doi:10.6041/j.issn.1000-1298.2025.12.048

Accurate Detection of Cucumber Powdery Mildew Fungus in Microscopic Images Based on PG - YOLO v8s

ZHANG Yiding[1]　HAN Zonghuan[2]　QIAO Chen[2]　ZHANG Lingxian[2]

(1. *College of Engineering*, *China Agricultural University*, *Beijing* 100083, *China*

2. *College of Information and Electrical Engineering*, *China Agricultural University*, *Beijing* 100083, *China*)

Abstract: Cucumber is one of the most important vegetables and economic crops in the world. The occurrence of fungal diseases in cucumbers seriously threatens the safety of cucumber production, with powdery mildew being one of the most common fungal diseases. With the rapid development of computer technology, more and more deep learning algorithms are being applied to identify powdery mildew fungus. However, existing algorithms suffer from low accuracy in recognizing small and occluded targets, as well as insufficient localization precision. To address this issue, the parallelized patch-aware attention (PPA) module was firstly introduced into the backbone network of YOLO v8s. By employing a parallel multi-branch structure and attention mechanism, it effectively captured multi-scale features of small targets, preserved critical information during multiple downsampling processes, and enhanced the performance of small target detection. Additionally, the global-to-local spatial aggregation (GLSA) module was introduced into the neck, which combined global contextual information with local detail features, significantly improving the model's feature representation capability. This module enhanced the detection performance for small targets and complex scenes by better capturing multi-scale features. Experimental results showed that PG - YOLO v8s significantly improved powdery mildew fungus detection performance compared with YOLO v8s. The network achieved high precision in detecting powdery mildew fungus, with notable improvements in the detection accuracy of small and occluded targets. The research result can provide a high-throughput method for detecting powdery mildew fungus, enabling precise early detection and guiding early intelligent decision-making in cucumber production. This approach can help to improve disease control efficiency, ensure cucumber yield and quality, and it was of great significance for the sustainable development of agricultural production.

Key words: cucumber; powdery mildew fungus; YOLO v8; object detection; small objects

CLC number: TP3　**Document code**: A　**Article ID**: 1000-1298(2025)12-0522-12　**OSID**:

0　Introduction

Cucumber is one of China's major economic crops, with extensive cultivation and a significant role in the vegetable market[1]. However, fungal diseases pose a serious threat to cucumber production. Powdery mildew is one of the most common fungal diseases in cucumbers, leading to severe declines in quality and yield. In recent years, with the rapid development of computer technology, significant progress has been made in identifying, detecting, segmenting, and estimating the severity of cucumber powdery mildew. Researches[2-9] collectively focused on detecting and quantifying cucumber powdery mildew by using spectral imaging, image processing, and deep learning techniques. These works employed approaches such as visible, near-infrared, and hyperspectral imaging, convolutional neural networks, and multimodal data fusion to achieve accurate lesion segmentation, recognition, and quantitative assessment. Overall, they significantly enhanced the accuracy, automation, and early-stage detection of cucumber powdery mildew,

Received date: 2024-09-10　Accepted date: 2024-10-10

Foundation items: National Natural Science Foundation of China (62176261); China Postdoctoral Science Foundation (2024M763577)

First author: ZHANG Yiding, postdoctoral researcher. E-mail: zhangyiding117@163.com

Corresponding author: ZHANG Lingxian, professor. E-mail: zhanglx@cau.edu.cn

providing technical support for rapid identification, grading, and intelligent disease monitoring in agriculture. FERNÁNDEZ et al.[10] developed a method for detecting diseased cucumber plants by using close-range multispectral imaging technology. The research team successfully distinguished between healthy and infected plants by analyzing key spectral features in multispectral images, significantly enhancing disease detection accuracy. The study demonstrated the effectiveness of multispectral imaging in early disease detection, especially in close-range monitoring applications. REN et al.[11] addressed the low segmentation accuracy of cucumber leaf diseases in complex backgrounds by proposing an image segmentation algorithm based on saliency detection. The algorithm firstly used superpixel segmentation to obtain cucumber leaf edges, then generated a saliency map by using a novel weight calculation method and saliency seed selection method, and finally segmented the diseased leaves through manifold ranking and threshold segmentation. The lesions were precisely segmented with mathematical morphology. Compared with traditional Otsu and k-means algorithms, this method significantly reduced redundant segmentation issues, with an error rate below 5% and short execution time, laying a solid foundation for building an automated disease recognition system.

The existing studies focused on identifying, detecting, segmenting, and estimating the severity of cucumber powdery mildew and other diseases from the perspective of visible and hyperspectral images. However, this macroscopic approach struggled to unravel the pathogenic mechanisms of powdery mildew or precisely analyze the infection process. Therefore, accurately detecting the key infection structures (conidiophores and conidia) from microscopic images can help better understand the infection process of powdery mildew. Researches[12－22] focused on automated detection, recognition, and counting of fungal spores and microbial colonies by using intelligent imaging and deep learning techniques. These works integrated methods such as YOLO-based object detection, U－Net segmentation, diffraction-polarization imaging, and multi-feature fusion, achieving accurate spore identification and real-time monitoring across various plant diseases. Overall, they advanced early disease detection, quantitative analysis, and efficient monitoring by enhancing detection accuracy, speed, and adaptability under complex environmental conditions. Literatures [12－22] utilized microscopic image processing technology to detect the infection structures of cucumber pathogens, providing a deeper understanding of the infection process. Researchers had employed advanced image processing and deep learning methods, including intelligent spore capturing devices, improved YOLO models, and multi-feature fusion techniques, significantly enhancing the accuracy and efficiency of spore identification and detection. However, existing algorithms faced challenges in accurately detecting small or occluded targets and providing sufficient localization accuracy. To address these issues, a high-precision PG－YOLO v8s detection algorithm was proposed for cucumber powdery mildew infection structures.

