorRT inference library is selected for acceleration. TensorRT is a high-performance inference optimization framework from Nvidia that provides low-latency and high-throughput deployment inference acceleration for models on Nvidia GPU. The YOLOv8-FCD model training weight file was converted into WTS intermediate file and imported into Jetson Nano for compilation operation, and the model object was serialized to generate inference engine. Inference and post-processing opera-

tions can be performed by deserializing the engine file. Table 5 lists the test results of device deployment.



(a) Original images (b) YOLOv8 (c) YOLOv8-FCD

Figure 11 The heatmaps before and after the improvement of the detection model

Note: The three images in the first row correspond to grape inflorescences, and the three images in the second row correspond to grape fruitlets

Table 5 Comparison of device deployment detection frame rates (FPS)

Model	Desktop computers	Embedded devices	TensorRT
YOLOv8n	116.5	6.9	–
YOLOv8-FCD	125.3	8.1	45

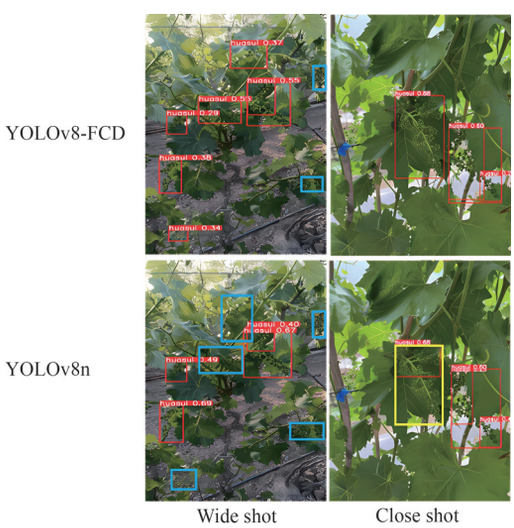


Figure 12 Comparison of detection effects of YOLOv8-FCD and YOLOv8n

Note: The red box is the prediction box, the yellow box is the false detection target, and the blue box is the missed detection target

It can be seen from Table 5 that before TensorRT acceleration, the detection speed of the improved YOLOv8-FCD model is relatively low due to the limited computing power of the embedded device. After acceleration, the model detection speed is increased by 5.56 times, the detection frame rate is 45 frames/s, and the detection speed is 21.26 ms. In order to further verify the detection performance of the YOLOv8-FCD model, the images of grape flower heads in backlight, cloudy days and blocked fields that are difficult to detect were selected for testing. The test results of YOLOv8-FCD and YOLOv8n models on Jetson Nano are shown in Figure 12.

According to Figure 12, in the wide shot, YOLOv8n missed 5 targets and YOLOv8-FCD missed 2 targets. In the close shot, YOLOv8n did not correctly detect 1 target, and YOLOv8-FCD prediction results were correct. The

detection ability of the improved model is better than that of the original model under both conditions.

4 Conclusion

1) In this study, a dataset of grape inflorescences and fruitlets under different lighting conditions was collected for the identification of grape plant protection robots. Additionally, this paper proposes a lightweight grape inflorescences and fruitlets target detection algorithm YOLOv8-FCD based on the improved YOLOv8n convolutional neural network.

2) The improved YOLOv8-FCD network model achieves a detection precision of 93.7% and the mean average precision of 94.6% for grape inflorescences and fruitlets, which are 8.2% and 2.6% higher than the original YOLOv8n model, respectively. The computational load is reduced to 6.2 G *FLOPs*, accounting for 76.5% of the original model, and the model size is reduced by 14.29%.

3) This paper's model further improves the detection accuracy of grape inflorescences and fruitlets while maintaining lightweight and fast detection speed, providing a theoretical basis and technical support for the automation of grape plant protection.

References:

[1] Tian Y, Chen G M, Li J F, et al. Current status of global grape industry development[J]. Tropical Agricultural Science, 2018, 38(6): 96-101+105.

[2] Crupi P, Alba V, Masi G, et al. Effect of two exogenous plant growth regulators on the color and quality parameters of seedless table grape berries[J]. Food Research International, 2019, 126: 108667.

[3] Rademacher W. Plant growth regulators: Backgrounds and uses in plant production[J]. Journal of Plant Growth Regulation, 2015, 34(4): 845-872.

[4] Han X Y, Mi Y F, Wang H L, et al. Influence of GA3 and CPPU on the quality attributes and peelability of "Wuhe Cuibao" grape[J]. Agronomy, 2025, 15(8): 1986.

[5] Liu Q, Peng J Q, Cheng J H, et al. Research progress on the application of plant growth regulators in grape production[J]. Heilongjiang Agricultural Science, 2019(8): 169-174.

