Grape detection in natural environment based on improved YOLOv8 network

Meng Junjie,[1] Cao Ziang,[1] Guo Dandan,[1] Wang Yuwei,[1] Zhang Dashan,[1] Liu Bingyou,[2] Hou Wenhui[1]

[1]Anhui Provincial Engineering Laboratory of Intelligent Agricultural Machinery, School of Engineering, Anhui Agriculture University, Hefei, Anhui; [2]Key Laboratory of Electric Drive and Control of Anhui Province, Anhui Polytechnic University, Wuhu, Ahhui, China

Abstract

In the pursuit of intelligent and efficient grape picking, rapid and precise detection of grape locations serves as the fundamental cornerstone. However, amidst the natural environment, grape detection encounters various interference factors, such as fluctuating light intensity, grape leaf obstructions, and grape overlap, all of which can undermine detection accuracy. To address these challenges, this study proposes a grape detection method leveraging an enhanced YOLOv8 network, wherein the conventional CIoU is replaced with Wise-IoU (WIoU) to augment network precision. Additionally, an efficient multi-scale attention module (EMA) is introduced to heighten the network's focus on grapes. To expedite detection, the original network backbone is substituted with the CloFormer_xxs network. The collected grape images undergo preprocessing to ensure image quality, forming the basis of the dataset. Furthermore, the dataset is augmented using Disadvantages-Enhance (DE), a novel data enhancement mode, thereby enhancing the robustness and generalization of network. The comprehensive comparison and ablation experiments are conducted to demonstrate the advantageous effects of the proposed modules on the network. Subsequently, the improved network's superiority in grape detection is validated through comparative analyses with other networks, showcasing superior accuracy and faster detection speeds. The network achieves a remarkable accuracy of 92.1%, average accuracy of 94.7%, with preprocessing and post-processing times of 15ms and 0.8ms, respectively. Consequently, the enhanced network presented in this study offers a viable solution for facilitating intelligent and efficient grape picking operations.

Correspondence: Hou Wenhui, Anhui Provincial Engineering Laboratory of Intelligent Agricultural Machinery, School of Engineering, Anhui Agriculture University, Hefei, Anhui, 230036 China. E-mail: hwh303@ahau.edu.cn

Key words: automatic grape picking; disadvantages-enhance; EMA; grape detection; smart agriculture; YOLOv8.

Funding acknowledgement statement: this work was funded by University Natural Science Research Project of Anhui Province (2022AH050872), Opening Project of Key Laboratory of Electric Drive and Control of Anhui Province, Anhui Polytechnic University (no. DQKJ202203) and Anhui Province science and technology major project (202103a06020015).

Received: 7 April 2024.
Accepted: 13 July 2024

Journal of Agricultural Engineering 2024; LV:1594
doi:10.4081/jae.2024.1594

Publisher's note: all claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

Introduction

China became the world's largest producer of grapes as early as 2010 and has continued to maintain its leading position (Tian *et al.*, 2017). As woody vines belonging to the Vitaceae family, grapes require artificial supports to maintain their vines. Currently, manual harvesting remains the primary method for grape harvesting. With the expansion of grape planting scale, the continuously rising labor costs have become the main constraint on grape production. Therefore, machine-based grape harvesting automation has become a crucial necessity (Sheridan *et al.*, 2016; Wouter *et al.*, 2014; Van *et al.*, 2010). Accurate and rapid grape location recognition is an indispensable prerequisite for achieving automated grape harvesting. However, during the harvest season, grapes are embedded in interwoven branches and swollen grape leaves, making visual detection of grapes complex.

Since the 1970s, more and more scholars have devoted themselves to the research of object detection technology and achieved some results (Abdullahi *et al.*, 2017), which greatly promoted the development of agriculture, transportation industry and human-computer interaction (Ji *et al.*, 2023; Wang *et al.*, 2018).With the development of computer technology and image processing, target detection technology for fruit detection has mainly undergone two stages: traditional machine learning-based and deep learning-based approaches.

Traditional machine learning-based object detection techniques were extensively utilized in the early stages of fruit detection, relying on feature extraction and pattern recognition. Commonly used features encompass color, shape, and texture features (Zhuang *et al.*, 2018). Rabby *et al.* (2018) extracted shape and color features of oranges and apples, enhancing the edge detection algorithm for their detection. Pothen *et al.* (2016) proposed a fruit recognition method incorporating an improved directional gradient histogram and texture descriptors, employing a random forest classifier for classification, achieving accuracy rates of 82% and 80% in grape and apple detection, respectively. Zeeshan *et al.* (2020) created a feature space by extracting color, shape, and texture features, combining them with a support vector classifier to achieve fruit classification with a detection accuracy of 87.06%. Patel *et al*. (2011) calculated brightness, color, direction, edge features, performed weighted integration and segmentation operations, resulting in binary images and extracted regional fruit positions with an accuracy rate of 90%.

Castro *et al*. (2019) evaluated various machine learning algorithms combined with three color spaces (RGB, HSV, and Lab), demonstrating that the SVM classifier in the Lab color space achieved the highest accuracy at 93.02%. Despite improving detection accuracy, these methods often suffer from slow network detection speed and poor robustness in recognizing fruits in complex scenes. Fu *et al*. (2019) tested bananas in a natural environment based on texture and color features, achieving an average detection time of only 1.325 seconds. However, the method's accuracy was low, unsuitable for real detection in natural environments. Lu *et al*. (2015) proposed a method combining color and contour information to identify citrus fruits, successfully identifying them in orchards. Nevertheless, this method exhibited numerous missed and incorrect detections in fruit recognition under complex backgrounds. The techniques described previously rely on extracting color, shape, and texture features, which restrict their ability to identify fruits with pronounced appearance differences and pose limitations on recognizing specific fruits. During detection, these methods encounter challenges such as illumination variations, occlusions, overlaps, among others, complicating feature extraction and reducing accuracy. In contrast, recent years have seen a shift towards deep learning-based object detection techniques, which have increasingly supplanted traditional machine learning approaches for fruit detection. Deep learning-based fruit detection networks are categorized into two types: those that are based on candidate regions and those that utilize regression for target detection (Hubel *et al*., 1962). These advanced methods offer enhanced precision and adaptability in identifying a broader range of fruits under varying conditions. Popular target detection networks based on candidate regions include R-CNN, Fast R-CNN, Faster R-CNN, Mask-RCNN, among others. These networks operate in two stages: first, numerous candidate boxes are generated through convolution operations on feature maps, followed by classification and position regression of these candidate boxes. Consequently, they are referred to as two-stage networks. Girshick *et al*. (2014) introduced R-CNN, an object detection algorithm relying on candidate boxes, in 2014. However, this approach involves extracting features from each object candidate box in every image, leading to substantial computational space requirements during training.

Subsequently, Girshick *et al*. (2015) enhanced R-CNN, presenting Fast R-CNN in 2015, utilizing the VGG16 neural network. This modification reduced the original 2000 CNN operations to just one, accelerating training by nine times and test speed by 213 times. Ren *et al*. (2015) further refined this concept with Faster R-CNN, substituting an RPN network for the SS algorithm to generate candidate boxes, thereby shortening both training and detection times while enhancing accuracy. He *et al*. (2017) introduced the Mask-RCNN network, leveraging the ROI Align and bilinear interpolation methods to improve network accuracy by obtaining pixel values of floating-point coordinates. Le *et al*. (2019) applied the improved Mask R-CNN network to banana detection, achieving an average accuracy of 92.5%, which increased to 93.8% with dataset expansion. Gao *et al*. (2020) proposed a multi-class apple detection method for dense leaf fruit trees based on a fast regional convolutional neural network, addressing issues of branch and leaf occlusion, thereby enhancing network detection accuracy in complex environments. While these methods boast high accuracy, their two-stage network structure contributes to slower detection speeds.