1　Materials and methods

1.1　Infection cycle and critical infection structures of cucumber powdery mildew pathogen

The infection cycle of the cucumber powdery mildew pathogenwas illustrated in Fig. 1. The pathogen spreaded through spores via wind, water, or insects. Upon landing on cucumber leaves, stems, or other surfaces, the spores germinated if the environmental conditions were suitable (temperature 20～25℃, high humidity). The germinated spored form appressoria, penetrating plant tissues through stomata or the epidermis. The hyphae expanded and colonized the plant surface, appearing as a white powdery substance. The hyphae then produced conidiophores, which were long, upright structures that produced conidia at their tips. These conidia, vital for pathogen dissemination, were spreaded to healthy plants by wind, rain, or other carriers. Under suitable conditions, conidia germinated and infected new host plants, leading to secondary infections. During the infection process, conidiophores and conidia were vital structures in primary and secondary infections, making them crucial targets for early disease detection and management. Therefore, detecting and identifying these infection structures at the microscopic level was essential for early prevention, effective control, and informed application

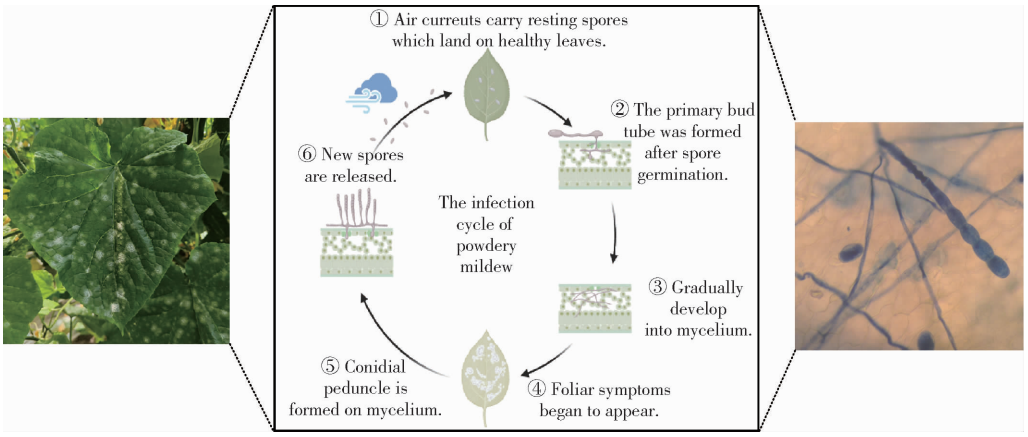


Fig. 1 Diagram of infection cycle and infection structures of cucumber powdery mildew fungus

of fungicides against cucumber powdery mildew.

1.2 Data acquisition and preprocessing

The experimental materials were provided by the Plant Protection Research Institute of Tianjin Academy of Agricultural Sciences. Since the powdery mildew pathogen is an obligate parasite, infected cucumber leaves were used as the source of the pathogen for the experiments. The cucumber leaves were collected from the Modern Agricultural Technology Innovation Base of Tianjin Academy of Agricultural Sciences, located in the western part of Wuqing District, Tianjin. The cucumber variety used was 'Mici' (Fig. 2). In situ staining was used to detect the critical infection structures of the powdery mildew pathogen. The preparation of in situ stained leaves involved steps such as fixation, decolorization, and staining.



Fig. 2 Leaves of cucumber infected with powdery mildew used in experiment

The infected leaves were carefully removed intactly. A 2 cm^2 hole punch was firstly used to cut the leaves, then placed in a fixative solution (a 1∶1 mixture of glacial acetic acid and anhydrous ethanol) for 24 h to preserve their morphology and structure. The leaf discs were then soaked in a decolorizing solution (aqueous chloral hydrate) for 1 ~ 3 h to break down internal chlorophyll, facilitating staining. Finally, the decolorized leaf discs were stained with Trypan blue for 30 min to fully infiltrate the leaf tissues, turning the critical infection structures yellow-blue. After staining, the discs were rinsed in clean water 2 ~ 3 times to remove excess dye. The stained leaf discs were mounted on slides, covered with a cover slip, and prepared for observation. Trypan blue staining solution can stain leaf discs yellow or blue-violet, depending on the activity of the plant cells.

Since the powdery mildew primarily infected the upper surface of leaves, the leaf discs were placed face-up on the slide and observed under an Olympus BX51 microscope at 400x magnification. The focus was adjusted to make the critical infection structures (conidiophores and conidia) as clear as possible, and 400 micrographs with a resolution of 960 pixels × 960 pixels were captured and saved as .jpg files.

The collected micrographs were annotated for object detection by using Labelme software[23]. Conidiophores were labeled as "Cdp" and conidia as "Cd". The annotation process was shown in Fig. 3.

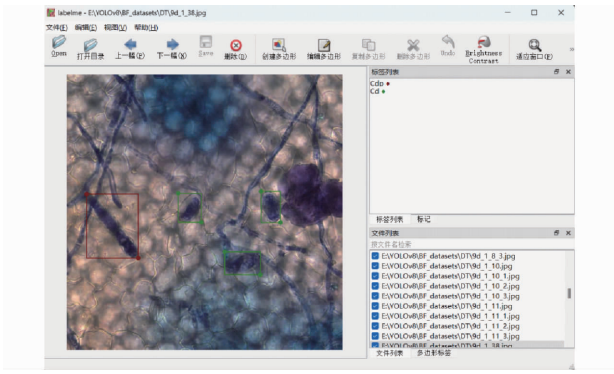
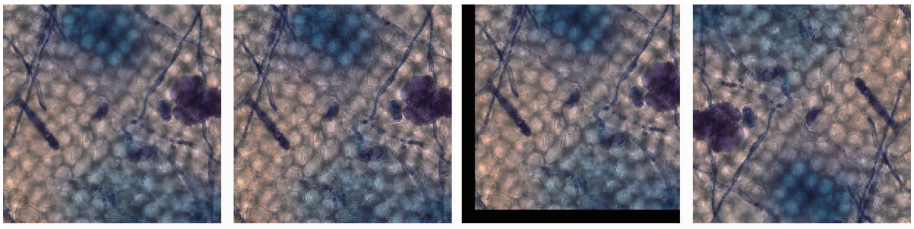


Fig. 3 Annotation process of powdery mildew fungus microscopic image dataset

The dataset of 400 images was augmented by using methods such as noise addition, translation, and flipping, resulting in 1 600 images. The expanded dataset was then divided into training, validation, and

test sets in an 8∶1∶1 ratio. The augmented dataset was shown in Fig. 4.



(a) Original image　(b) Noisy　(c) Translation　(d) Flipped

Fig. 4　Microscopic image dataset of powdery mildew fungus

1.3　Experimental platform and evaluation metrics

The experiment was conducted on a Windows 11 operating system, using PyTorch 2.1.2 as the deep learning framework. The hardware platform included an Intel Core i9－13900KF 3.0 GHz CPU and an NVIDIA GeForce RTX 4090 GPU with 24 GB of VRAM. During the model training process, the batch size was set to 8, with 150 training epochs, and an initial learning rate of 0.001 by using the Adam optimizer.

