RSA-YOLO: tomato leaf disease and pest detection model based on improved YOLOv8

Qin Dong

dongqin@ycit.edu.cn

Yancheng Institute of Technology

Fei Wang

Yancheng Institute of Technology

Mingping Tang

Yancheng Institute of Technology

Wei Wu

Jiangsu Kesheng Group Co., Ltd.

Research Article

Keywords: tomato leaf, disease and pest detection, target detection, YOLOv8n, RFA

Posted Date: November 11th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7972007/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Signal, Image and Video Processing on February 3rd, 2026. See the published version at https://doi.org/10.1007/s11760-026-05125-8.

RSA-YOLO: tomato leaf disease and pest detection model based on improved YOLOv8

Qin Dong[1*], Fei Wang[1], Mingping Tang[1], Wei Wu[2]

[1]School of Information Engineering, Yancheng Institute of Technology,Yancheng 224000, People's Republic of China.

[2]Kesheng Group Co., Ltd., Jiangsu Province, People's Republic of China.

*Corresponding author(s). E-mail(s): dongqin@ycit.edu.cn;

Abstract

Currently, the detection of diseases and pests on tomato leaves presents several critical challenges, including ambiguous multi-scale target recognition, significant background interference, and limited adaptability to varying lighting conditions. To address these issues, this study proposed an efficient detection model for tomato leaf diseases and pests based on YOLOv8n, named RSA-YOLO. First, a Receptive-Field Attention (RFA) mechanism was integrated into the backbone network to overcome the limitations of conventional convolutional kernel parameter sharing by dynamically adjusting the spatial features within the convolutional kernel's receptive-field. This enhancement effectively improves the model's capability to extract features from complex patterns. Second, to enhance both efficiency and accuracy in multi-scale feature fusion, the original Spatial Pyramid Pooling Fast (SPPF) module in YOLOv8n was replaced with the Spatial Pyramid Pooling with Efficient Layer Aggregation Network (SPPELAN) module. Finally, an Adaptive Spatial Feature Fusion (ASFF) mechanism was incorporated into the detection head to strengthen the scale invariance of features, thereby better addressing variations in target size. Experimental results demonstrated that RSA-YOLO significantly outperformed the original YOLOv8n, with Precision (P) increasing by 1.6%, mean Average Precision (mAP@0.5) improving by 2.6%, and F1-score rising by 2.4%. Furthermore, compared to YOLOv3-tiny, YOLOv5, YOLOv6, YOLOv9, YOLOv11, YOLOV12, and YOLOv13, RSA-YOLO achieved mAP@0.5 improvements of 6.8%, 4.1%, 4.5%, 2.4%, 2.6%,3.5% and 1.9%, respectively. The results indicate that RSA-YOLO significantly improves detection accuracy while maintaining a moderate and controllable computational cost and model size. This demonstrates its feasibility for deployment on portable or resource-constrained devices in agricultural scenarios and provides technical support for monitoring and controlling tomato leaf diseases and pests.

Keywords: tomato leaf, disease and pest detection, target detection, YOLOv8n, RFA

1 Introduction

Tomatoes, a globally important cash crop, are typically cultivated in warm and humid environments that are highly susceptible to pathogens and pests. Common tomato leaf diseases and pests—such as septoria leaf spot, late blight, and leaf miner infestations—can significantly reduce both tomato yield and quality, leading to substantial economic losses for farmers [1]. Currently, the prevention and control of tomato leaf diseases and pests face challenges due to their diversity and unpredictable outbreaks [2]. Although various chemical control methods are available, curative treatments often incur higher costs than preventive measures. Furthermore, the excessive use of chemical pesticides not only destabilizes agricultural ecosystems but also promotes pest resistance, thereby increasing control expenses [3]. Therefore, early detection of abnormalities during the tomato growth phase, coupled with timely management of diseases and pests, can enhance both yield and quality while reducing pesticide dependence. Consequently, timely detection of tomato leaf diseases and pests at an early stage is critical [4].

In the field of crop disease and pest detection, traditional methods predominantly rely on manual judgment, which is both inefficient and prone to subjective bias, leading to low accuracy and inconsistent results [5]. As the scale of agricultural production expands and geographic diversity increases, these traditional methods struggle to adapt to the complex and dynamic nature of modern agricultural environments. This is largely due to their limited monitoring coverage and inadequate real-time responsiveness [6]. Specifically, in the detection of early-stage and complex diseases and pests, traditional methods often fail to produce satisfactory outcomes.

In recent years, machine learning techniques have gradually been introduced into the field of disease and pest detection to address the limitations of traditional manual judgment. Madhav et al. [7] utilized Support Vector Machine (SVM) algorithms to classify diseases and control pesticide residues, following the

segmentation and processing of fruit images using image processing techniques. Bera et al. [8] developed PND-Net, which integrated Graph Convolutional Network (GCN) with Convolutional Neural Network (CNN) and captured multi-scale features through spatial pyramid pooling. The performance improvements in plant nutrient deficiency and disease classification were validated on multiple laboratory datasets. Prabu et al. [9] proposed an enhanced SVM model based on an arithmetic optimization algorithm (BSVM-AOA), which was combined with a vector-valued active contour model for image segmentation and utilized a gray-level co-occurrence matrix for feature extraction, achieving high-precision detection of plant leaf diseases. Wu et al. [10] improved the ResNet model for strawberry disease identification by strategically embedding two Squeeze-and-Excitation (SE) blocks, which increased recognition accuracy while maintaining low computational costs. However, these machine learning methods are only suitable for single disease recognition in standard laboratory images and cannot identify the specific location of the disease.

With the significant advancements in deep learning, target detection methods based on Convolutional Neural Networks (CNNs) have gradually become mainstream, leading to two primary technical approaches: two-stage and single-stage detection models. Common two-stage detection models include R-CNN [11], Faster R-CNN [12], and VFNet [13], among others. Using Faster R-CNN as a representative example, this approach achieves superior detection accuracy by generating candidate regions in the first stage and performing fine classification and regression in the second stage. However, it suffers from inherent drawbacks, such as high computational complexity and slow inference speed.

In contrast, single-stage detectors, represented by YOLO [14, 15] and SSD [16], significantly improve efficiency by integrating the target localization and classification tasks into a single, fully convolutional network. For instance, Ullah et al. [17] proposed the EffiMob-Net model for tomato leaf disease identification, which enhanced feature extraction by fusing EfficientNetB3 and MobileNet, mitigated overfitting by combining regularization, dropout, and batch normalization techniques, and optimized the model using hyperparameter tuning to achieve high-precision disease detection. Luo et al. [18] proposed the Light-SA YOLOv8 model, specifically optimized for citrus disease and pest detection. The model improved adaptability to light changes through the BRA self-attention mechanism and combined the FasterNet lightweight architecture with an enhanced AFPN feature fusion strategy to maintain high detection efficiency under complex background interference. In greenhouse vegetable disease detection, Wang et al. [19] streamlined the YOLOv8 network parameters through the C2fGhost module and integrated the Occlusion Perception Attention Module (OAM) to enhance local feature expression, significantly improving detection in dense occlusion scenarios. For crop-specific disease detection, Lu et al. [20] proposed the YOLOv8 Rice model to address the challenge of recognizing small, irregular targets in rice disease detection. The model improved the detection of complex plant lesions by incorporating deformable convolution into the C2f module to enhance deformation adaptability, and by using a weighted bidirectional feature pyramid to refine multiscale feature fusion. Qiang et al. [21] developed a dual-backbone network using ResNet50 and VGG16 in parallel, improving the SSD detection algorithm. By exploiting the complementary nature of deeper semantic features and shallower detail information, the model significantly improved localization accuracy for citrus diseases and pests.