The deep learning-based single-stage object detection networks, such as SSD and YOLO series algorithms, have gained prominence for their efficiency and reduced computation. In 2015, Redmon *et al*. (2016) introduced YOLOv1, treating object detection as a regression problem and utilizing a single neural network to output bounding boxes directly. Wei *et al*. (Wei *et al*., 2016) proposed SSD in 2016, employing small convolution checks and a 3×3 convolution kernel to predict category scores and box offsets, thereby reducing detection time. Subsequent versions like YOLOv2 by Redmon *et al*. (2017) and YOLOv3 by Redmon *et al*. (2018) improved backbone networks and introduced mechanisms like anchor frames and feature pyramid networks to enhance performance. Afterwards, Tian *et al*. (2019) introduced an improved YOLOv3-dense model for detecting apples across different growth stages, demonstrating enhanced performance under occlusion conditions. Bochkovskiy *et al*. (2020) proposed YOLOv4 in 2020, incorporating data enhancement and multi-scale fusion to improve network learning ability. Jocher *et al*. (2020) proposed YOLOv5 in the same year, utilizing structures like Focus and CSP to reduce parameter computation and memory usage. In subsequent years, improvements continued with the introduction of YOLOv7 by Wang *et al*. (2023), focusing on model reparameterization and auxiliary head training methods to increase accuracy. Ji *et al*. (2021) proposed an apple detection method based on Shufflenetv2-YOLOX, achieving improved performance in detection speed and AP value. Jocher *et al*. (2023) proposed YOLOv8, which further enhanced object detection by refining network architecture.

While object detection techniques have made some progress, deep learning based fruit detection models still face challenges in grape detection in natural environments. In the natural environment, grape fruit bodies are densely distributed in clusters, the fruit bodies are compact and adhering, and the branches and leaves are occluded or mutually occluded. As a result, there is a large number of false detections, missed detections, and re-detections in grape detection in natural environments, which poses certain difficulties and challenges for accurate grape fruit detection. To address these issues, an improved YOLOv8 grape detection model is proposed with the aim of improving accuracy and speed in grape detection scenarios. The following is a summary of the major contributions of this study:

- To enhance the network's robustness and generalization, we propose a novel data augmentation method, DE, aimed at diversifying the dataset.
- In order to enhance detection accuracy, a new architecture for grape detection in the complex natural environment is provided based on YOLOv8-GRAPE network, which use WIoU avoiding false detections, missed detections, and re-detections. Meanwhile, the EMA module is integrated to heighten the network's focus on grapes within intricate environments.
- In pursuit of improved detection speed, we integrate the CloFormer_xxs network to substitute the traditional backbone network within the YOLOv8 model.

The paper proceeds as follows: In the second section, we detail the image acquisition, processing, and data augmentation processes, along with the introduction of a novel data augmentation method. Moving to the third section, we present both the conventional YOLOv8 network and the modified version, elucidating the framework of each network structure. The fourth section entails training the network using the dataset. Additionally, to validate the superiority of our enhanced model, we train other networks with the same dataset, with results outlined in the following section. In the seventh section, we conduct a comparative experiment to further validate the superiority of our model, culminating in the presentation of final conclusions.

Persimmon dataset

Image collection

First, we collected images of grapes in their natural environment in a vineyard, and then filtered the images with low pixels and no grapes. The following is a detailed description of the data collection:

- Equipment location and collection: we used an Intel RealSense d455 camera to collect images of grapes in the natural environment at Dawei Farmer's Vineyard in Baohe District, Hefei City, Anhui Province. The data acquisition device is shown in Figure 1.
- Collection environment: we collected data sets of grapes under the conditions of backlight, direction light, strong light intensity and weak light intensity, respectively. At the same time, we also collected images of grapes under leaf occlusion, non-occlusion, overlap between grapes, and non-overlap between grapes to ensure the diversity of the dataset and prevent the network from overfitting.
- Image processing: firstly, low-pixel and low-quality images without grapes in the original image were filtered, and 500 high-quality images were obtained. The processed images were shown in Figure 2, and then Labelimg was used to manually label the obtained high-quality images and name grape.

Dataset enhancement

To prevent overfitting in the training process due to too few datasets and to improve the generalization ability of the model, 500 high-quality images were augmented. We propose a novel data augmentation method, DE, to improve the image fidelity, whose flowchart is shown in Figure 3.

Specifically, the input image is first transformed into a grayscale image to reduce the space occupied in the process of enhancement, then significant features and non-significant features are extracted through threshold segmentation, significant features are retained and non-significant features are sharpened using convolution check for image sharpening, and finally significant features and non-significant features are merged. We call this data enhancement method Disadvantages-Enhance. In addition, we also use five traditional methods, namely flipping, rotating 90°, color dithering, color enhancement and sharpening to enhance the image. Specifically, we flip the image left and right, rotate 90° clockwise, and set different random factor value intervals to adjust the image saturation, brightness, contrast, and sharpness to achieve the effect of color dithering. An image containing only transparency information is converted to an image containing both gray and transparency for color enhancement, and regions in the image with small relative differences in pixel values are reduced and regions with large relative differences in pixel values are increased for image sharpening. With data augmentation, the dataset was increased to 1500 images. Then, based on the Pycharm platform, the dataset partitioning code is used to randomly split the dataset, with 80% as training set, 10% as validation set, and 10% as test set. The flowchart of data augmentation is shown in Figure 4.

Network model

Improved network

Based on the traditional YOLOv8 network, we make several improvements that improve the performance of the network during grape detection and shorten the training and detection time. The structure of the improved network is shown in Figure 5.

Optimizing the loss function

Each sample is passed through the model and a prediction

YOLOv8-SIoU	0.858	0.907	2.2	1.7
YOLOv9	0.897	0.905	2.6	2.3
YOLOv7	0.915	0.899	2.8	2.5
YOLOv5	0.884	0.902	3.2	2.9
YOLO-v3	0.916	0.890	2.7	3.4

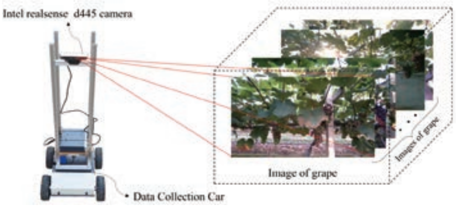


Figure 1. Data acquisition device.

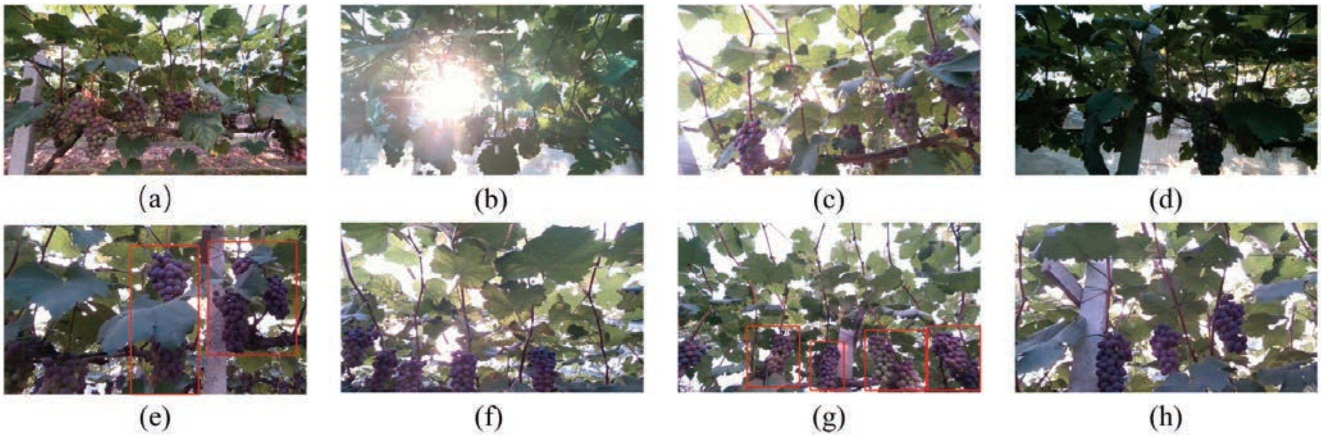


Figure 2. a) Backlight. b) Phototropism. c) Light intensity. d) Low light. e) Occlusion; f) No occlusion. g) Overlap. h) No overlap.