The model performance was evaluated by using mAP@0.5 and mAP@0.5∶0.95 metrics. The mAP@0.5 measured the mean average precision at an IoU threshold of 0.5, indicating that predictions had at least 50% overlap with the ground truth labels. The mAP@0.5∶0.95 measured mean average precision across multiple IoU thresholds, typically ranging from 0.5 to 0.95 with a step of 0.05, representing the model's average precision across varying overlap criteria.

The counting performance of critical infection structures in powdery mildew was evaluated by using MAE (mean absolute error) and RMSE (root mean square error) metrics. MAE directly reflected the average deviation between predicted and actual values but was less sensitive to outliers. RMSE, being more sensitive to large errors, amplified larger deviations due to the squared error, making it more responsive to significant prediction errors. This meant RMSE can better reflect model performance under extreme conditions.

2　Construction of precision detection model for cucumber powdery mildew pathogen in microscopic images based on PG－YOLO v8s

The overall structure of PG－YOLO v8s was divided into three main parts: Backbone, Neck, and Head, as shown in Fig. 5. The Backbone was responsible for feature extraction, the Neck for feature fusion and enhancement, and the Head for final object detection. This layered design allowed PG－YOLO v8s to maintain computational efficiency while delivering superior detection performance.

The Backbone was the feature extraction module of PG－YOLO v8s, designed to extract multi-scale features from input images. PG－YOLO v8s used CSPDarknet53 as the Backbone, which employed convolutional layers, pooling layers, and activation functions to extract image features gradually. CSPDarknet53 introduced a cross stage partial network (CSPNet) structure, which splited feature maps into two parts for parallel processing, enhancing feature extraction efficiency and accuracy. During this process, the spatial resolution of the feature maps was gradually decreased while the semantic information was increased. The design goal of the Backbone was to extract high-quality features with low computational cost.

The Neck connected the backbone and the Head and primarily handled multi-scale feature fusion and enhancement. The Neck structure in PG－YOLO v8s used a path aggregation network (PANet). FPN fused feature maps from different levels by using a top-down pathway and lateral connections, generating multi-scale feature maps rich in contextual information. PANet further optimized this process by enhancing feature flow with a bottom-up pathway, allowing the model to capture features of objects of different sizes better. The Neck included operations like upsampling, downsampling, and lateral connections, which enhanced the expressiveness and contextual richness of the final feature maps.

The Head was the prediction module of PG－YOLO v8s, responsible for generating the final object detection results. The Head typically consisted of

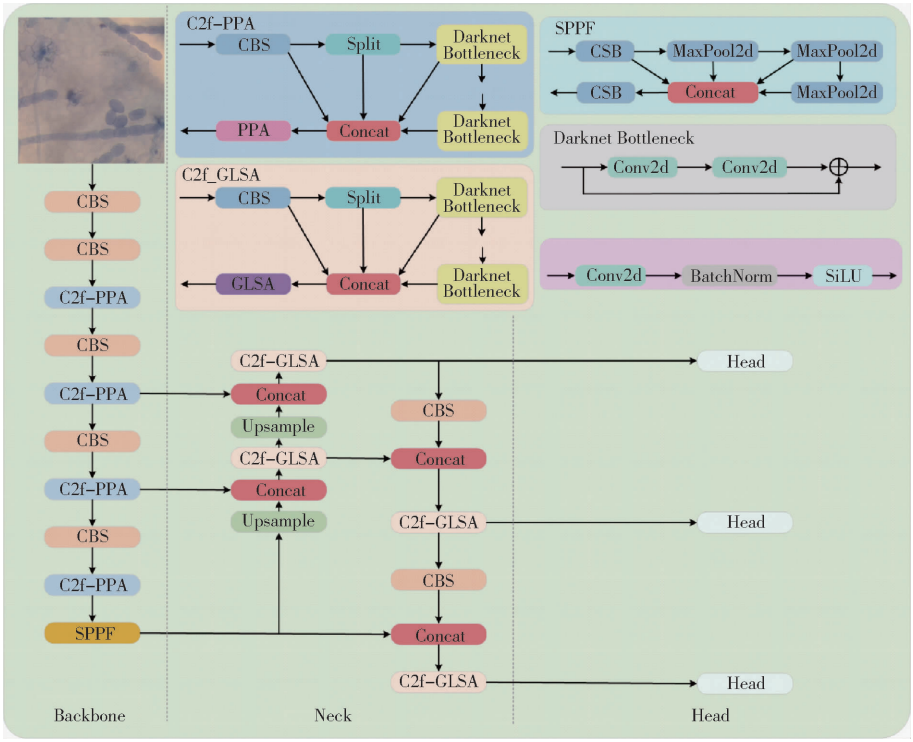


Fig. 5 Network structure diagram of PG - YOLO v8s

multiple convolutional layers that processed the feature maps from the Neck and outputed the bounding box coordinates, class probabilities, and confidence scores for object detection. The Head of PG - YOLO v8s generated multiple anchor boxes for each grid cell and calculated the regression parameters and class probabilities for these anchor boxes. The final predictions underwent post-processing steps such as non-maximum suppression (NMS) to remove redundant bounding boxes and retained the most likely targets. The design of the Head ensured that PG - YOLO v8s can accurately identify and locate objects in images while maintaining high detection speed. Through these designs, PG - YOLO v8s achieved fast and precise object detection.

2.1 C2f - PPA

PG - YOLO v8s introduced the parallel perception attention (PPA) module into the C2f part of the Backbone, constructing C2f - PPA[24]. The structure of PPA was shown in Fig. 6. PPA enhanced small object detection by extracting features at different scales and levels through a parallel multi-branch strategy. The module included local, global, and serial convolution branches, where each branch independently extracted features, and the results were combined for comprehensive feature representation. Additionally, an adaptive attention mechanism enhanced features via channel and spatial attention modules, ensuring critical information was preserved during down-sampling.

PPA significantly improved small object detection by combining parallel multi-branch feature extraction and adaptive attention mechanisms. The parallel branches included local, global, and serial convolution branches. The local and global branches extracted features by using different patch sizes, while the serial branch refined features through three layers of 3×3 convolutions. The extracted features were then fused by adding the outputs of all branches. The PPA further enhanced these features by using channel attention, which selected key channel features, and spatial attention, which highlighted critical spatial features. After batch normalization and activation, the enhanced features were output. This design boosted detection accuracy and robustness for small objects and ensured precise recognition in complex backgrounds.