The above studies show that although CNNs have achieved notable progress in the detection of leaf diseases and pests, there remains room for improvement in recognition accuracy. In particular, under complex and unstructured image backgrounds, CNNs exhibit limited capability in extracting subtle disease features, often failing to capture disease-specific details effectively. Moreover, adverse weather conditions also impair detection performance. To address these challenges, this study was based on the YOLOv8n [22] detection algorithm. Specifically, a Receptive Field Attention (RFA) [23] module was incorporated to enhance the feature extraction capability. To further strengthen the representation power of the feature pyramid, the original Spatial Pyramid Pooling Fast (SPPF) module was replaced with a Spatial Pyramid Pooling with Efficient Layer Aggregation Network (SPPELAN) [24] module. Additionally, an Adaptive Spatial Feature Fusion (ASFF) [25] mechanism was integrated into the detection head to optimize cross-scale feature interactions. With these enhancements, the proposed RSA-YOLO model achieved accurate identification of four types of tomato leaf diseases and pests. This study evaluated the model's training performance using multiple metrics, providing reliable data support for the detection and treatment of tomato leaf diseases and pests.

2 Proposed methods

2.1 YOLOv8 network architecture

YOLOv8, an enhanced version of YOLOv5, supports image classification, object detection, and instance segmentation. Its architecture consists of three core components: the backbone, neck, and head. The backbone employs an enhanced CSP-Darknet [26, 27] architecture, which decomposes the model's computational tasks into different stages through phased feature extraction, thereby enabling efficient feature representation. The neck integrates a bidirectional feature pyramid with PAN structure to combine multi-scale features, thereby improving small-object detection. The head adopts a decoupled design, separating the classification and bounding box regression tasks. Additionally, the loss function dynamically combines classification confidence and localization quality via a Task-Aligned Assigner [28] for sample assignment, while Distribution Focal Loss [29] optimizes the distribution of bounding boxes and is combined with CIoU Loss [30] to enhance localization accuracy. The YOLOv8 backbone extracts features, which are then fused by the neck to form rich multi-scale feature representations. Detection is performed by the head. This design improves both the speed and accuracy of YOLOv8.

2.2 Improvement of YOLOv8n

To achieve precise detection of tomato leaf lesions and pest-infested regions under complex backgrounds and multi-scale conditions, an efficient model named RSA-YOLO was proposed based on YOLOv8n. The YOLOv8 model faces challenges in simultaneously preserving the fine details of small targets and the semantic information of large ones while processing multi-scale objects. This limitation results in insufficient feature extraction, especially in scenarios with complex backgrounds and varying lighting conditions, where important details are

often lost. To enhance its feature extraction capability, the RFA module was introduced into the backbone, replacing the Conv and C2f modules, which significantly improved the robustness of cross-scale feature extraction. Furthermore, the SPPF module in the feature pyramid was replaced with the SPPELAN module to optimize the efficiency of multi-scale feature fusion. Finally, the ASFF module was integrated into the detection head to address scale inconsistencies within the feature pyramid, thereby mitigating performance degradation during multi-scale detection. Through these designs, the detection accuracy for large, medium, and small targets was significantly enhanced, and the model's adaptability to complex, multi-scale scenarios was comprehensively strengthened. The overall architecture of RSA-YOLO is illustrated in Fig. 1.

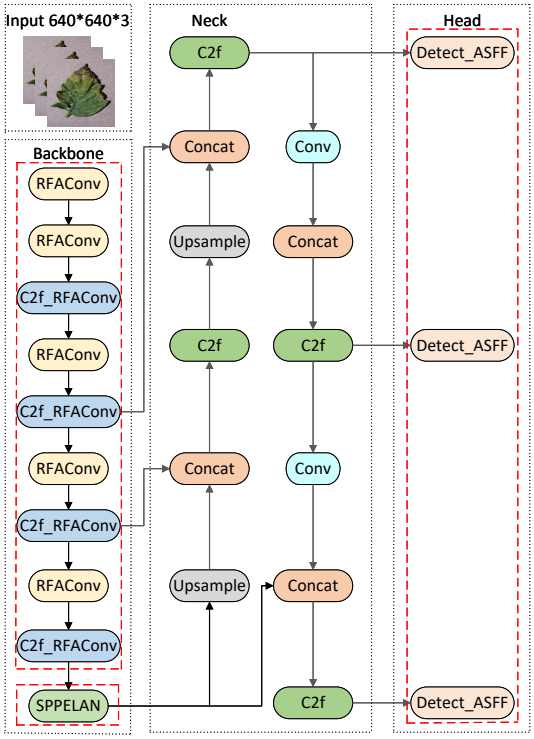


Fig. 1: The structure of RSA-YOLO

2.2.1 Feature enhancement module based on RFA

Traditional convolution operations are prone to local noise interference when processing complex scenes, leading to limited ability in capturing long-range dependencies and contextual information, which may result in misdetection or omission. To address this limitation, the RFA module was introduced in this paper, as shown in Fig. 2. Taking the 3×3 convolutional kernel as an example, the "Spatial Feature" represents the initial input feature map, while "Receptive-Field Spatial Feature" refers to the feature mapping generated by the sliding window transformation, consisting of non-overlapping sliding windows. Each receptive-field slider corresponds to a 3×3 window region in the feature map, and these spatial units serve as the basic elements for allocating attention weights.

Inspired by RFA, we proposed a novel module named RFAConv as a replacement for the standard convolution operation, as illustrated in Fig. 3.The input feature map $X \in R^{C \times H \times W}$ is divided into three groups: Group1, Group2, and Group3, following an average pooling operation. Here, C, H and W represent the channel number, height, and width of the input, respectively. Then, information interaction is achieved via a 1×1 group convolution operation that generates three corresponding convolutional kernels: Kernel1, Kernel2, and Kernel3. Subsequently, an attention map with dimensions $9C \times H \times W$ is produced by applying the Softmax function, emphasizing the importance of each feature within the receptive-field. Meanwhile, the input feature maps undergo