[6] Wodzicki L M, Madden L V, Long E Y, et al. Evaluation of a laser-guided intelligent sprayer for disease and insect management on grapes[J]. American Journal of Enology and Viticulture, 2023, 74(2): 0740024.

[7] Bhalekar D G, Parray R A, Ingle P V, et al. Agrochemical spray technology adoption and safety awareness assessment in crop protection in vineyard cultivation[J]. Current Natural Sciences & Engineering Journal, 2024, 1(3): 188-196.

[8] Wang C L, Liu S C, Wang Y W, et al. Application of convolutional neural network-based detection methods in fresh fruit production: A comprehensive review[J]. Frontiers in Plant Science, 2022, 13: 868745.

[9] Sozzi M, Cantalamessa S, Cogato A, et al. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms[J]. Agronomy, 2022, 12(2): 319.

[10] Girshick R, Donahue J, Darrell T, et al. Region based convolutional networks for accurate object detection and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1): 142-158.

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

[12] Shahin S, Sadeghian R, Sareh S, et al. Faster R-CNN-based decision making in a novel adaptive dual-mode robotic anchoring system[C]//2021 IEEE International Conference on Robotics and Automation (ICRA 2021), May 30-June 5, 2021, Xi'an, Shaanxi, China. New York: IEEE, 2021: 11010-11016.

[13] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multiBox detector[C]//14th European Conference on Computer Vision (ECCV), October 8-16, 2016, Amsterdam, Netherlands. Cham: Springer, 2016: 21-37.

[14] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//International Conference on Computer Vision and Pattern Recognition (CVPR), June 26-July 1, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.

[15] Redmon J, Farhadi A. Yolov3: An incremental improvement[PP/OL]. arXiv(2018-04-18)[2025-01-01]. https://arxiv.org/abs/1804.02767.

[16] Zhu X, Lyu S, Wang X, et al. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021), October 11-17, 2021, Montreal, QC, Canada. New York: IEEE, 2021: 2778-2788.

[17] 董文轩, 梁宏涛, 刘国柱, 等. 深度卷积应用于目标检测算法综述[J]. 计算机科学与探讨, 2022, 16(5): 1025-1042.
Dong W X, Liang H T, Liu G Z, et al. Review of deep convolution applied to target detection algorithms[J]. Journal of Frontiers of Computer Science and Technology, 2022, 16(5): 1025-1042. (in Chinese)

[18] Gao G H, Shuai C Y, Wang S Y, et al. Using improved YOLOv5s to recognize tomatoes in a continuous working environment[J]. Signal, Image and Video Processing, 2024, 18(5): 4019-4028.

[19] 杜文圣, 王春颖, 朱衍俊, 等. 采用改进Mask R-CNN算法定位鲜食葡萄疏花夹持点[J]. 农业工程学院, 2022, 38(1): 169-177.
Du W S, Wang C Y, Zhu Y J, et al. Fruit stem clamping points location for table grape thinning using improved Mask R-CNN[J]. Transactions of the Chinese Society of Agricultural Engineering, 2022, 38(1): 169-177. (in Chinese)

[20] 孙俊, 吴兆祺, 贾忆琳, 等. 基于改进YOLOv5s的果园环境葡萄检测[J]. 农业工程学报, 2023, 39(18): 192-200.
Sun J, Wu Z Q, Jia Y L, et al. Detecting grape in an orchard using improved YOLOv5s[J]. Transactions of the Chinese Society of Agricultural Engineering, 2023, 39(18): 192-200. (in Chinese)

[21] 王金鹏, 何萌, 甄乾广, 等. 基于COF-YOLOv 8n的油茶果静、动态检测计数[J/OL]. 农业机械学报, 2024-01-17. https://link.cnki.net/urlid/11.1964.S.20240117.0910.002.
Wang J P, He M, Zhen Q G, et al. Camellia oleifera fruit static and dynamic detection counting based on improved COF-YOLOv 8n[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2024-01-17. https://link.cnki.net/urlid/11.1964.S.20240117.0910.002. (in Chinese)

[22] Chen J, Kao S H, He H, et al. Run, don't walk: Chasing higher FLOPS for faster neural networks[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023), June 18-22, 2023, Vancouver, COL, Canada. New York: IEEE, 2023: 12021-12031.

[23] Wang J Q, Chen K, Xu R, et al. CARAFE: Content-aware reassembly of features[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV 2019), October 27-November 2, 2019, Seoul, South Korea. New York: IEEE, 2019: 3007-3016.

[24] Yu Z, Huang H, Chen W, et al. Yolo-facev2: A scale and occlusion aware face detector[J]. Pattern Recognition, 2024, 155: 110714.

[25] Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization[C]//2017 IEEE/CVF International Conference on Computer Vision (ICCV 2017), October 22-29, Venice, Italy. New York: IEEE, 2017: 618-626.

责任编辑： 刘 敏