value is obtained. The difference between the predicted value and the true value is called the loss. The smaller the loss, the better the accuracy of the model. A loss function is essentially a function of the type used to compute the difference between the predicted value and the true value. YOLOv8 has only two branches at the output of the network: the classification branch uses VFL, and the regression branch uses CIoU and DFL (distributed focal loss). CIoU is a loss calculation function for boundary prediction. It first receives the predicted value from the forward propagation of the model, then computes the difference between the predicted value and the true value, and finally provides data for the backward propagation of the model. The formula for the calculation can be stated as follows:

$$Loss_{CIoU} = R + d^2\left(b, b^{gt}\right) / c^2 + \alpha v \tag{1}$$

$$R = 1 - IoU \tag{2}$$

Where b is the central point of the prediction box, b^{gt} is the central point of the real box, d is the euler distance between the two central points, and IoU is the intersection ratio, α is the weight function and v measures the consistency of the aspect ratio. The formula for calculating α and v is as follows:

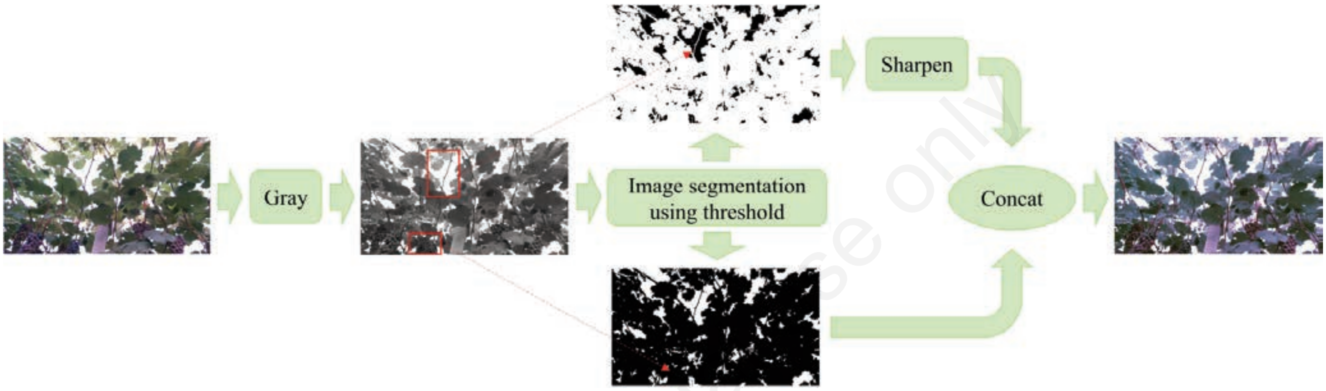


Figure 3. Schematic diagram of the Disadvantages-Enhance system.

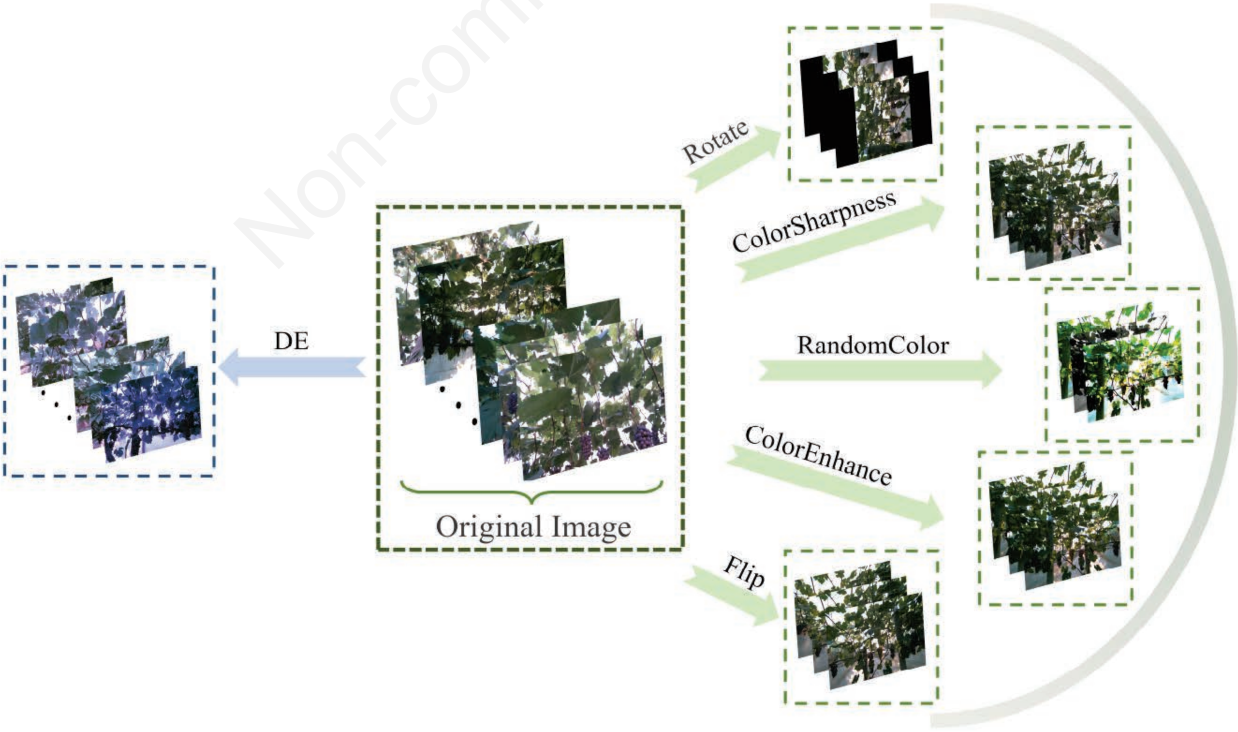


Figure 4. Schematic diagram of the data augmentation.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \quad (3)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (4)$$

where, w is the width of the prediction box, h is the height of the prediction box, w^{gt} is the width of the real box, and h^{gt} is the height of the real box.

Based on DIoU, CIoU introduces aspect ratio consideration to address the problem that DIoU fails to distinguish certain cases in Bounding Box Regression. However, the aspect ratio difference reflected by ν in CIoU formula is not the real difference between the width and height respectively and their confidence, so sometimes it will hinder the effective optimization similarity of the model. The traditional loss function only considers the overlap of the true and predicted boxes, and does not consider the region between the two, which may cause the network to be biased in evaluating the results. During dataset production, the quality of image annotations is not uniform due to many factors, but traditional loss functions increase the penalty for low-quality samples, which reduces the generalization ability of the model. Therefore, we optimize the traditional YOLOv8 network to use WIoU v2 (Tong *et al*., 2023) instead of CIoU in traditional YOLOv8. WIoU is a dynamic rather than monotonic FM loss function that combines Wise with an IO-based loss. The loss function uses outliers instead of IoU to evaluate sample quality and has an intelligent gradient gain allocation strategy. This strategy not only reduces the competitiveness of high-quality samples, but also reduces the penalty for low-quality samples, so that the loss function can focus on normal-quality samples and improve the accuracy and generalization ability of the network. Effective learning under limited conditions is key for real-time detection, and WIoU improves the overall performance of the network by balancing learning on low-quality samples with high-quality samples. The calculation formula of WIoU v2 can be expressed as:

$$R_{WIoU} = e^{\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}} \quad (5)$$

$$L_{WIoUv2} = \left(\frac{L_{IoU}^*}{\overline{L}_{IoU}}\right)^{\gamma} R_{WIoU} L_{IoU} \quad (6)$$

where R_{WIoU} is the penalty term of WIoU_loss, and is the coordinate of the upper-left corner of the anchor box, x_{gt} and y_{gt} is the coordinate of the upper-left corner of the target box, W_g and H_g represents the width and height of the minimum bounding box, L_{WIoUv2} is the WIoU_loss v2 function, L_{IoU} represents the IoU loss function, L^* IoU represents the monotonic focusing coefficient of L_{IoU}, and L_{IoU} represents the mean of L_{IoU}.