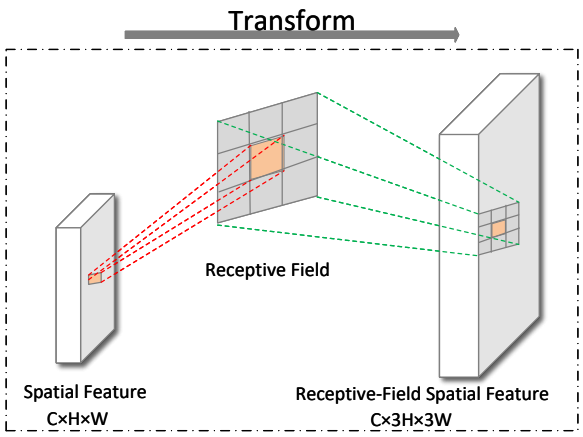


Fig. 2: The receptive-field spatial features are obtained by transforming the spatial features

a 3×3 group convolution to extract receptive field spatial features with dimensions $9C \times H \times W$. After reshaping, the attention map is fused with the receptive-field spatial features, and a reweighting mechanism is applied to reinforce features in key regions. Finally, the fused features are downsampled using a 3×3 convolution with a stride of 3, and the output preserves the enhanced feature representations with dimensions $C \times H \times W$. This module dynamically assigns weights at the receptive field level, mitigating the limitations of fixed convolutional kernels while maintaining manageable computational overhead. It adaptively focuses on important regions and suppresses noise, thereby enhancing multi-scale feature representation in complex scenes. The computation of RFA can be formulated as shown in Equation (1):

$$\begin{aligned} F &= \text{Softmax}(g^{1\times1}(\text{AvgPool}(X))) \times \text{ReLU}(\text{Norm}(g^{k\times k}(X))) \\ &= A_{rf} \times F_{rf} \end{aligned} \tag{1}$$

where $g^{i\times i}$ denotes a group convolution of size $i \times i$, k indicates the convolutional kernel size, *Norm* refers to normalization, and X is the input feature map. F is computed through the multiplication of the attention map A_{rf} and the transformed receptive-field spatial feature F_{rf}. The feature maps obtained by RFA are non-overlapping receptive-field spatial features generated through reshaping, with their attention maps aggregating the feature information of each slider to avoid parameter sharing. In addition, the feature size increases by a factor of k after reshaping, so a $k \times k$ convolutional kernel with a stride of k is applied for feature extraction.

After completing the feature enhancement, the RFAConv mechanism was innovatively integrated into the C2f module of YOLOv8n. The resulting enhancement module retains the benefits of the original residual connections while optimizing feature processing through a partitioning strategy based on receptive-field sliders. This module decouples the input features into

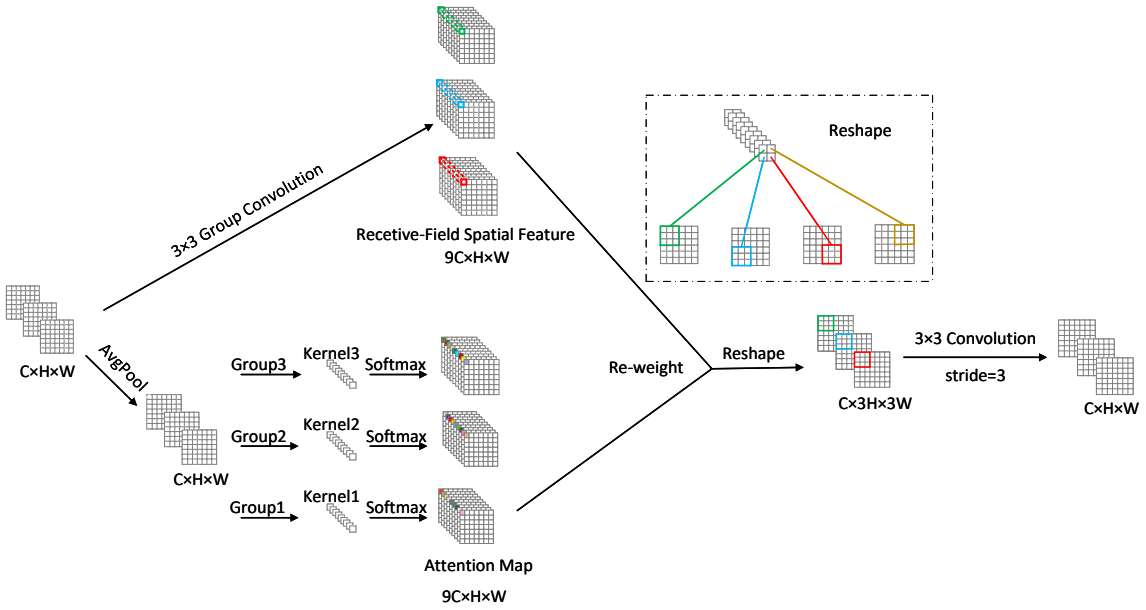


Fig. 3: The structure of RFAConv

multiple sets of receptive-field units for parallel processing. Furthermore, it combines with a deep feature extraction network to enhance the global semantic characterization capability while preserving local detail integrity. This design effectively mitigates the vanishing gradient problem by establishing a cross-level feature interaction mechanism and significantly improves the model's adaptability to variations in target scale through multi-scale feature fusion.

2.2.2 Optimization of SPPELAN structure for multi-scale spatial pyramids

SPPF extracts multi-scale features by performing three cascaded max-pooling operations, which reduces computational complexity. However, since each pooling operation takes the output of the previous one as its input, this cascading process can lead to excessive compression of feature information at certain scales, resulting in the loss of critical details. To address this issue, we introduced SPPELAN, which applies three independent max-pooling operations instead of repeated pooling. This architecture enables parallel feature extraction across different scales, thereby preventing information loss caused by repeated pooling. The architecture is illustrated in Fig. 4.

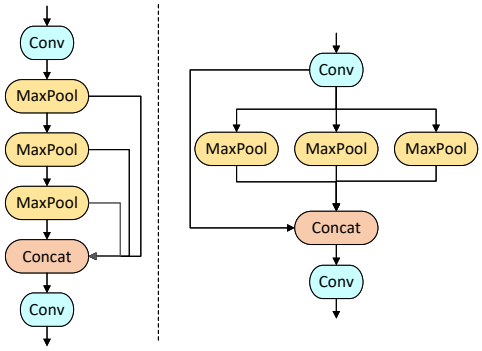
In contrast to SPPF, SPPELAN employs three separate max-pooling operations to effectively aggregate information from various feature maps. This approach not only improves the capture of multi-level semantic features from the input data but also ensures better information transfer. The final output of SPPELAN is obtained by concatenating the results from the three parallel max-pooling operations with the original convolutional feature map. The computational procedure is presented in Eqs. (2)–(4).

$$x = \mathrm{Conv}(X_{\mathrm{input}}) \tag{2}$$

$$y_{\max} = \mathrm{Cat}(\mathrm{Max}(x), \mathrm{Max}(x), \mathrm{Max}(x), x) \tag{3}$$

$$y_{\mathrm{out}} = \mathrm{Conv}(y_{\max}) \tag{4}$$

where *Conv* denotes the convolution operation, *Max* indicates the max-pooling operation, *Cat* denotes the concatenation operation along the channel dimension, X_{input} represents the input function, $y_{\max}$ is the output feature of max-pooling, and y_{out} is the final output feature.