2.2 C2f - GLSA

The PG - YOLO v8s introduced the global-to-local spatial aggregation module (GLSA) into the Neck part's C2f structure, resulting in the C2f - GLSA[25]. The structure of the GLSA was shown in Fig. 7. The GLSA module extracted and integrated both local and global spatial features by combining global spatial attention (GSA) and local spatial attention (LSA).

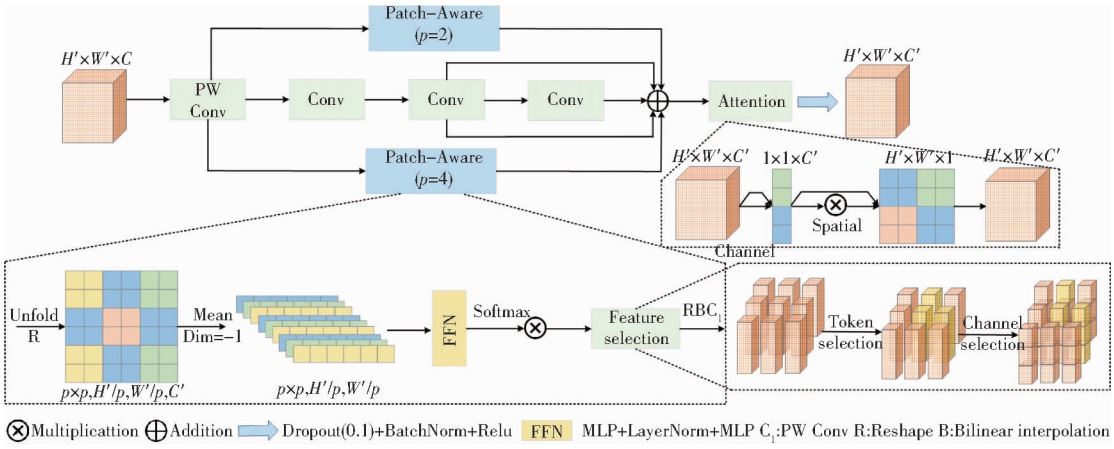


Fig. 6　Structure of PPA

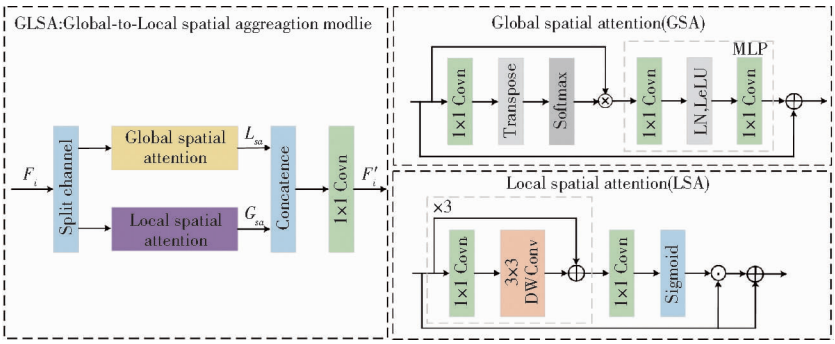


Fig. 7　Structure diagram of GLSA

The GSA module emphasized the long-range spatial relationships of each pixel, making the features more expressive, while the LSA module effectively extracted local details in the regions of interest. By dividing the feature map into two groups and feeding them separately into the GSA and LSA modules, the GLSA module balanced the model's accuracy and computational resources. This dual-stream design ensured the model's flexibility and precision in handling both detailed and global features. The GLSA module can be formulated as follows:

$$F_i^1, F_i^2 = Split(F_i) \tag{1}$$

$$F_i' = C_{1\times1}(Concat(G_{sa}(F_i^1), L_{sa}(F_i^2))) \tag{2}$$

Where F_i denoted the input feature map, G_{sa} represented global spatial attention, L_{sa} represented local spatial attention, and F_i' was the output feature map.

The GSA module can be formulated as follows:

$$Att_G(F_i^1) = Softmax(Transpose(C_{1\times1}(F_i^1))) \tag{3}$$

$$G_{sa}(F_i^1) = MLP(Att_G(F_i^1) \otimes F_i^1) + F_i^1 \tag{4}$$

Where $Att_G(\cdot)$ denoted the attention operation, $C_{1\times1}$ represented a 1 × 1 convolution, and $\otimes$ indicated matrix multiplication. The $MLP(\cdot)$ consisted of two fully connected layers with ReLU non-linear activation and normalization layers. The first layer of the $MLP(\cdot)$ transformed the input into a higher-dimensional space with an expansion ratio of 2, while the second layer restored the dimensions to match the input.

The LSA module can be formulated as follows:

$$Att_L(F_i^2) = \sigma(C_{1\times1}(F_c(F_i^2))) + F_i^2 \tag{5}$$

$$L_{sa} = Att_L(F_i^2) \odot F_i^2 + F_i^2 \tag{6}$$

Where $F_c(\cdot)$ represented a cascade of three 1 × 1 convolutional layers and a 3 × 3 convolutional layer. $Att_L(\cdot)$ denoted the local attention operation, $\sigma(\cdot)$ was the *Sigmoid* function, and $\odot$ represented pointwise multiplication.

3　Results and analysis

3.1　Selection of base model

Table 1 showed the comparative analysis results of accuracy for advanced object detection models. As can be seen from Table 1, the YOLO v8s model performed excellently across multiple performance metrics. Compared with networks such as YOLO v3, YOLO v5s, YOLO v6s, YOLO v9s, and YOLO v10s, YOLO v8s demonstrated significant performance advantages, with mAP^{box}@0.5 exceeding these five networks by 2.2, 2.2, 2.1, 2.3, and 4.6 percentage points, respectively. Moreover, mAP^{box}@0.5:0.95 exceeded these networks by 3.5, 5.8, 4.7, 2.4, and 7

percentage points, respectively. This indicated that YOLO v8s can achieve high-precision detection of the main infection structures of powdery mildew. Meanwhile, the mAP^{box}@0.5 of YOLO v8s was 0.9 percentage points lower than that of YOLO v7, while mAP^{box}@0.5:0.95 was 2.6 percentage points higher than that of YOLO v7. The lower mAP^{box}@0.5 of YOLO v8s suggested that YOLO v7 had slightly higher detection accuracy at lower thresholds, indicating that YOLO v7 may be more sensitive in detecting whether an object was presented. The higher mAP^{box}@0.5:0.95 of YOLO v8s showed that it performed better over a wider range of IoU thresholds, especially under higher precision requirements (i.e., when stricter bounding box overlap was needed), where YOLO v8s outperformed YOLO v7.