Introduction of efficient multi-scale attention module

High-precision grape detection networks are an important guarantee for automatic grape picking. In real-time monitoring of grapes in a natural environment, the collected grape detection images contain not only grape information but also a large amount of interference information. Therefore, being able to focus on important feature information and suppress other useless information is crucial for grape detection networks in the natural state. Attention mechanisms have flexible structural properties that can be easily ported to network architectures, while enhancing learning of critical parts. Motivated by this, we introduce an attention mechanism in the traditional YOLOv8 model to increase the atten-

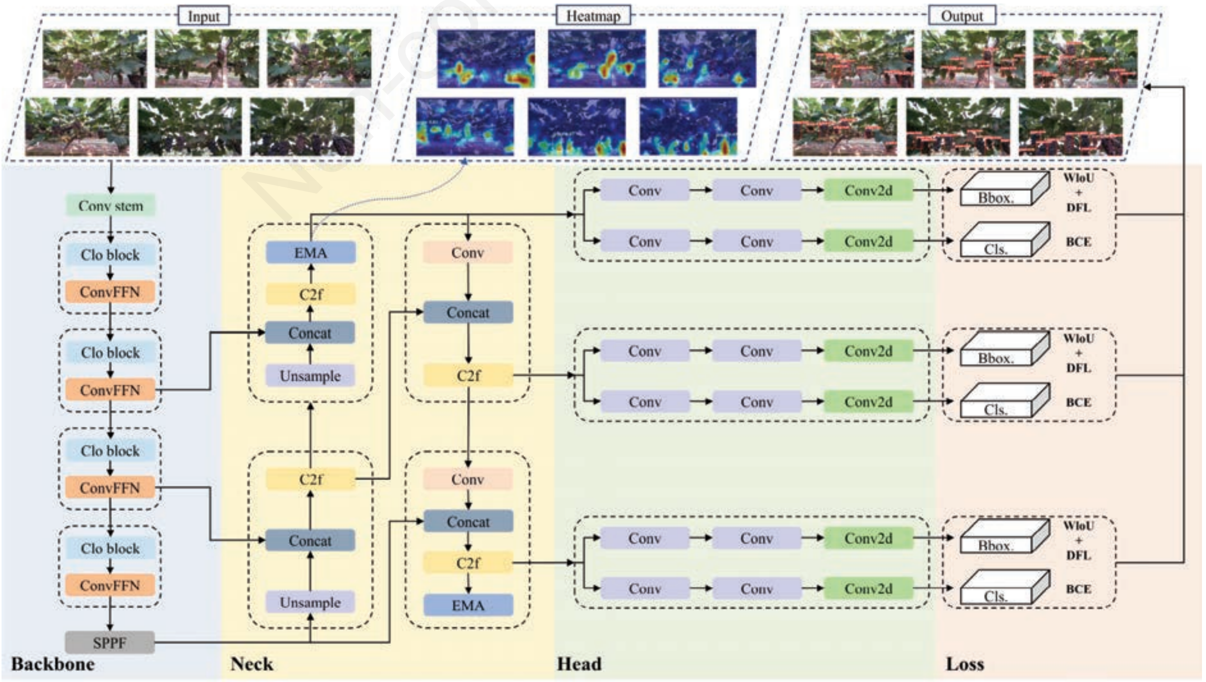


Figure 5. Structure diagram of the improved network.

tion of the network on key parts. In principle, there are three main types of attention mechanisms: spatial attention mechanism, channel attention mechanism, and hybrid spatial and channel attention mechanism. Most attention mechanisms use channel dimensionality reduction to model cross-channel relations, which may introduce some side effects during feature extraction. Therefore, we introduce an efficient multi-scale attention module (EMA) (Ouyang *et al*., 2023), which adopts a general approach to reshape part of the channel into batch dimensions and divide the channel dimensions into multiple sub-features so that the spatial semantic features are evenly distributed within each feature group, thus avoiding channel dimensionality reduction by general convolution. The above approach not only preserves the information on each channel, but also reduces the computational overhead. The structure of EMA is shown in Figure 6. From Figure 6, we can see that EMA adopts three paths for image feature extraction, the first two paths are named as 1×1 branches, and the third path is named as 3×3 branches. On two 1×1 branches, there are two global mean pool layers that encode channels in the space in both directions, respectively. On the 3×3 branch, there is a 3×3 kernel for extracting multi-scale features. Two 1×1 branches are connected in parallel with a 3×3 branch. This structure can aggregate multi-scale spatial structure information while achieving fast response to the entire structure for better performance.

Lightweight treatment

Fast and accurate grape detection networks are essential guarantees for real-time grape detection in natural environments. Traditional object detection networks tend to be computationally intensive and have long detection times, which will have a huge impact on real-time grape detection. Based on the traditional

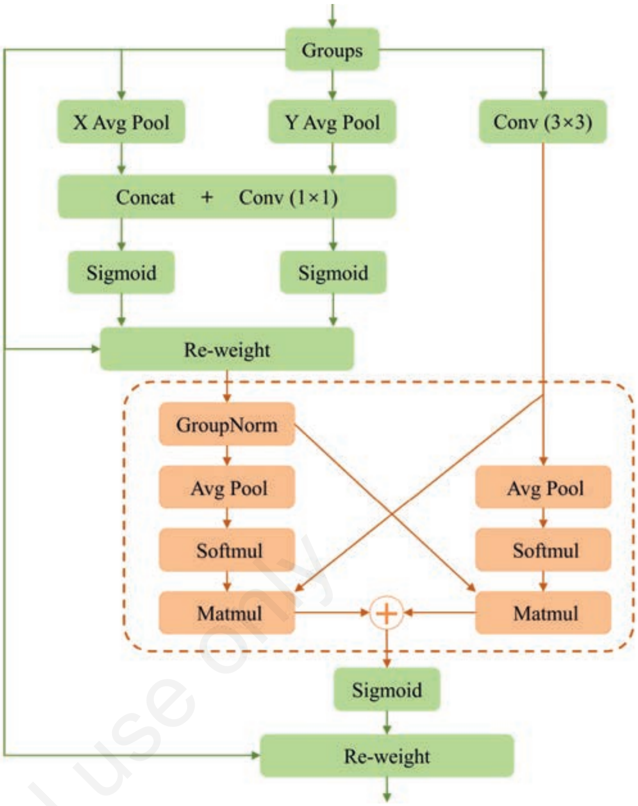


Figure 6. Structure diagram of the efficient multi-scale attention module (EMA).