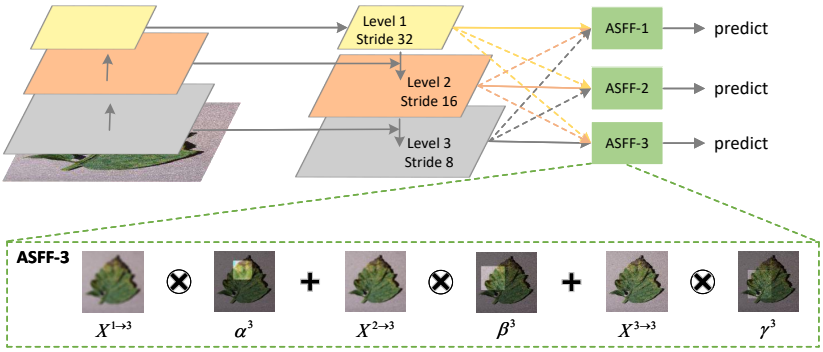
(a) SPPF (b) SPPELAN

Fig. 4: The structure comparison diagram of SPPF and SPPELAN: a demonstrates the SPPF structure, and b presents the SPPELAN structure

2.2.3 Detection head design based on ASFF

The original YOLOv8n detection head employs a fixed feature pyramid structure, achieving multi-scale feature fusion through simple upsampling and feature concatenation. However, this approach lacks the capability for cross-scale dynamic adjustment and does not effectively resolve the scale inconsistency problem within the feature pyramid. Consequently, it provides insufficient semantic information for small objects and loses spatial detail for large objects, leading to reduced detection accuracy. To address these limitations, we proposed the ASFF module and replaced the original detection head with an ASFF-based head. ASFF is a data-driven fusion strategy that dynamically adjusts the fusion weights of each scale feature map at every spatial location based on the input content, thereby suppressing conflicting information and enhancing feature representation. It adaptively selects the most relevant features according to object size and location, achieving a balance

Fig. 5: The structure of ASFF



between semantic richness and spatial detail, and significantly improving multi-scale detection performance.

As illustrated in Fig. 5, ASFF acts on three different scales of feature layers (Level 1, Level 2, and Level 3), corresponding to distinct spatial resolutions and semantic levels. ASFF-1, ASFF-2, and ASFF-3 each adaptively fuse feature maps from different levels at their respective scales. The process comprises two key steps:

(1) Feature Resizing: Feature maps from different scales are first resized to the same resolution. Up-sampling: For low-resolution feature maps, the channel dimension is first reduced using a 1×1 convolution and then upscaled via interpolation. Down-sampling: For a down-sampling ratio of $1/2$, a 3×3 convolution with a stride of 2 is applied to simultaneously reduce spatial resolution and adjust the channel dimension. When a larger down-sampling ratio (e.g., $1/4$) is required, max pooling with a stride of 2 is first applied, followed by a convolution with stride 2.

(2) Adaptive Fusion. At each scale l ($l \in \{1,2,3\}$), the aligned feature maps are adaptively fused. Specifically, feature maps from other scales are first realigned to match the spatial resolution of scale l, and then fused through a weighted fusion process using learned weights. The fused features at the l-th layer are formulated as shown in Equation (5).

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot X_{ij}^{1\to l} + \beta_{ij}^{l} \cdot X_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot X_{ij}^{3\to l} \tag{5}$$

where $X_{ij}^{n\to l}$ represents the feature vector at position (i,j) on the feature map aligned from layer n to scale l. y_{ij}^{l} denotes the feature vector of the (i,j)-th channel in the feature map y^{l} after weighted fusion. α_{ij}^{l}, β_{ij}^{l}, and γ_{ij}^{l} are adaptively learned weights representing the importance of the three different scales of the feature map for layer l. α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} can be scalar variables and are consistent across all channels. By enforcing $\alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1$ [31] and ensuring that the values of α_{ij}^{l}, β_{ij}^{l}, and γ_{ij}^{l} range between $[0,1]$, Equation (6) is defined as follows:

$$\alpha_{ij}^{l} = \frac{e^{\lambda_{\alpha_{ij}}^{l}}}{e^{\lambda_{\alpha_{ij}}^{l}} + e^{\lambda_{\beta_{ij}}^{l}} + e^{\lambda_{\gamma_{ij}}^{l}}} \tag{6}$$

where the weights are computed by the Softmax function. $\lambda_{\alpha_{ij}}^{l}$, $\lambda_{\beta_{ij}}^{l}$ and $\lambda_{\gamma_{ij}}^{l}$ are the key parameters used to control the magnitude of the weights. Through 1×1 convolutional layers, the weight scalar maps λ_{α}^{l}, λ_{β}^{l} and λ_{γ}^{l} are learned from $X_{1\to l}$, $X_{2\to l}$ and $X_{3\to l}$, respectively, and are optimized using standard backpropagation.

Using the adaptive aggregation approach described above, features from all layers are adaptively fused at each scale. The resulting output feature maps $\{y_1, y_2, y_3\}$ retain the most relevant information from multiple scales while effectively suppressing conflicting information. These fused feature maps can be directly utilized for object detection tasks, and subsequent processing follows the original YOLOv8n detection pipeline.

3 Experiment and analysis

3.1 Tomato leaf diseases and pests dataset

The dataset employed in this study was obtained from the Tomato Leaf Diseases and Pests dataset on the Roboflow platform, comprising four categories: late blight, septoria leaf spot, leaf miner infestation, and healthy leaves. Representative samples are shown in Fig. 6. Images were collected from both natural farmland and laboratory environments. Field samples were captured under varying illumination (morning, noon, and afternoon) and weather conditions (sunny, cloudy, and rainy), whereas laboratory samples were acquired under controlled lighting with simple backgrounds. Leaf morphologies included intact, occluded, and overlapping cases, enhancing dataset diversity and supporting model generalization under diverse conditions. All images were manually annotated using the LabelImg tool. The dataset was initially partitioned into training, validation, and test sets. To further improve sample diversity and model generalization, data augmentation—including horizontal flipping, brightness adjustment, fog simulation, and random cropping—was applied exclusively to the training set. Brightness adjustment involved randomly scaling contrast by a factor ranging from 0.8 to 1.2 and shifting brightness by an offset between -20 and 20. Fog simulation was performed by adding a haze effect with a density coefficient randomly selected from the interval $[0.03, 0.08]$. These augmentations effectively increased the number of training samples and enhanced model robustness. Following augmentation, the total dataset comprised $5,573$ images, with the training, validation, and test sets split in a ratio of $7:1.5:1.5$. The detailed data distribution is summarized in Table 1.