The main infection structures of powdery mildew presented challenges such as small target size and significant depth of field issues, making them difficult to detect and prone to false detections and missed detections. Given the high precision required for detection, the higher mAP^{box}@0.5:0.95 of YOLO v8s made it more suitable for detecting the main infection structures of powdery mildew.

Table 1 Comparative analysis of accuracy of advanced object detection models

Models	P/%	R/%	mAP^{box}@0.5/%	mAP^{box}@0.5:0.95/%	Parameters	Flops
YOLO v3[26]	76.1	78.0	81.7	60.8	1.0×10^8	2.8×10^{11}
YOLO v5s	79.8	76.3	81.7	58.5	9.1×10^6	2.4×10^{10}
YOLO v6s[27]	83.6	72.5	81.8	59.6	1.6×10^7	4.4×10^{10}
YOLO v7[28]	79.8	79.7	84.8	61.7	3.7×10^7	1.1×10^{11}
YOLO v8s	79.4	79.6	83.9	64.3	1.1×10^7	2.8×10^{10}
YOLO v9s[29]	77.6	79.1	81.6	61.9	7.2×10^6	2.7×10^{10}
YOLO v10s[30]	80.3	71.1	79.3	57.3	8.1×10^6	2.5×10^{10}

3.2 Ablation study

Table 2 showed the performance comparison results of YOLO v8 models at different scales. As seen from Table 2, as the model size was increased from "n" to "x", the overall performance of the network was improved, but the network's complexity (in terms of parameters and computational cost) was also increased. The YOLO v8n model had an mAP^{box}@0.5 of 83.3% and an mAP^{box}@0.5:0.95 of 62.8%, with a parameter count of only 3.0×10^6 and a computational cost of 8.1×10^9 Flops. The YOLO v8s model had an mAP^{box}@0.5 of 83.9% and an mAP^{box}@0.5:0.95 of 64.3%, representing improvements of 0.6 and 1.5 percentage points over YOLO v8n, respectively. The YOLO v8m model further improved mAP^{box}@0.5 and mAP^{box}@0.5:0.95 by 0.5 and 3.1 percentage points over YOLO v8s, but its parameter count reached 2.6×10^7 and computational cost rised to 7.9×10^{10} Flops, indicating a significant increase in network complexity. The performance of YOLO v8l was slightly lower than that of YOLO v8m, but with even higher network complexity. The YOLO v8x model achieved the highest accuracy, with mAP^{box}@0.5 and mAP^{box}@0.5:0.95 reached up to 84.7% and 67.4%, respectively. However, its parameter count and computational cost were extremely high, at 6.8×10^7 and 2.6×10^{11} Flops, making it unsuitable for deployment. Taking all factors into consideration, the YOLO v8s model was selected as the base model for further improvement, as it struck a good balance between model performance and network complexity.

Table 3 showed the results of the ablation experiments. As illustrated in Table 3, the mAP^{box}@0.5 and mAP^{box}@0.5:0.95 of YOLO v8s were 83.9% and 64.3%, respectively. After introducing the PPA

Table 2 Comparative analysis of different network architectures

Model	P/%	R/%	mAP^{box}@0.5/%	mAP^{box}@0.5:0.95/%	Parameters	Flops
n	82.8	74.1	83.3	62.8	3.0×10^6	8.1×10^9
s	79.4	79.6	83.9	64.3	1.1×10^7	2.8×10^{10}
m	79.2	80.3	84.4	67.4	2.6×10^7	7.9×10^{10}
l	78.9	79.8	84.3	66.7	4.4×10^7	1.6×10^{11}
x	81.5	75.9	84.7	67.4	6.8×10^7	2.6×10^{11}

Table 3　Results of ablation experiments

YOLO v8s	PPA	GLSA	mAP^{box}@0.5/%	mAP^{box}@0.5:0.95/%	Parameters	Flops
√			83.9	64.3	1.1×10^7	2.8×10^{10}
√	√		84.4	64.4	1.3×10^7	3.2×10^{10}
√		√	84.1	64.5	1.5×10^7	3.3×10^{10}
√	√	√	84.6	65.4	1.7×10^7	3.7×10^{10}

Note: √ represented the adoption of this method.

module, which enhanced feature representation across different scales and levels through parallel multi-branch feature extraction and an adaptive attention mechanism, the detection accuracy for small objects was improved. This results in an increase of 0.5 and 0.1 percentage points in mAP^{box} @ 0.5 and mAP^{box}@0.5:0.95 compared with YOLO v8s. After introducing the GLSA module, which combined GSA and LSA to simultaneously extract and fuse local spatial detail information and global spatial semantic information, the detection accuracy for small objects and boundary refinement capabilities were further improved. This resulted in an increase of 0.2 percentage points in both mAP^{box}@0.5 and mAP^{box}@0.5:0.95 compared with that of YOLO v8s. When both modules were introduced simultaneously, the mAP^{box} @ 0.5 and mAP^{box} @ 0.5:0.95 of PG－YOLO v8s reached up to 84.6% and 65.4%, representing improvements of 0.7 and 1.1 percentage points over YOLO v8s, respectively. Therefore, PG－YOLO v8s can achieve high-precision detection of powdery mildew infection structures.

Table 4 showed the performance comparison analysis results of different attention modules. As shown in Table 4, the SDFM achieved mAP^{box}@0.5 and mAP^{box}@0.5:0.95 rates of 81.5% and 58.3%, respectively, with relatively poor detection accuracy for small objects. The CGAFusion achieved mAP^{box}@0.5 and mAP^{box}@0.5:0.95 rates of 82.4% and 60.0% respectively, showing some improvement in detecting occluded and tiny objects compared with SDFM. FADC had a higher detection accuracy with mAP^{box}@0.5 reaching 82.7%, but its mAP^{box}@0.5:0.95 was 0.5 percentage points lower than that of CGAFusion, indicating that the model might struggle to accurately locate the precise boundaries of targets. Similar to FADC, MLCA can also detect the infection structures of powdery mildew with high precision but it was found difficult to accurately determine the peripheral boundaries of the infection structures. GLSA achieved mAP^{box}@0.5 and mAP^{box}@0.5:0.95 rates of 84.1% and 64.5%, respectively, the best performance among all experimental groups, demonstrating that GLSA had strong capability in detecting clear targets, occluded targets, and tiny targets. This facilitated high-precision detection of the main infection structures of powdery mildew and consequently allowed for accurate counting of these structures.