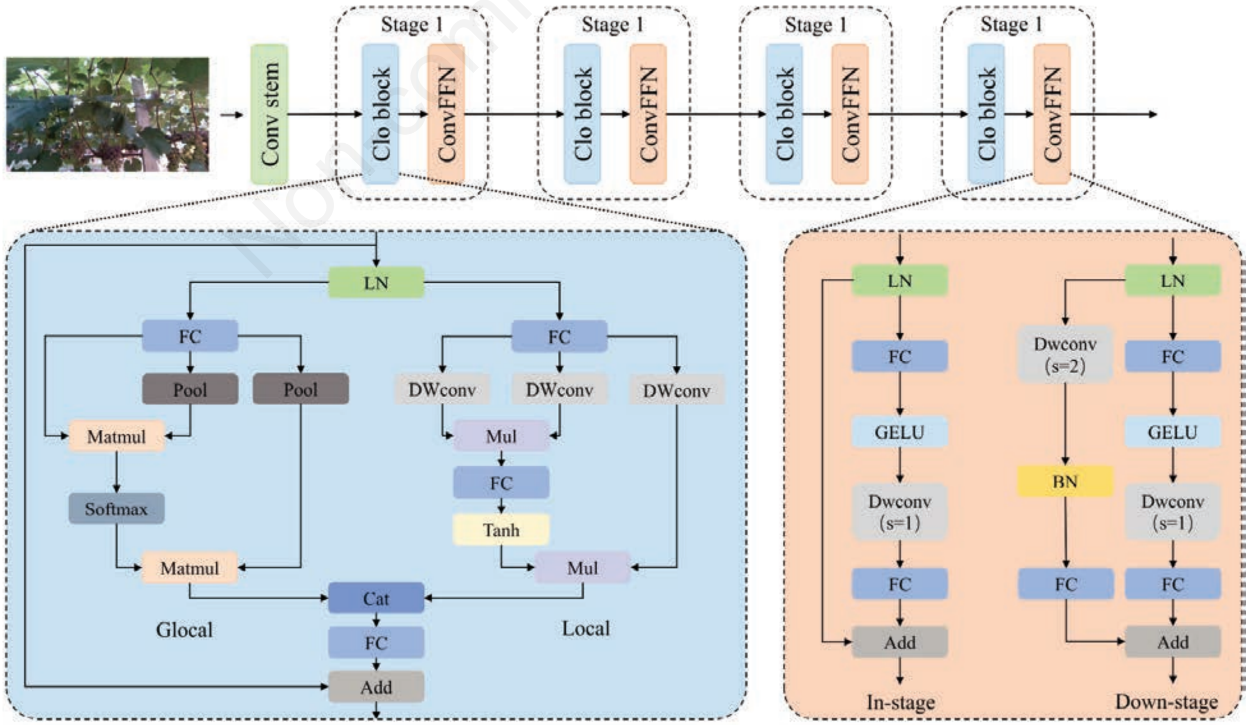


Figure 7. Structure diagram of the CloFormer_xxs.

YOLOv8 model, we replace the backbone part of the network with CloFormer_xxs network (Fan *et al*., 2023) to reduce the computation and detection time of the network. CloFormer_xxs is a lightweight and efficient visual backbone network whose structure is shown in Figure 7. The CloFormer_xxs network consists of a convolutional system Conv stem and four stages, each containing a Clo block module and a ConvFFN module. Specifically, the input image is fed into the convolutional system to obtain image features, indicating that the system consists of four convolution steps of 2, 2, 1 and 1 in sequence. The obtained image features are then passed through four Clo blocks and ConvFFN modules to extract hierarchical features, where the Clo block module consists of a global branch and a local branch. Global branches are used to extract low-frequency global information, effectively reducing the number of floating-point operations required by the attention mechanism. The local branch is used to extract high-frequency global information and AttnConv is used to process high-frequency local information to achieve fusion of shared and context-aware weights for better processing of high-frequency local information. The ConvFFN module can aggregate local information by incorporating it into the FFN. There are two types of ConvFFN modules used in the CloFormer_xxs network, the first one is the intra-phase ConvFFN, which directly uses skip joins, and the other one is the inter-phase ConvFFN, which is mainly used for downsampling and augmentation operations.

Training operation

To some extent, the training environment will have some impact on the training duration and results. The experimental environment is divided into hardware environment and software environment. The specific details of the training environment are given in Table 1.

Training process and Results

Precision (P), recall (R) and mean average precision 50 (mAP50) were used to describe the performance of the model. Where P represents the proportion of positive samples that the model can correctly predict. R represents the proportion of all true positive samples that the model can find. mAP50 represents the average accuracy across multiple classes at an IoU threshold of 0.5. We import the above data set into the optimized model for training and save the training results in "train". The hyperparameters during training are as follows: imgsz is 640 pixels ×640 pixels, batch is 16, epoch is 300, workers is 0, device is 0, patience is 50, learning rate is 0.01, etc. We import the test set into the trained model, make the model detect the images in the test set and save the detection results in the file "detect". The results are shown in Table 2.

Experiment

Comparative experiment based on WIoU

Based on the fact that there are three versions of WIoU, namely WIoU v1, WIoU v2, and WIoU v3, we conduct a comparison experiment on the three versions of WIoU to obtain the optimal loss that is suitable for grape detection. Based on the traditional YOLOv8 model, we replaced each of the three versions of WIoU in the traditional model with CIoU, and then trained the model using the aforementioned datasets. The training results are shown in Table 3. The replacement models are named YOLOv8-WIoU v1, YOLOv8-WIoU v2, and YOLOv8-WIoU v3. As can be seen from Table 3, both P and mAP50 of YOLOv8-WIoU v2 are higher than those of the other two models, thus YOLOv8-WIoU v2 has the best detection.

We then further analyze the training results. We compare the various losses of the models in Table 3, and the results are shown in Figure 8. Box_loss is used to calculate the gap between the predicted boundary box and the real boundary box, DFL_loss is used to represent the gap between the distance field predicted by the model and the real distance field, and Cls_loss is used to calculate the gap between the predicted category and the real category. It can be seen from the Figure 8, that all kinds of losses of YOLOv8-WIoU v2 are lower than those of the other two versions, indicating that this model can accurately locate the position of grapes and correctly identify grapes, and this model can also accurately describe the shape of the boundary box. To further verify the superiority of WIoU v2, we compare YOLOv8-WIoU v2 with a network that replaces the other loss functions. Based on the traditional YOLOv8

Table 1. Training environment.

Environment	Name	Configuration
Hardware environment	CPU	Intel (R) Core™ i9-12900K
	GPU	NVIDIA GeForce RTX 3090
	ROM	24GB
	RAM	32GB
Software environment	System	Window 10
	CUDA	12.1
	Pytorch	2.1.0
	Python	3.8.5

Table 2. Indicators.

	P	R	mAP50
Training set	0.947	0.820	0.921
Validation set	0.926	0.914	0.937
Test set	0.941	0.879	0.916

Table 3. Training results of YOLOv8-WIoU v1, YOLOv8-WIoU v2, YOLOv8-WIoU v3.

	P	mAP50
YOLOv8-WIoU v1	0.908	0.913
YOLOv8-WIoU v2	0.933	0.915
YOLOv8-WIoU v3	0.916	0.902

Table 4. Training metrics for networks with various loss functions are introduced.

	P	mAP50
YOLOv8-WIoU v2	0.933	0.915
YOLOv8-CIoU	0.828	0.827
YOLOv8-DIoU	0.886	0.900
YOLOv8-MPDIoU	0.903	0.904
YOLOv8-EIoU	0.880	0.905
YOLOv8-GIoU	0.908	0.912
YOLOv8-SIoU	0.858	0.907

model, SIoU, DIoU, MPDIoU, EIoU, and GIoU are replaced by CIoU, and the replaced models are named YOLOv8-SIoU, YOLOv8-DIoU, YOLOv8-MPdiou, YOLOv8-IoU, and YOLOv8-GIoU, respectively. We still train them with the above dataset, and the training results are shown in Table 4. We can see that YOLOv8-WIoU v2 achieves the best results and both P and mAP50 are higher than the networks with the other introduced loss functions. To verify the effect of the network in the instance detection in Table 4, we performed instance detection on the model in the above Table and the detection effect is shown in Figure 9. We can see that YOLOv8-WIoUv2 does not have re-detections, false detections, and missed detections in the natural environment of grape detection. In contrast, the other models have more re-detections, false detections, and missed detections, which proves that reducing the penalty of low-quality samples during network training helps improve the accuracy of the network. It is further shown that WIoU v2 has certain advantages in the grape detection process in natural environments. Where the yellow arrows point, there are missed checks, error checks, and re-checks.

Comparative experiment based on EMA

In the complex environment of grape growing, the attention of the network to the grapes is crucial. In order to increase the network's attention to grapes, we introduce the EMA module into the YOLOv8 model, which uses a multi-scale parallel subnetwork to establish short and long dependencies, and uses the general method to reshape part of the channel batch dimension and divide the channel dimension into multiple sub-features. Spatial semantic features are uniformly distributed within each feature group, thus avoiding information loss and allowing the network to focus more on the features of the grape itself. To verify the superiority of EMA, we embed EMA, CBAM, GAM, SA, SE, ECA, AIFI and CA attention mechanisms based on the traditional YOLOv8 model. They are named YOLOv8-EMA, YOLOv8-CBAM, YOLOv8-GAM, YOLOv8-SA, YOLOv8-SE, YOLOv8-ECA, YOLOv8-AIFI and YOLOv8-CA, respectively. The same datasets are used to train the models described above, and the training results are shown in Table 5. As it can be seen, networks with EMA have some performance benefits, with higher P, R, and mAP50 than networks with other attention mechanisms. To better demonstrate the advantage of EMA in making the network pay more attention to grape compared to other attention mechanisms, we performed a heatmap visualization of the seventh layer of the above model using Grad-CAM separately, and the results are shown in Figure 10, where we can see that the introduction of EMA enables the network to reduce the focus on non-critical parts and increase the focus on grape. It is worth noting that while the network focuses more on the grape, its confidence also improves.