Table 1. Distribution of number of images of tomato leaves

Disease Category	Training Set	Test Set	Validation Set
Late Blight	1493	336	350
Septoria Leaf Spot	1147	228	253
Leaf Miner	3156	643	742
Healthy	5575	856	1255

Fig. 6: The sample images of various types of tomato leaves

(a) Late Blight (b) Septoria Leaf Spot (c) Leaf Miner (d) Healthy

3.2 Evaluation metrics

To objectively and comprehensively evaluate the detection model's performance on the test set, the following metrics were used in this study: Precision (P), Recall (R), mean Average Precision (mAP), F1-score, Giga Floating-point Operations per Second (GFLOPs), and the number of parameters. P represents the proportion of true positives among all predicted positives, while R represents the proportion of true positives among all actual positives. The F1-score was employed as an evaluation metric to address the issue of class imbalance by providing a harmonic mean of P and R, thereby ensuring a balanced assessment of the model's predictive performance. The specific calculation formulas are provided in Equations (7)–(9).

$$P = \frac{TP}{TP+FP} \times 100\% \tag{7}$$

$$R = \frac{TP}{TP+FN} \times 100\% \tag{8}$$

$$F1-\text{score} = 2 \times \frac{P \times R}{P+R} \times 100\% \tag{9}$$

GFLOPs are used to measure computational efficiency. The number of parameters reflects the model's complexity and computational cost. mAP is calculated as the mean of average precision scores across all categories and is used to evaluate the overall detection performance of the model. The calculation for Equation (10) is shown below:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{10}$$

3.3 Implementation details

The experimental setup was equipped with an Intel i7-12700H CPU, a 64-bit Windows 11 operating system, and an NVIDIA RTX 3050 Ti GPU. Additionally, the experiments were conducted on a GPU server featuring an NVIDIA Tesla A40 48GB 300W GPU. The algorithms were implemented using the PyTorch deep learning framework in Python. RSA-YOLO was trained on the training dataset with hyperparameters set as follows: a batch size of 8, 300 training epochs, and an initial learning rate of 0.01. The Stochastic Gradient Descent (SGD) optimizer was employed during training.

3.4 Comparison of different attention

To evaluate the performance of the RFA mechanism, YOLOv8n was selected as the baseline model. Comparative experiments were conducted with the Convolutional Block Attention Module (CBAM) [32], the Similarity-based Attention Module (SimAM) [33], Coordinate Attention (CA), and the Squeeze-and-Excitation (SE) module, assessing their parameter count, computational cost, and detection performance. The results are summarized in Table 2.

The experimental results indicate that RFA-YOLOv8 achieves the highest overall performance, with an mAP@0.5 of 79.3% and an F1-score of 78.5%, outperforming all other attention mechanisms. In contrast, SimAM-YOLOv8 exhibits a decline in performance, with the mAP@0.5 and F1-score decreasing to 77.8% and 77.1%, respectively. Although CBAM-YOLOv8 and CA-YOLOv8 show modest improvements in mAP@0.5 compared with the baseline model, their performance remains inferior to RFA-YOLOv8, with precision limited to 84%. Moreover, while CBAM-YOLOv8, CA-YOLOv8, and SE-YOLOv8 maintain relatively low computational costs, their mAP@0.5 values are still lower than that of RFA-YOLOv8. Overall, RFA-YOLOv8 achieves a more favorable trade-off between detection accuracy and computational efficiency, representing the most balanced model among the evaluated attention mechanisms.

Table 2. Comparative experiments on various attentional mechanisms

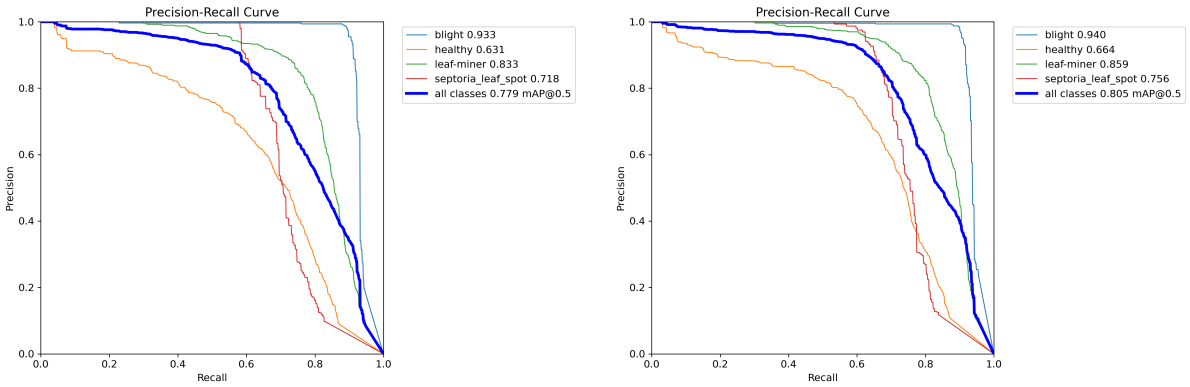
Model	P/%	mAP@0.5/%	F1-score/%	GFLOPs	Params/M
RFA-YOLOv8	85.6	79.3	78.5	8.7	3.07
SimAM-YOLOv8	85.3	77.8	77.1	8.9	3.15
CBAM-YOLOv8	84.0	78.0	77.4	8.3	3.11
CA-YOLOv8	84.0	78.4	77.9	8.2	3.02
SE-YOLOv8	85.3	78.1	77.9	8.2	3.01

3.5 Ablation experiments

To validate the proposed RSA-YOLO model for detecting tomato leaf diseases and pests, ablation experiments were conducted using the original YOLOv8n as the baseline, focusing on three key components: the RFA, ASFF, and SPPELAN modules. Evaluation metrics included precision, recall, mAP@0.5, F1-score, GFLOPs, and the number of parameters. The results for different combinations of these modules are summarized in Table 3, demonstrating that each contributed to improving detection performance to varying extents. Compared with the baseline, the introduction of the RFA module increased recall by 2% and improved both mAP@0.5 and F1-score by 1.4%, attributable to its dynamic receptive-field mechanism, which significantly enhanced the model's ability to localize disease and pest targets. Integration of the ASFF module alone yielded precision, recall, mAP@0.5, and F1-score values of 85.4%, 71.9%, 78.2%, and 78.1%, respectively; its multi-scale feature fusion capability effectively enhanced the extraction of subtle features. When the SPPELAN module was applied independently, precision rose to 86.0% and mAP@0.5 to 78.4%, indicating that its refined feature-extraction architecture improved the representation of core target features. For combined configurations, integrating RFA with ASFF achieved a recall of 73.5%, an mAP@0.5 of 79.7%, and an F1-score of 79.1%, illustrating that these modules complemented each other in optimizing different stages of object detection—feature fusion and localization. Similarly, combining RFA with SPPELAN yielded a precision of 85.8%, a recall of 71.7%, an mAP@0.5 of 78.8%,

Table 3: Ablation experiments on each improved module

RFA	ASFF	SPPELAN	P/%	R/%	mAP@0.5/%	F1-score/%	GFLOPs	Params/M
			85.2	70.4	77.9	77.1	8.2	3.01
✓			85.6	72.4	79.3	78.5	8.7	3.07
	✓		85.4	71.9	78.2	78.1	10.4	4.38
		✓	86	71.5	78.4	78.1	8.3	3.17
✓	✓		85.5	73.5	79.7	79.1	10.8	4.44
✓		✓	85.8	71.7	78.8	78.1	8.8	3.23
	✓	✓	86.5	73.2	79.2	79.3	10.5	4.54
✓	✓	✓	86.8	73.3	80.5	79.5	11	4.61



(a) YOLOv8 (b) RSA-YOLO

Fig. 7: Comparison of Precision–Recall performance between YOLOv8 and RSA-YOLO

and an F1-score of 78.1%. Although this combination underperformed relative to the RFA–ASFF and ASFF–SPPELAN variants, it still confirmed the positive contributions of RFA and SPPELAN in enhancing feature localization and representation, respectively. The ASFF–SPPELAN combination delivered the highest precision among dual-module variants (86.5%) and the second-highest overall F1-score (79.3%), while maintaining strong recall (73.2%) and mAP@0.5 (79.2%), demonstrating effective synergistic optimization of feature fusion and recognition. Overall, the complete model incorporating RFA, ASFF, and SPPELAN simultaneously achieved the best comprehensive performance. Compared with the baseline, precision increased by 1.6% to 86.8%, recall by 2.9% to 73.3%, mAP@0.5 by 2.6% to 80.5%, and F1-score by 2.4% to 79.5%. These results strongly support the effectiveness of the proposed module combinations in improving the accuracy and robustness of tomato leaf disease and pest detection. Furthermore, although the GFLOPs and parameter count of the complete model increased, this computational cost was justified by the substantial performance gains.