Table 4　Comparison of different attention module performance

Module	mAP^{box}@0.5/%	mAP^{box}@0.5:0.95/%	Parameters	Flops
SDFM[31]	81.5	58.3	1.3×10^7	3.0×10^{10}
CGAFusion[32]	82.4	60.0	1.2×10^7	2.9×10^{10}
FADC[33]	82.7	59.5	1.1×10^7	2.7×10^{10}
MLCA[34]	83.2	59.5	1.1×10^7	2.8×10^{10}
GLSA[25]	84.1	64.5	1.5×10^7	3.3×10^{10}

3.3　Comparative analysis of performance of advanced object detection models

Table 5 presented the comparative analysis results of accuracy on the test set for advanced object detection models. As shown in Table 5, the detection capabilities of YOLO v3 and YOLO v10s were relatively poor, with mAP^{box}@0.5 values of 89.1% and 88.4%, respectively, both below 90.0%. This indicated that these two models had weaker perception abilities for detecting the main infection structures of powdery mildew, leading to a higher likelihood of missed detections. On the other hand, networks like YOLO v5s, YOLO v6s, YOLO v7, and YOLO v9s had mAP^{box}@0.5 values of 90.5%, 90.9%, 90.8%, and 90.2%, respectively, showing only minor performance differences. However, their mAP^{box}@0.5:0.95 values were 66.0%, 68.2%, 65.6%, and 69.3%, respectively, indicating more significant performance differences under higher precision requirements. YOLO v5s and YOLO v7 showed weaker performance under stricter metrics, suggesting that they had less capability in detecting the edges of the main infection structures

Table 5 Comparative analysis of accuracy of advanced object detection models on test dataset

Models	P/%	R/%	mAP^{box}@0.5/%	mAP^{box}@0.5:0.95/%	Parameters	Flops
YOLO v3	89.3	79.2	89.1	69.7	1.0×10^{8}	2.8×10^{11}
YOLO v5s	87.1	83.5	90.5	66.0	9.1×10^{6}	2.4×10^{10}
YOLO v6s	85.9	87.3	90.9	68.2	1.6×10^{7}	4.4×10^{10}
YOLO v7	87.2	83.5	90.8	65.6	3.7×10^{7}	1.1×10^{11}
PG-YOLO v8s	88.5	84.5	91.8	70.0	1.7×10^{7}	3.7×10^{10}
YOLO v9s	83.1	84.0	90.2	69.3	7.2×10^{6}	2.7×10^{10}
YOLO v10s	81.9	82.8	88.4	66.8	8.1×10^{6}	2.5×10^{10}

of powdery mildew. In contrast, YOLO v6s and YOLO v9s can detect the main infection structures of powdery mildew with a higher intersection over union (IoU), allowing for more accurate counting. The proposed PG-YOLO v8s network performed the best among all experimental groups, achieving mAP^{box}@0.5 and mAP^{box}@0.5:0.95 values of 91.8% and 70.0%, respectively. It can detect the main infection structures of powdery mildew with high precision, offering significant advantages over other networks.

Fig. 8 was a comparison chart of the segmentation effects of advanced object detection models. As shown in Fig. 8, the detection performance of various networks in a complex background was illustrated. According to label Fig. 8b, the image contained four conidiophores and two conidia. YOLO v3 mistakenly recognized the occluded conidia at the lower left corner of the frame as a conidiophore, increasing the counting error of infection structures. YOLO v5s had poor detection accuracy for occluded targets, failing to detect both a conidiophore and a conidia at the lower left corner of the frame. YOLO v6s not only missed a conidia at the lower left corner of the frame but also produced an erroneous detection box, repeatedly

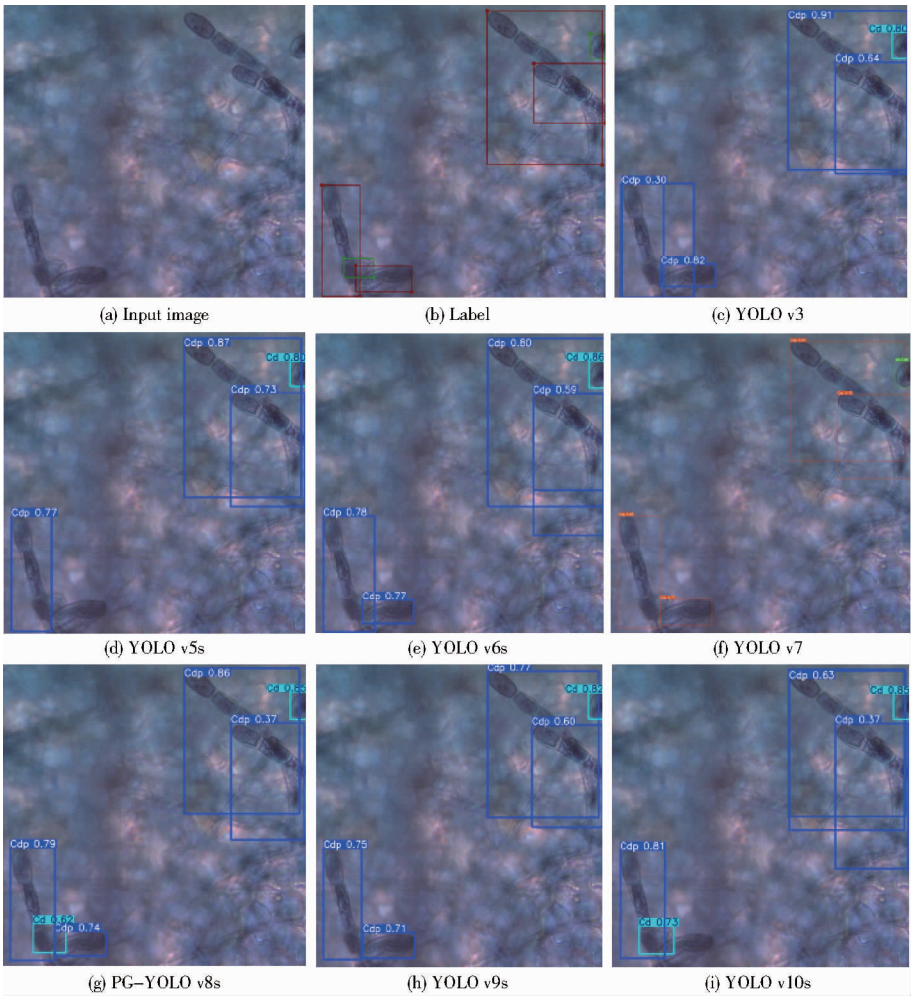


Fig. 8 Comparative analysis of detection results from different object detection models

detecting a conidiophore at the upper right corner of the frame, thereby increasing the counting error of infection structures. Both YOLO v7 and YOLO v9s showed poor detection accuracy for occluded targets, failing to detect the conidia occluded by a conidiophore at the lower left corner of the frame. YOLO v10s demonstrated weak perception ability for occluded targets, missing a conidiophore at the lower left corner of the frame, and wrongly detecting the two conidiophores at the upper right corner as three, resulting in a significant counting error of infection structures. Only the PG－YOLO v8s network correctly detected all four conidiophores and two conidia, with the generated bounding boxes most closely matching the labels, enabling high-precision detection of the main infection structures of powdery mildew.