Comparative experiment based on CloFormer_xxs

To evaluate the superiority of CloFormer_xxs network in terms of detection speed, 100, 150, 200, 250, 300, 350, 400, 450 and 500

Table 5. Training metrics of various attention mechanism networks are introduced.

	P	R	mAP50
YOLOv8-EMA	0.905	0.778	0.868
YOLOv8-CBAM	0.854	0.756	0.836
YOLOv8-GAM	0.787	0.774	0.813
YOLOv8-SA	0.902	0.715	0.816
YOLOv8-SE	0.859	0.756	0.809
YOLOv8-ECA	0.875	0.775	0.863
YOLOv8-AIFI	0.898	0.733	0.836
YOLOv8-CA	0.875	0.762	0.836

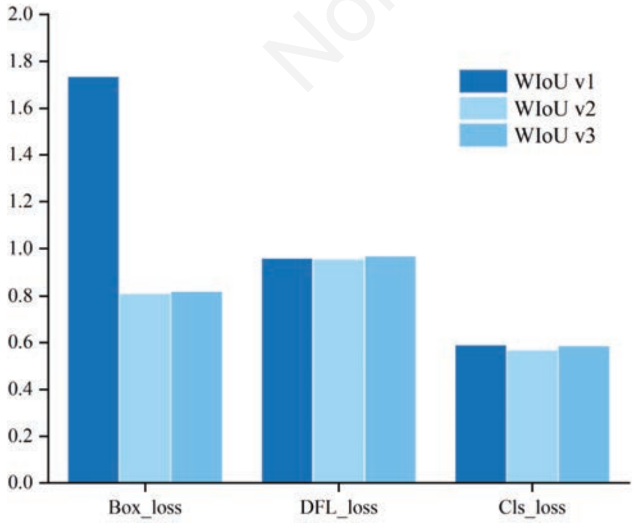


Figure 8. Box_loss, DFL_loss, Cls_loss for each version of WIoU.

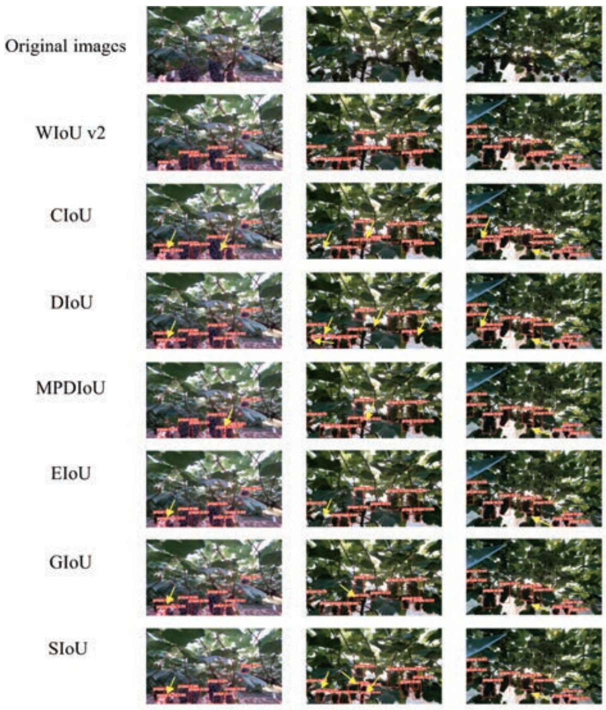


Figure 9. Detection plot of the network with each loss function introduced.

images were detected by the conventional YOLOv8 model and the model introduced with CloFormer_xxs network, respectively. We will name the model that introduces the CloFormer_xxs network YOLOv8-CloFormer_xxs. The test results are shown in Figure 11. The results show that the YOLOv8-CloFormer_xxs model outperforms the traditional models in terms of image pre-processing and post-processing during case detection.

Ablation experiment

To verify the effect of the traditional YOLOv8 model with EMA, WIoU v2 and CloFormer_xxs networks, we conducted ablation experiments. We will name the network that introduces EMA and WIoU v2 simultaneously YOLOv8-WIoU v2+EMA, and name the network that introduces EMA and CloFormer_xxs simultaneously YOLOv8-EMA+CloFormer_xxs, The network that introduces both WIoU v2 and CloFormer_xxs is named Yolov8-Wiou v2+CloFormer_xxs, and the network that introduces both WIoU v2, EMA, and CloFormer_xxs is named YOLOv8-GRAPE. The training results are shown in Table 6.

Through ablation experiments, it is proved that the optimal detection effect can be obtained when WIoU v2, EMA, and CloFormer_xxs are introduced into the network at the same time.

Comparative experiments with other networks

To further verify that the improved network has some superiority in grape detection in natural environments. We train YOLOv8-GRAPE with other detection networks using the above dataset, and the comparison results are shown in Table 7, where it can be seen that YOLOv8-GRAPE has the best performance, which again proves that the improved model proposed in this paper has better detection accuracy and speed in grape detection in natural environments.

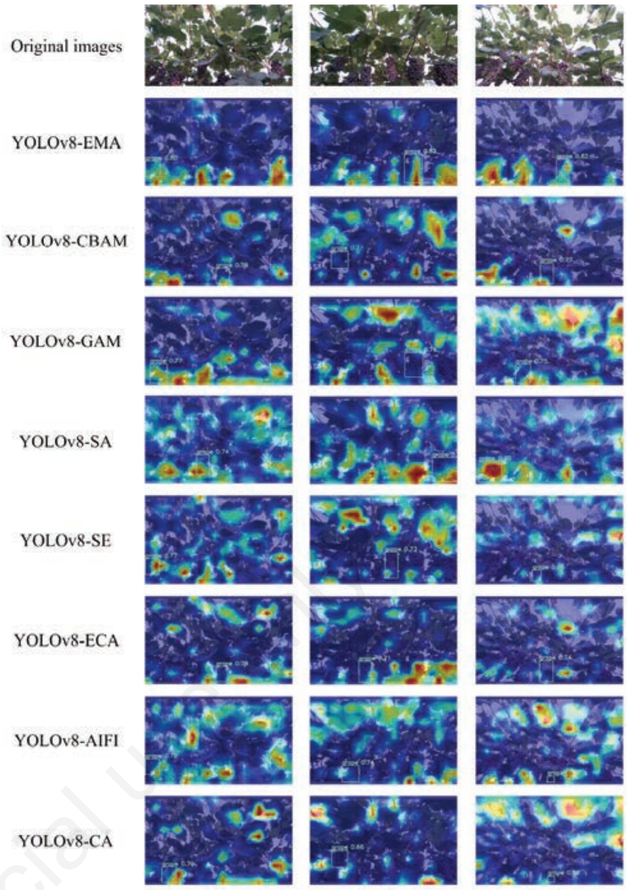


Figure 10. Visual heat map comparison.

Table 6. Ablation experiment.