Fig. 7 presents the precision–recall (PR) curves of the baseline YOLOv8 model and the proposed RSA-YOLO on the tomato leaf disease and pest dataset. Overall, RSA-YOLO outperforms YOLOv8 across most recall levels, with an increased area under the PR curve, indicating improved accuracy and robustness for detecting different disease and pest types. This result further confirms the effectiveness of the proposed enhancement modules.

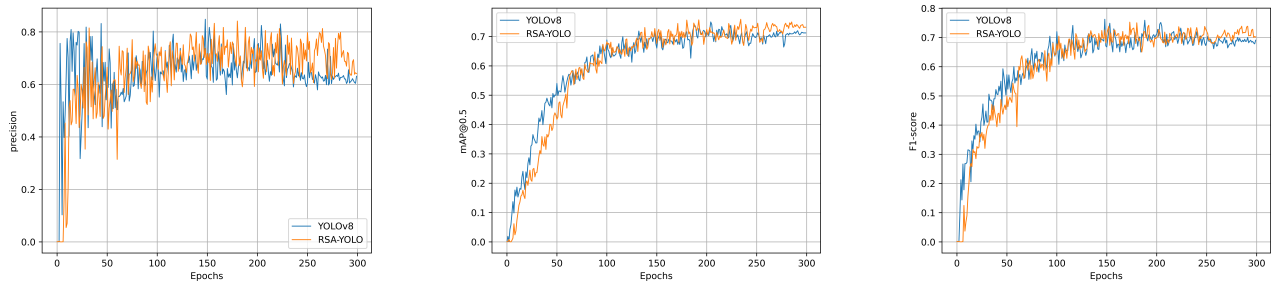
3.6 Analysis of experimental results on different datasets

To further evaluate the performance and generalization capability of the proposed RSA-YOLO model, we conducted additional experiments on an external dataset, referred to as Dataset I. This dataset, obtained from the Kaggle platform (Tomato Leaf Diseases Detection Computer Vision), contains seven categories, including several not present in the original dataset. Table 4 summarizes the evaluation results in terms of detection performance and computational cost, including P, mAP@0.5, F1-score, GFLOPs, and model parameter count. Compared with the baseline YOLOv8, RSA-YOLO improved P, mAP@0.5, and F1-score by 5.2%, 1.4%, and 2%, respectively. Although the computational cost (GFLOPs) increased from 8.2 to 11.0 and the number of parameters grew from 3.01 M to 4.61 M, the observed improvements demonstrate the effectiveness of the proposed modifications. These results indicate that RSA-YOLO not only improves detection performance for small or challenging objects but also demonstrates robust generalization to novel categories and previously unseen samples, confirming its adaptability under diverse data conditions.

Table 4. Comparison of Dataset I before and after improvement

Data I	P/%	mAP@0.5/%	F1-score/%	GFLOPs	Params/M
YOLOv8	65.2	72.7	70.9	8.2	3.01
RSA-YOLO (ours)	70.4	74.1	72.9	11.0	4.61

As shown in Fig. 8, the training curves on Dataset I demonstrate that RSA-YOLO consistently outperforms the baseline YOLOv8 across multiple evaluation metrics. In terms of precision, RSA-YOLO exhibits a more stable upward trend after the initial fluctuations, indicating stronger robustness in distinguishing positive samples. For mAP@0.5, both models improve rapidly during the early epochs; however, RSA-YOLO reaches a higher convergence level and maintains a more stable plateau in the later stages, suggesting superior overall detection capability. With respect to the F1-score, although the two models follow similar trajectories, RSA-YOLO achieves slightly higher values across most epochs, reflecting a better balance between



(a) P (b) mAP@0.5 (c) F1-score

Fig. 8: Comparison of model training results on Dataset I

precision and recall. Collectively, these results confirm that the proposed improvements effectively enhance the model's generalization performance on a previously unseen dataset.

3.7 Comparison of different algorithms

To validate the effectiveness of the proposed RSA-YOLO model, we conducted comparative experiments with several mainstream object detection algorithms under identical conditions, including YOLOv3-tiny, YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv11, YOLOv12 and YOLOv13. All models were trained on the tomato leaf pest and disease dataset using uniform settings and evaluated on the corresponding test set. The results are summarized in Table 5.

Table 5 shows that RSA-YOLO achieves superior performance across the main evaluation metrics, with a precision of 86.8%, mAP@0.5 of 80.5%, and an F1-score of 79.5%, outperforming the other models. Compared with the lightweight YOLOv13 (2.46 M parameters), RSA-YOLO provides higher detection accuracy while maintaining moderate computational costs (11.0 GFLOPs, 4.61 M parameters). In addition, compared with YOLOv9, which incurs substantially higher computational overhead (118.2 GFLOPs, 31.4 M parameters), RSA-YOLO attains a more favorable trade-off between accuracy and complexity. These findings suggest that RSA-YOLO not only improves detection accuracy over existing mainstream models but also offers a balanced compromise between model size and computational efficiency, underscoring its potential for practical deployment in agricultural applications.

Table 5. Comparative experiments of various mainstream target detection models

Model	P	mAP@0.5	F1	GFLOPs	Params
YOLOv3-tiny	79.5	73.7	73.5	18.9	12.13
YOLOv5	82.3	76.4	76.2	7.3	4.13
YOLOv6	86.3	76	76	11.8	4.23
YOLOv8	85.2	77.9	77.1	8.2	3.01
YOLOv9	84.8	78.1	77.9	118.2	31.4
YOLOv11	84.8	78.1	77.9	6.4	2.50
YOLOv12	85.1	77	78.2	6.0	2.52
YOLOv13	85.9	78.6	78.5	6.3	2.46
RSA-YOLO (ours)	86.8	80.5	79.5	11.0	4.61

Using detection precision, recall, mAP@0.5, and mAP@0.5:0.95 as evaluation metrics, the performance curves for the different models on the test set were plotted, as shown in Fig. 9. During the initial phase of training, the curves for all evaluation metrics across all models increased rapidly, which indicated effective learning of data features. As training progressed, RSA-YOLO demonstrated significant advantages in detection precision, mAP@0.5, and mAP@0.5:0.95, all of which showed considerably higher values than the other models. Notably, the recall curve of RSA-YOLO remained high, reflecting its ability to detect more true objects and substantially reduce the miss rate. Furthermore, RSA-YOLO not only converged faster but also exhibited smaller fluctuations in the evaluation curves of metrics during the later stages of training, indicating improved generalization and training stability.