3.4 Comparison of counting performance of different networks for main infection structures of powdery mildew

The main infection structures of powdery mildew fungi include conidiop hores (Cdp) and conidia (Cd). Table 6 showed the comparison results of the counting performance of different networks for the main infection structures of powdery mildew. As shown in Table 6, YOLO v3 and YOLO v6s had relatively large counting errors for conidiophores, with MAE (Cdp) both at 0.588 and RMSE (Cdp) at 0.968 and 1.101, respectively. This indicated that these two networks struggled to accurately recognize conidiophores in complex backgrounds. YOLO v5s, YOLO v7, YOLO v9s, and YOLO v10s had relatively acceptable counting accuracy for conidiophores, with MAE (Cdp) values of 0.475, 0.444, 0.438, and 0.500, respectively. The PG－YOLO v8s achieved the highest accuracy for conidiophore counting, with MAE (Cdp) and RMSE (Cdp) reaching 0.388 and 0.716, respectively.

Table 6　Comparison of counting performance of different networks on main infection structures of powdery mildew fungus

Model	MAE (Cdp)	RMSE (Cdp)	MAE (Cd)	RMSE (Cd)	MAE (All)	RMSE (All)
YOLO v3	0.588	0.968	0.325	0.632	0.838	1.275
YOLO v5s	0.475	0.766	0.656	1.126	1.044	1.460
YOLO v6s	0.588	1.101	0.375	0.725	0.838	1.332
YOLO v7	0.444	0.795	0.706	1.272	0.988	1.750
PG－YOLO v8s	0.388	0.716	0.406	0.818	0.769	1.217
YOLO v9s	0.438	0.750	0.650	1.095	1.000	1.432
YOLO v10s	0.500	0.806	0.531	1.028	0.844	1.247

Regarding conidium counting, YOLO v3, YOLO v6s, and PG－YOLO v8s demonstrated higher accuracy. The MAE (Cd) values were 0.325, 0.375, and 0.406, respectively, and the RMSE (Cd) values were 0.632, 0.725, and 0.818, respectively. In contrast, the other networks had MAE (Cd) values greater than 0.5 and RMSE (Cd) values greater than 1, making accurate conidium counting challenging.

Overall, the PG－YOLO v8s network achieved the highest counting accuracy for the main infection structures of powdery mildew, with MAE (All) and RMSE (All) reaching 0.769 and 1.217, respectively, making it the most outstanding performer among all experimental groups. It achieved high accuracy in both individual counting of conidiophores and conidia, as well as in overall counting. Therefore, the PG－YOLO v8s network was suitable for practical application, providing important guidance for the early prevention and control of cucumber powdery mildew, grasping key moments in disease management, and making informed decisions on medication use.

4 Conclusion

Cucumber is one of the most important vegetables and economic crops globally, but fungal diseases pose a serious threat to its production safety, with powdery mildew being one of the most common fungal diseases. With the rapid development of computer technology, more and more deep learning algorithms are being applied to the identification of powdery mildew pathogens. However, current algorithms have low recognition accuracy when dealing with small and occluded targets, and their localization capabilities need improvement. To address this issue, a PPA module was firstly introduced into the Backbone network of YOLO v8s. Through the parallel application of a multi-branch structure and attention mechanism, this module effectively captured multi-scale features of small targets, retaining critical information during downsampling, and significantly improved the detection performance for small targets. Additionally, a GLSA module was introduced in the Neck part, which combined global contextual information with local detail features, significantly enhancing the model's feature representation ability and improving detection

performance in complex scenes. Experimental results showed that PG－YOLO v8s significantly outperformed YOLO v8s in detecting powdery mildew pathogens, especially in terms of accuracy for small and occluded target detection. An efficient method for detecting powdery mildew pathogens was proposed, enabling accurate identification of the pathogens in the early stages of disease development, thereby providing guidance for early intelligent decision-making in cucumber production. This, in turn, improved disease control efficiency, ensured cucumber yield and quality, and contributed to the sustainable development of agricultural production.

References

[1] HUI Chengzhang, LIU Shujun, LIU Aiqun. Research on the high-quality development of cucumber industry in the new era[J]. Agricultural Science & Technology and Equipment, 2023(6):85－87. (in Chinese)

[2] XU Jitong, JIN Hairong, TONG Wenyu, et al. Segmentation and detection of powdery mildew by comparison of near-infrared and fluorescence spectra[J]. Spectroscopy and Spectral Analysis, 2023,43(6):1731－1738. (in Chinese)

[3] LIN K, GONG L, HUANG Y, et al. Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network[J]. Frontiers in Plant Science, 2019, 10: 155.

[4] XU J T, ZHANG Z, GUO Y H, et al. Detection of cucumber powdery mildew based on spectral and image information[J]. Journal of Biosystems Engineering, 2023, 48(2): 115－122.

[5] FERNÁNDEZ C I, LEBLON B, WANG J, et al. Cucumber powdery mildew detection using hyperspectral data[J]. Canadian Journal of Plant Science, 2021, 102(1): 20－32.

[6] GONG L, YU C, LIN K, et al. A lightweight powdery mildew disease evaluation model for its in-field detection with portable instrumentation[J]. Agronomy, 2021, 12(1): 97.

[7] BAI Xuebing, YU Jianshu, Fu Zetian, et al. Segmentation and detection of cucumber powdery mildew based on visible spectrum and image processing[J]. Spectroscopy and Spectral Analysis, 2019,39(11):3592－3598. (in Chinese)

[8] ELDESOKY A A I, EL-NABI A, OMRAN E E. Detecting mildew diseases in cucumber using image processing technique[J]. Misr Journal of Agricultural Engineering, 2023, 40(3): 243－258.

[9] BAI X, FU Z, STANKOVSKI S, et al. A three-dimensional threshold algorithm based on histogram reconstruction and dimensionality reduction for registering cucumber powdery mildew[J]. Computers and Electronics in Agriculture, 2019, 158: 211－218.