Component	Choice							
WIoU v2	√	√		√	√			
EMA	√	√	√			√		
Cloformer_xxs	√		√	√			√	
P	0.947	0.921	0.897	0.805	0.933	0.905	0.880	0.828
R	0.820	0.703	0.762	0.767	0.850	0.778	0.769	0.782
mAP50	0.921	0.826	0.836	0.831	0.915	0.868	0.860	0.827

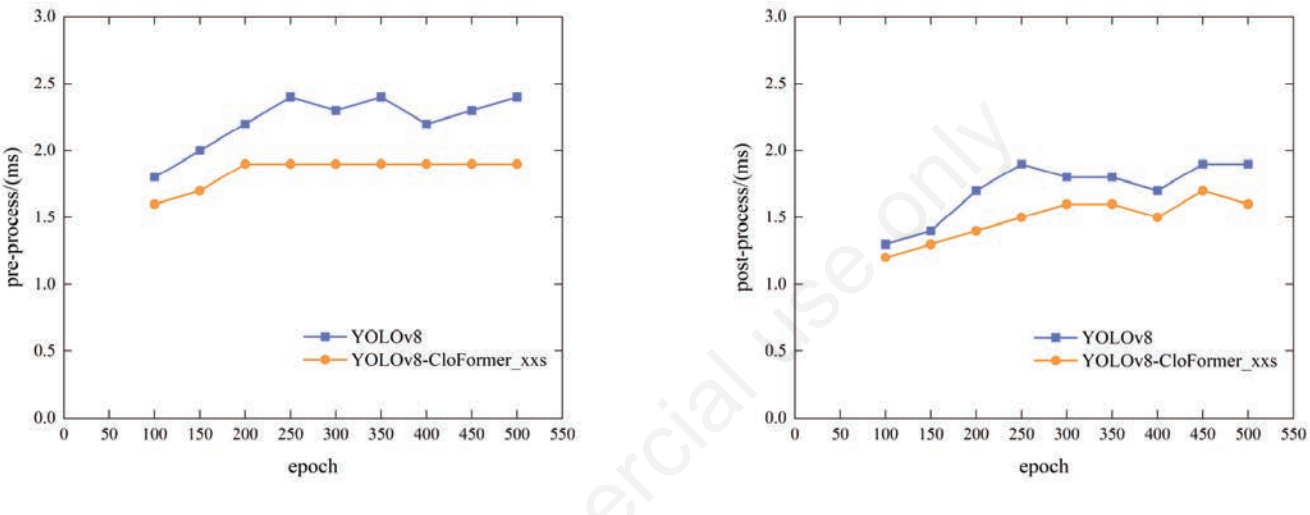
Table 7. Comparison of various networks.

	P	mAP50	Pre-procession/(ms)	Post-procession/(ms)
YOLOv8-GRAPE	0.947	0.921	1.5	0.8
YOLOv8	0.828	0.827	2.4	1.9
YOLOv8- CBAM	0.854	0.836	2.2	2.5
YOLOv8-SA	0.902	0.816	2.5	2.1
YOLOv8-EIoU	0.880	0.905	1.9	1.7
YOLOv8-SIoU	0.858	0.907	2.2	1.7
YOLOv9	0.897	0.905	2.6	2.3
YOLOv7	0.915	0.899	2.8	2.5
YOLOv5	0.884	0.902	3.2	2.9
YOLO-v3	0.916	0.890	2.7	3.4

Case detection

To show the detection effect of the improved network in natural environments, we split the above test set according to the conditions of backlight, toward light, bright, dark, occlusion, no occlusion, overlap and no overlap. Among them, there are 64 images with backlight, 56 with toward light, 72 with bright, and 75 with dark. There are 77 images with occlusion, 61 images with no occlusion, 83 images with overlap, and 73 images without no overlap. The above images are fed into the network for detection, where the ratio of the number of successful identifications to the total number of samples is the identification rate, and the results are shown in Figure 12, where it can be seen that the recognition rate of the network for grapes can reach more than 92% under the above conditions, and the specific grape detection results under different conditions are shown in Figure 13.

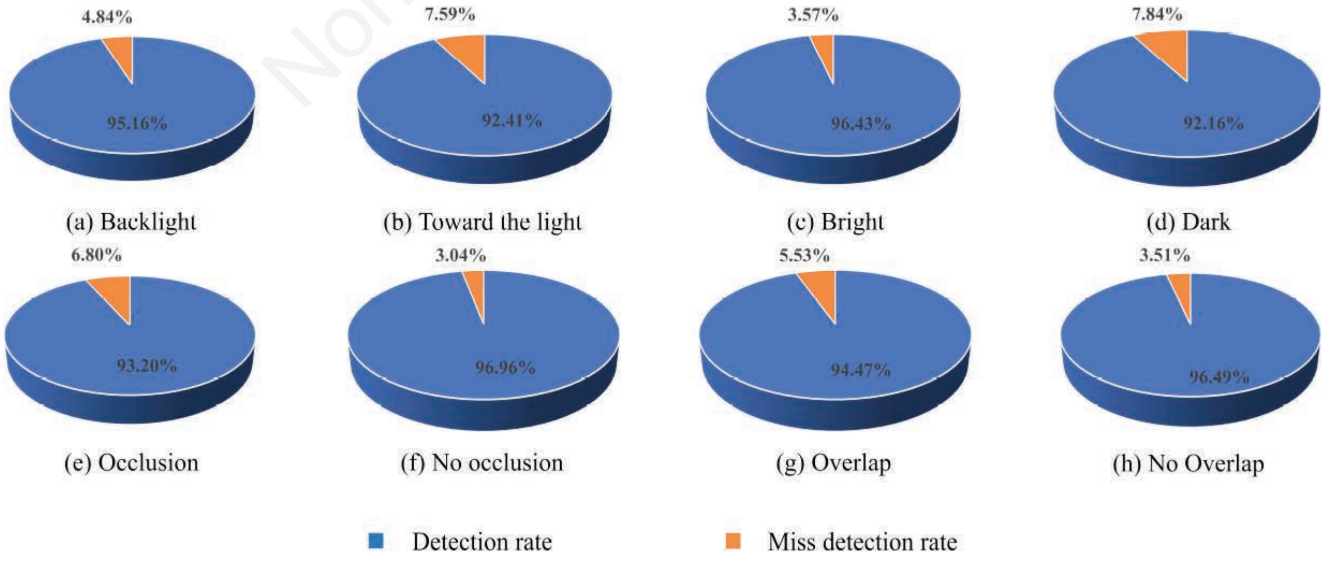
The improved network does not suffer from false detections, missed detections and re-detections in the above environments, and the detection accuracy is high, which proves that the network still performs well in complex environments (Figure 13).



(a) prepocession

(b) postpocession

Figure 11. Comparison of image pre-processing and post-processing.



(a) Backlight

(b) Toward the light

(c) Bright

(d) Dark

(e) Occlusion

(f) No occlusion

(g) Overlap

(h) No Overlap

Figure 12. Recognition rates of grapes under various conditions.

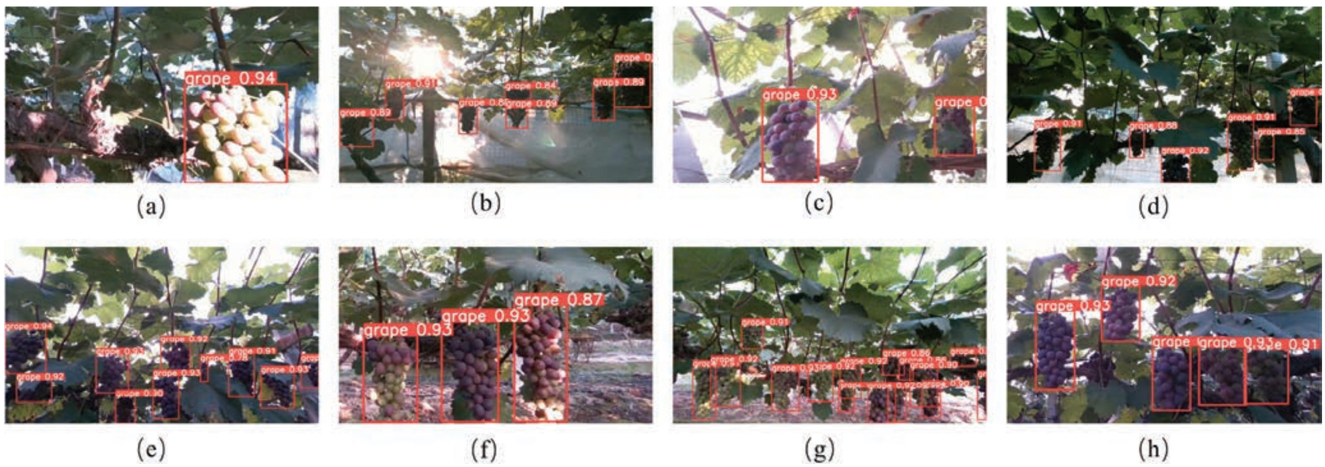


Figure 13. **a**) Backlight. **b**) Phototropism. **c**) Light intensity. **d**) Low light. **e**) Occlusion. **f**) No occlusion. B) Overlap. **h**) No overlap.