3.8 Visualization and analysis of results

To clearly demonstrate the effectiveness of RSA-YOLO, images of different tomato leaf diseases and pests were used as input. These images were detected using both the original YOLOv8 and RSA-YOLO models, and the corresponding visualization results are presented in Fig. 10.

As illustrated in the figure, compared with the original YOLOv8 model, RSA-YOLO not only generates bounding boxes that more accurately correspond to the actual target sizes, but also substantially reduces both missed detections and false positives, indicating improved localization accuracy and enhanced detection reliability for tomato leaf diseases and pests.

4 Conclusion

In this study, to address the challenges posed by the varying sizes of tomato leaf diseases and pests, as well as the complex backgrounds, we proposed a novel model named RSA-YOLO. By integrating the RFA module into the backbone of YOLOv8, the model more effectively captured fine-grained details of small objects and semantic features of larger ones, thereby enhancing multi-scale feature extraction. To further enhance the representational power of the feature pyramid, the original SPPF module in the backbone was replaced with the SPPELAN module, resulting in improved efficiency and accuracy in multi-scale feature fusion. Additionally, the ASFF module was incorporated into the detection head to ensure scale invariance of the enhanced features and further boost detection accuracy. Experimental results demonstrated that RSA-YOLO significantly outperformed the original YOLOv8, achieving a 1.6% increase in precision, a 2.6% improvement in mAP@0.5, and a 2.4% gain in F1-score. Compared to YOLOv3-tiny, YOLOv5, YOLOv6, YOLOv9, YOLOv11, YOLOv12 and YOLOv13, RSA-YOLO achieved higher mAP@0.5 values by 6.8%, 4.1%, 4.5%, 2.4%, 2.6%, 3.5%, and 1.9%, respectively. These results indicate that RSA-YOLO provides a more effective technical solution for the early detection of tomato leaf diseases and pests.

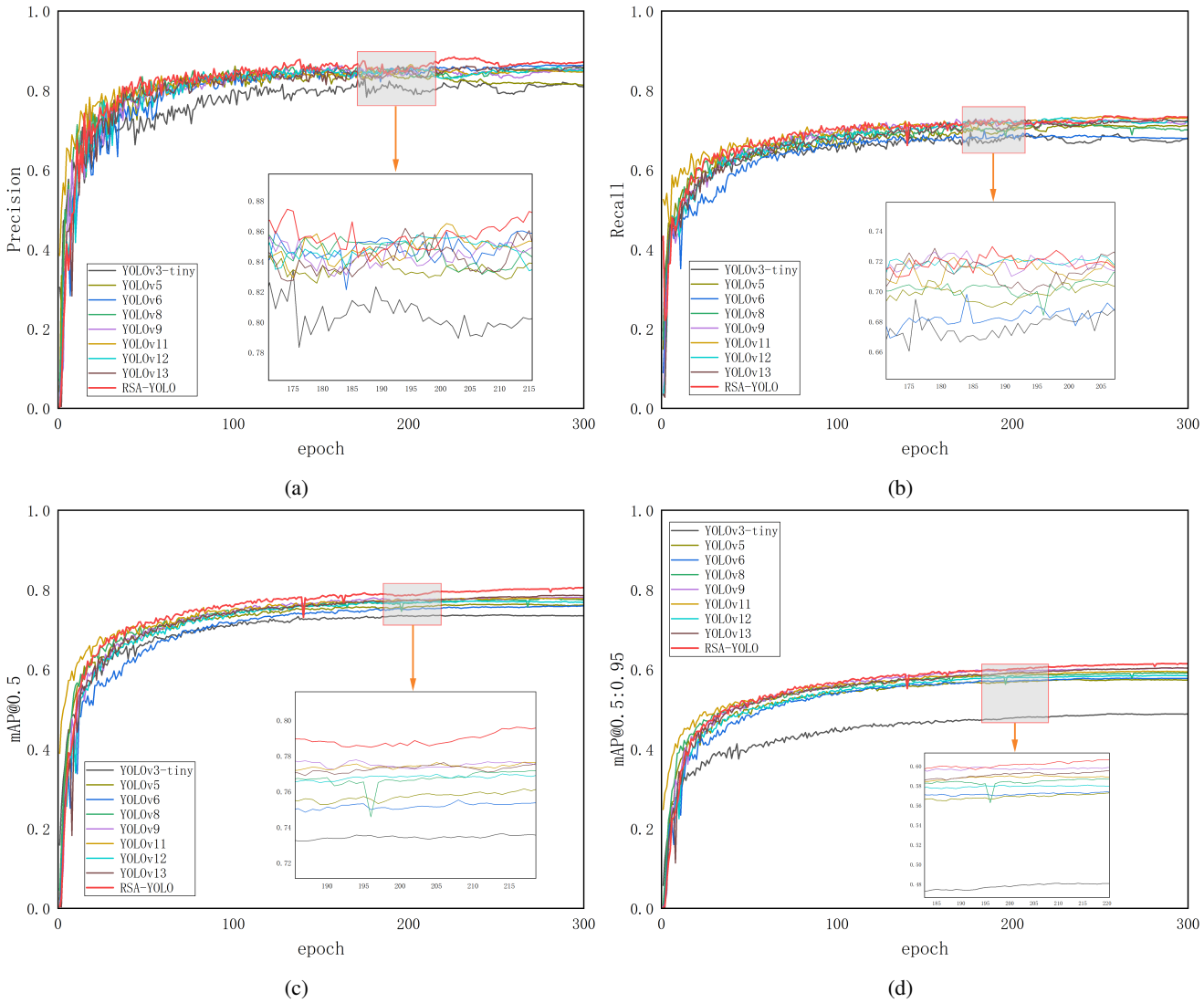


Fig. 9: The line plots of Precision, Recall, mAP@0.5 and mAP@0.5: 0.95 for different models: a Precision; b Recall; c mAP@0.5; d mAP@0.5: 0.95

Although the RSA-YOLO model achieves a significant improvement in detection accuracy and effectively alleviates the challenges posed by multi-scale blurred targets, background interference, and illumination variations, its parameter count is slightly higher than that of the baseline model. Nevertheless, this increase remains within a reasonable range and has little impact on overall performance. Meanwhile, although missed detections have been substantially reduced, some healthy leaves may still be overlooked, as illustrated in Fig. 10. These cases typically occur when leaves exhibit abnormal morphology, severe overlap, or strong similarity to the surrounding environment, revealing the model's limitations under extreme and complex conditions. Future work will focus on further enhancing deployment performance by applying model compression techniques such as pruning and quantization, thereby reducing model size while maintaining accuracy. These efforts aim to improve efficiency and practicality, facilitating real-time disease detection in field applications on smartphones or portable agricultural devices.