[10] FERNÁNDEZ C I, LEBLON B, WANG J, et al. Detecting infected cucumber plants with close-range multispectral imagery [J]. Remote Sensing, 2021, 13(15): 2948.

[11] REN Shougang, LU Haifei, YUAN Peisen, et al. Segmentation algorithm of cucumber leaf disease image based on saliency detection[J]. Transactions of the Chinese Society for Agricultural Machinery, 2016, 47(9):11－16. (in Chinese)

[12] DANPING W, BOTAO W, YUE Y. The identification of powdery mildew spores image based on the integration of intelligent spore image sequence capture device [C] // 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE, 2013: 177－180.

[13] LI Ming, DING Zhihuan, ZHAO Jingxuan, et al. Detection method for cucumber downy mildew sporangia in a solar greenhouse based on improved YOLOv5s[J]. Journal of Chinese Agricultural Mechanization, 2023,44(5):63－70. (in Chinese)

[14] WANG Y, ZHANG X, TAHA M F, et al. Detection method of fungal spores based on fingerprint characteristics of diffraction-polarization images[J]. Journal of Fungi, 2023, 9(12): 1131.

[15] QIAO C, LI K, ZHU X, et al. Detection of cucumber downy mildew spores based on improved YOLOv5s[J]. Information Processing in Agriculture, 2025, 12(2): 179－194.

[16] LI K, ZHU X, QIAO C, et al. The gray mold spore detection of cucumber based on microscopic image and deep learning[J]. Plant Phenomics, 2023, 5: 0011.

[17] WANG Y, DU X, MA G, et al. Classification methods for airborne disease spores from greenhouse crops based on multifeature fusion[J]. Applied Sciences, 2020, 10(21): 7850.

[18] QIAO Chen, HAN Mengyao, GAO Wei, et al. Quantitative detection of cucumber downy mildew spores at multi-scale based on Faster－NAM－YOLO[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(12): 288－299. (in Chinese)

[19] LIANG X, WANG B. Wheat powdery mildew spore images segmentation based on U－Net[J]. Journal of Physics: Conference Series, 2020, 1631(1): 012074.

[20] ZHANG B, ZHOU Z, CAO W, et al. A new few-shot learning method of bacterial colony counting based on the edge computing device[J]. Biology, 2022, 11(2): 156.

[21] HE S, MINN K T, SOLNICA-KREZEL L, et al. Deeply-supervised density regression for automatic cell counting in

microscopy images[J]. Medical Image Analysis, 2021, 68: 101892.
[22] ALBARADEI S A, NAPOLITANO F, ULUDAG M, et al. Automated counting of colony forming units using deep transfer learning from a model for congested scenes analysis[J]. IEEE Access, 2020, 8: 164340－164346.
[23] RUSSELL B C, TORRALBA A, MURPHY K P, et al. LabelMe: a database and web-based tool for image annotation[J]. International Journal of Computer Vision, 2008, 77: 157－173.
[24] XU S, ZHENG S, XU W, et al. Hcf-net: hierarchical context fusion network for infrared small object detection[C]//2024 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2024: 1－6.
[25] TANG F, XU Z, HUANG Q, et al. DuAT: dual-aggregation transformer network for medical image segmentation[C]// Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Singapore: Springer Nature Singapore, 2023: 343－356.
[26] REDMON J, FARHADI A. YOLO v3: an incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
[27] LI C, LI L, JIANG H, et al. YOLOv6: a single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.
[28] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 7464－7475.
[29] WANG C Y, YEH I H, LIAO H Y M. YOLO v9: learning what you want to learn using programmable gradient information [J]. arXiv preprint arXiv:2402.13616, 2024.
[30] WANG A, CHEN H, LIU L, et al. YOLO v10: real-time end-to-end object detection[J]. arXiv preprint arXiv:2405.14458, 2024.
[31] TANG L, ZHANG H, XU H, et al. Rethinking the necessity of image fusion in high-level vision tasks: a practical infrared and visible image fusion network based on progressive semantic injection and scene fidelity[J]. Information Fusion, 2023, 99: 101870.
[32] CHEN Z, HE Z, LU Z M. DEA－Net: single image dehazing based on detail-enhanced convolution and content-guided attention[J]. IEEE Transactions on Image Processing, 2024, 33: 1002－1015.
[33] CHEN L, GU L, ZHENG D, et al. Frequency-adaptive dilated convolution for semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024: 3414－3425.
[34] WAN D, LU R, SHEN S, et al. Mixed local channel attention for object detection[J]. Engineering Applications of Artificial Intelligence, 2023, 123: 106442.

基于 PG－YOLO v8s 的黄瓜白粉病菌显微图像精准检测

张一丁[1]　韩宗桓[2]　乔　琛[2]　张领先[2]
(1. 中国农业大学工学院，北京 100083；2. 中国农业大学信息与电气工程学院，北京 100083)

摘要： 黄瓜是世界上最重要的蔬菜和经济作物之一。黄瓜真菌病害的发生严重威胁着黄瓜的生产安全，白粉病是其最常见的真菌病害之一。随着计算机技术的飞速发展，越来越多的深度学习算法应用于白粉病菌识别。然而，现有算法存在着微小、遮挡目标识别精度低，定位精度不足的问题。针对这一问题，本文首先将并行化感知注意模块(Parallelized patch-aware attention，PPA)引入 YOLO v8s 的骨干网络中，通过并行的多分支结构和注意力机制，有效地捕捉了小目标的多尺度特征，在多次下采样过程中保留了关键信息，提高了小目标检测的性能。此外，将全局到局部空间聚合模块(Global-to-local spatial aggregation，GLSA)引入 Neck 中，通过结合全局上下文信息和局部细节特征，显著提高了模型的特征表达能力。它能够更好地捕捉多尺度特征，从而增强小目标和复杂场景中的检测性能。实验结果表明，PG－YOLO v8s 对白粉病菌的检测性能较 YOLO v8s 大幅提升，网络对小目标和遮挡目标的检测精度显著提升，实现了高精度的白粉病菌检测。本文提供了一种高通量的白粉病菌检测方法，可在发病早期精准检测白粉病菌，指导黄瓜生产的早期智能决策。有助于提高病害防控效率，保障黄瓜产量和质量，对农业生产的可持续发展具有重要意义。

关键词： 黄瓜；白粉病菌；YOLO v8；目标检测；小目标

基金项目： 国家自然科学基金面上项目(62176261)和中国博士后科学基金项目(2024M763577)