Conclusions

This paper presents a method for grape detection in natural environments utilizing an improved YOLOv8 network. Leveraging DE and five conventional data augmentation techniques, we augment the original dataset to bolster network robustness and mitigate overfitting. Integration of WIoU enhances detection accuracy within the YOLOv8 framework, while the introduction of EMA focuses the network's attention on grapes. Furthermore, we implement lightweight processing and integrate the CloFormer_xxs network with the original one to accelerate pre-processing and post-processing speeds.

To assess the impact of the enhanced modules on model performance, we conducted a series of experiments. Firstly, regarding network accuracy, we evaluated three versions of WIoU, finding that WIoU v2 significantly enhances network performance. Subsequently, comparing WIoU v2 with other loss functions, we observed that the YOLOv8-WIoU v2 model surpasses alternatives in terms of accuracy and average accuracy, notably improving detection reliability in natural environments.

In terms of network attention, we explored the EMA module's effectiveness and observed that networks incorporating EMA exhibit higher precision, recall, and mAP50 compared to those with alternative attention mechanisms. Further, employing Grad-CAM for heatmap visualization, we noted that EMA enhances the network's focus on key areas, aiding in grape detection amidst complex backgrounds.

Regarding network lightweight processing, replacing the original backbone network with CloFormer_xxs yielded benefits in detection speed. Comparing with the original network, the proposed model demonstrated reduced image pre-processing and post-processing times.

To gauge the collective impact of the modified modules, we conducted ablation experiments, revealing that our model outperforms various combinations of individual enhancements. Additionally, comparing YOLOv8-GRAPE with other models, including YOLOv3, YOLOv5, YOLOv7, traditional YOLOv8, and variants with alternative attention mechanisms or loss functions, YOLOv8-GRAPE consistently exhibited superior performance, achieving higher mAP50, accuracy, and faster inference and preprocessing times.

In conclusion, the enhanced network introduced in this paper offers valuable insights for automating grape picking and yield prediction in vineyards. While demonstrating improved detection accuracy and speed, there is still room for further enhancements in future research. Continued efforts will focus on refining the accuracy and speed of grape detection for even greater efficacy in practical applications. The effect of this network in detecting complete overlap between grapes is not ideal. After the analysis, it was found that, in theory, the performance of the network can be further improved if the stereo fruits are collected during the data collection and the dataset containing the stereo information is imported into the network for training. This addresses the problem that grape detection is not ideal when the network completely overlaps between grapes.

References

Abdullahi, H.S., Sheriff, R., Mahieddine, F. 2017. Convolution neural network in precision agriculture for plant image recognition and classification. Proc. Seventh Int. Conf. Innovative Computing Technology. 10:256-272.

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv 2004.10934.

Castro, W., Oblitas, J., De-La-Torre, M., Cotrina, C., Bazán, K., Avila-George, H. 2019. Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. IEEE Access 7:27389-27400.

Fu, L., Duan, J., Zou, X., Lin, G., Song, S., Ji, B., Yang, Z. 2019. Banana detection based on color and texture features in the natural environment. Comp. Electron. Agr. 167:105057.

Fan, Q., Huang, H., Guan, J., He, R. 2023. Rethinking local perception in lightweight vision transformer. arXiv 2303.17803.

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., Zhang, Q. 2020. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. Comp. Electron. Agr. 176:105634.

Girshick, R. 2015. Fast r-cnn. Proc. IEEE Int. Conf. on Computer Vision, Santiago. pp. 1440-1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Columbus. pp. 580-587.

He, K., Gkioxari, G., Dollár, P., Girshick, R. 2017. Mask r-cnn. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Venice. pp. 2980-2988.

Hubel, D.H., Wiesel, T.N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol 160:106-154.

Ji, X., Dong, Z., Han, Y., Lai, C.S., Qi, D. 2023. A brain-inspired hierarchical interactive in-memory computing system and its application in video sentiment analysis. IEEE T. Circ. Syst. Vid. 33:7928-7942.

Jocher, G. yolov5. Git code. 2020. Accessed 19 Sep 2022. Available from: https://github.com/ultralytics/yolov5

Le, T.T., Lin, C. Y. 2019. Deep learning for noninvasive classification of clustered horticultural crops–A case for banana fruit tiers. Postharvest Biol. Technol. 156:110922.

Liu, W., Anguelov, D., Erhan, D., Szegedy. C., Reed, S., Fu, C.Y., Berg, A. C. 2016. Ssd: Single shot multibox detector. In: B. Leibe, J. Matas, N. Sebe and M. Welling (eds.), Computer Vision – ECCV 2016. Lecture Notes in Computer Science, vol 9905. Cham, Springer. pp. 21-37.

Lu, J., Sang, N. 2015. Detecting citrus fruits and occlusion recovery under natural illumination conditions. Comp. Electron. Agr. 110:121-130.

Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., Huang, Z. 2023, June. Efficient multi-scale attention module with cross-spatial learning. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island. pp. 1-5.

Patel, H.N., Jain, R.K , Joshi M.V. 2011. Fruit detection using improved multiple features based algorithm. Int. J. Comp. Appl. 13:1-5.

Pothen, Z.S., Nuske, S. 2016. Texture-based fruit detection via images using the smooth patterns on the fruit. Proc. IEEE Int. Conf. on on Robotics and Automation (ICRA), Stockholm. pp. 5171-5176.

Rabby, M.K.M., Chowdhury, B., Kim, J. H. 2018. A modified canny edge detection algorithm for fruit detection & classification. Proc. 10th Int. Conf. on Electrical and Computer Engineering (ICECE), Dhaka. pp. 237-240.

Ren, S., He, K., Girshick, R., Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Proc. Adv. Neur. Inf. Process. Syst. Available from: https://papers.nips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

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

Redmon, J., Farhadi A. 2017. YOLO9000: better, faster, stronger. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 7263-7271.

Redmon J., Farhadi A. 2018. Yolov3: An incremental improvement. arXiv: 1804.02767.

Sheridan, T.B. 2016. Human–robot interaction: status and challenges. Hum. Factors 58:525-532.

Tian, Y., Chen, G., Li, J., Wang X., Liu, Y., Li, H.Y. 2018. Present development of grape industry in the world. Chin. J. Trop. Agric 38:96-105.

Tian,Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z. 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comp. Electron. Agr. 157:17-426.

Tong, Z., Chen, Y., Xu, Z, Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. arXiv: 2301.10051.

Van Henten, E.J., Schenk, E.J., Van Willigenburg, L.G., Meuleman, J., Barreiro, P. 2010. Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot. Biosyst. Eng. 106:112-124.

Wouter Bac, C., Van Henten, E.J., Hemming, J., Edan, Y. 2014. Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. J. Field Robot 31:888-911.

Wang, J., Chen, Y., Ji, X., Dong, Z., Gao, M., Lai, C. S. 2023. Vehicle-mounted adaptive traffic sign detector for small-sized signs in multiple working conditions. IEEE T. Intell. Transport. Syst. 25:710-724.

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

Yan, B., Fan, P., Lei, X., Liu, Z., Yang, F. 2021. A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sens. (Basel) 13:1619.

Zhuang, J.J., Luo, S.M., Hou, C.J., Tang, Y., He, Y., Xue, X.Y. 2018. Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. Comp. Electron. Agr. 152: 64-73.

Zeeshan, M., Prabhu, A., Arun, C., Rani, N.S. 2020. Fruit classification system using multiclass support vector machine classifier. Proc. Int. Conf. on Electronics and Sustainable Communication Systems (ICESC), Coimbatore. pp. 289-294.