Author contribution. Conceptualization, Q.D., F.W.; methodology, Q.D., F.W., and M.T.; software, Q.D., F.W.; investigation, Q.D., F.W.; writing,original draft preparation, F.W.; writing,review and editing, Q.D., M.T., and W.W.; visualization, Q.D.; All authors reviewed the results and approved the final manuscript.

Funding. Not applicable.

Data availability. The data generated in this study are not publicly available but are available upon request from the corresponding author.

Declarations

Conflict of interest. The authors declare no competing interests.

References

[1] Xu, Z., Shou, W., Huang, K., et al.: The current situation and trend of tomato cultivation in China. *Acta Physiol. Plant.* **22**, 379–382(2000) https://doi.org/10.1007/s11738-000-0061-y

[2] Fuentes, A., Yoon, S., Youngki, H., Lee, Y., Park, D. S.: Characteristics of tomato plant diseases—a study for tomato plant disease identification. In: *Proc Int Symp Inf Technol Converg*, vol. 1, pp. 226-231(2016)

[3] Wang, X., Liu, J., Zhu, X.: Early real-time detection algorithm of tomato diseases and pests in the natural environment. *Plant Methods* **17**, 43 (2021) https://doi.org/10.1186/s13007-021-00745-2

[4] Yao, J., Tran, S.N., Sawyer, S., et al.: Machine learning for leaf disease classification: data, techniques and applications. *Artif Intell Rev* **56**(Suppl 3), 3571–3616(2023) https://doi.org/10.1007/s10462-023-10610-4

Fig. 10: Comparison between YOLOv8n and RSA-YOLO for tomato leaf disease and pest detection results

[5] Ullah, N., Khan, J.A., Alharbi, L.A., et al.: An efficient approach for crops pests recognition and classification based on novel deeppestnet deep learning model. *IEEE Access* **10**, 73019–73032(2022) https://doi.org/10.1109/ACCESS.2022.3189676

[6] Liu, J., Wang, X.: Plant diseases and pests detection based on deep learning: a review. *Plant Methods* **17**, 22(2021) https://doi.org/10.1186/s13007-021-00722-9

[7] Madhav, M.U., Jyothi, D.N., Kalyani, P.: Prediction of pesticides and identification of diseases in fruits using Support Vector Machine (SVM) and IoT. In: *AIP Conference Proceedings*, vol. 2407, no. 1(2021) https://doi.org/10.1063/5.0074595

[8] Bera, A., Bhattacharjee, D., Krejcar, O.: PND-Net: plant nutrition deficiency and disease classification using graph convolutional network. *Sci. Rep.* **14**, 15537(2024) https://doi.org/10.1038/s41598-024-66543-7

[9] Prabu, M., Chelliah, B.J.: An intelligent approach using boosted support vector machine based arithmetic optimization algorithm for accurate detection of plant leaf disease. *Patt. Anal. Appl.* **26**, 367–379(2023) https://doi.org/10.1007/s10044-022-01086-z

[10] Wu, J., Abolghasemi, V., Anisi, M.H., et al.: Strawberry disease detection through an advanced squeeze-and-excitation deep learning model. *IEEE Transactions on AgriFood Electronics*(2024)

[11] Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587(2014)

[12] Ren, S., He, K., Girshick, R., et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28**(2015)

[13] Zhang, H., Wang, Y., Dayoub, F., et al.: Varifocalnet: An iou-aware dense object detector. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8514–8523(2021)

[14] Hussain, M.: Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE Access* **12**, 42816–42833(2024)

[15] Ge, Z., Liu, S., Wang, F., et al.: YOLOX: Exceeding YOLO series in 2021.(2021) https://doi.org/10.48550/arXiv.2107.08430

[16] Sun, H., Xu, H., Liu, B., He, D., He, J., Zhang, H., Geng, N.: MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. *Computers and Electronics in Agriculture* **189**, 106379(2021) https://doi.org/10.1016/j.compag.2021.106379

[17] Ullah, Z., Alsubaie, N., Jamjoom, M., Alajmani, S.H., Saleem, F.: EffiMob-Net: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images. *Agriculture* **13**, 737(2023) https://doi.org/10.3390/agriculture13030737

[18] Luo, D., Xue, Y., Deng, X., et al.: Citrus diseases and pests detection model based on self-attention YOLOv8. *IEEE Access*, **11**, 139872–139881(2023) https://doi.org/10.1109/ACCESS.2023.3340148

[19] Wang, X., Liu, J.: Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment. *Sci. Rep.* **14**, 4261(2024) https://doi.org/10.1038/s41598-024-54540-9

[20] Lu, Y., Yu, J., Zhu, X. et al.: YOLOv8-Rice: a rice leaf disease detection model based on YOLOv8. *Paddy and Water Environment* **22**, 695–710(2024) https://doi.org/10.1007/s10333-024-00990-w

[21] Qiang, J., Liu, W., Li, X., Guan, P., Du, Y., Liu, B., Xiao, G.: Detection of citrus pests in double backbone network based on single shot multibox detector. *Computers and Electronics in Agriculture* **212**, 108158(2023) https://doi.org/10.1016/j.compag.2023.108158

[22] Yaseen, M.: What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector.(2024)

[23] Zhang, X., Liu, C., Yang, D., et al.: RFAConv: Innovating spatial attention and standard convolutional operation.(2023) https://doi.org/10.48550/arXiv.2304.03198

[24] Wang, C.Y., Yeh, I.H., Mark, Liao, H.Y.: Yolov9: Learning what you want to learn using programmable gradient information. In: *European conference on computer vision*, pp. 1–21(2024) https://doi.org/10.1007/978-3-031-72751-1_1

[25] Liu, S., Huang, D., Wang, Y.: Learning spatial fusion for single-shot object detection.(2019) https://doi.org/10.48550/arXiv.1911.09516

[26] Wang, C.Y., Liao, H.Y.M., Wu, Y.H., et al.: CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390-391(2020)

[27] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection.(2020) https://doi.org/10.48550/arXiv.2004.10934

[28] Li, Y., Xia, Y., Zheng, G., et al.: YOLO-RWY: A Novel Runway Detection Model for Vision-Based Autonomous Landing of Fixed-Wing Unmanned Aerial Vehicles. *Drones* **8**(10), 571(2024) https://doi.org/10.3390/drones8100571

[29] Li, X., Lv, C.Q., Wang, W.H., et al.: Generalized focal loss: Towards efficient representation learning for dense object detection. *IEEE transactions on pattern analysis and machine intelligence* **45**(3), 3139-3153(2022) https://doi.org/10.1109/TPAMI.2022.3180392

[30] Zheng, Z.H., Wang, P., Ren, D.W., et al.: Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE transactions on cybernetics* **52**(8), 8574-8586(2021) https://doi.org/10.1109/TCYB.2021.3095305

[31] Wang, G., Wang, K., Lin, L.: Adaptively connected neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1781-1790(2019)

[32] Woo, S., Park, J., Lee, J.Y., et al.: Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3-19(2018)

[33] Yang, L., Zhang, R.Y., Li, L., et al.: Simam: A simple, parameter-free attention module for convolutional neural networks. In *International conference on machine learning*, pp. 11863-11874(2021)