Google Scholar Crossref doi scopus

Impact factor 6.2

Geoscience Journal

ISSN:1000-8527

Indexing:

- Scopus
- Google Scholar
- DOI, Zenodo
- Open Access

www.geoscience.ac

Registered

Optimization Study of a Tomato Leaf Disease Recognition Model Based on Multi-Strategy Improvement of YOLOv12

Zhang Hansheng, Jiang Shanchao*
School of Electrical Engineering, Yancheng Institute of Technology Yancheng Institute of Technology, 224051Yancheng, Jiangsu, China

ABSTRACT: *Tomato leaf diseases are critical factors affecting tomato yield and quality, yet their accurate identification remains challenging in complex field environments. This study aims to develop an improved YOLOv12 model for robust tomato leaf disease recognition. First, we constructed and annotated a comprehensive multi-environment dataset covering 10 common leaf diseases. To address YOLOv12's limitations, we introduced three key innovations: (1) Incorporating a Self-Attention (SE) module to enhance disease feature representation in cluttered backgrounds; (2) Replacing standard convolutions with GhostConv to reduce computational load while preserving feature discriminative power; (3) Adopting a scale-adaptive WIoU_v2 loss function to optimize gradient distribution across varying data quality. Ablation studies confirm these improvements synergistically enhance the model's adaptability to varying disease scales and environmental conditions. The refined model achieves a 0.9% mAP@0.5 improvement and a 1.9% mAP@0.5:0.95 increase compared to the original model, while reducing computational demands. The proposed system achieves an optimal balance among detection accuracy, inference speed, and lightweight design, offering a novel solution for automated tomato disease recognition.*

KEYWORD: Tomato Leaf Diseases, Deep Learning, Object Detection, YOLOv12, Multi-Objective Optimization

[1]

[1] *Corresponding author

1. INTRODUCTION

Tomatoes, as one of the most widely cultivated crops globally, are highly valued for their significant economic importance and nutritional content, and have become an essential part of daily diets[1]. However, tomato crop yields and quality continue to be threatened by various leaf diseases, such as late blight, bacterial leaf spot, and leaf spot disease[2]. These diseases manifest through symptoms such as leaf discoloration, wilting, defoliation, and fruit rot, ultimately compromising plant health and productivity[3]. Current detection of tomato leaf diseases primarily relies on visual inspection by growers and agricultural experts. While this method provides preliminary diagnostic evidence, the results are often influenced by the subjective experience of evaluators and variations in observation conditions [4]. Therefore, there is an urgent need for an efficient method to detect tomato leaf diseases.

In recent years, with the advancement of machine learning and deep learning, agricultural image classification has become a cornerstone for applying these technologies to real-world challenges[5]. Deep learning technology, in particular, has been employed to detect leaf diseases in plants[6]. Ananthi et al. proposed a lightweight Region Convolutional Neural Network (R-CNN) head model for detecting leaf diseases in tomato plants. They conducted experiments using multiple neural network architectures, including DenseNet, achieving impressive results in terms of accuracy, F1 score, recall, and precision[6]. SONG et al. proposed a lightweight real-time tomato detection and point-picking integrated network model (TDPPL-Net) based on YOLOv5, aiming to address the issues of large model size and excessive network parameters in existing object detection models. Experimental results demonstrate that TDPPL-Net reduces parameter count by 59.84% compared to the original YOLOv5, with model size shrinking to just 40% of the original. It achieves a mAP of 93.36 and delivers real-time detection speeds of 31.41 FPS on IPC without GPU acceleration—representing a 170.31% improvement over YOLOv5[8].

With the advancement of deep learning, the YOLO series has gained widespread adoption due to its algorithms' speed, high accuracy, and ease of deployment. To improve tomato health detection, QUACH et al. proposed a tomato classification, detection, and counting system based on an explainable mobile network model utilizing the YoloV8 framework and Grad-Cam++. They also compared the effectiveness of MobileNet models for classifying tomato physiological states, achieving metrics exceeding 95% across all indicators[9]. Liu et al. proposed an LGC-YOLOv10 model, which builds upon YOLOv10 by integrating three modules—LSKA, Ghost, and CAMixing—into the new LGC-YOLOv10 architecture. This approach targets tomato ripeness detection. Testing revealed that the LGC-YOLOv10 model significantly improved mAP50-95 (B) values after 30 and 50 training epochs, achieving a 7.2% increase at 30 epochs and an 8.3% increase at 50 epochs. However, the YOLO series still exhibits challenges in tomato leaf disease recognition, including low accuracy for small targets, early-stage symptoms, and similar diseases[10].

Therefore, to address the limitations of the YOLO series in identifying tomato leaf diseases, this study selected the YOLOv12n model as the foundation for improvement. First, the SE attention module was introduced into the backbone network to enhance the model's ability to represent disease features in cluttered backgrounds. Second, GhostConv was employed

to replace standard convolutions in the neck network, reducing computational load while preserving feature discriminative power. Finally, a scale-adaptive WIoU_v2 loss function was adopted to optimize gradient distribution across data of varying quality.

2. MATERIALS AND METHODS

2.1 Material and data processing

Tomato leaf lesions, as a key characteristic of plant foliar diseases, exhibit morphological changes closely linked to environmental conditions. This study constructed a phenotypic image database of tomato leaf lesions by purchasing and collecting publicly available datasets online. Through rigorous image preprocessing and target annotation, a dataset comprising ten categories of tomato leaf diseases—including healthy tomato leaves—was established.

2.2 Data preprocessing and set construction

To enhance data quality and utilization efficiency, it is necessary to perform systematic preprocessing and dataset construction on the raw images collected, preparing them for deep learning tasks.

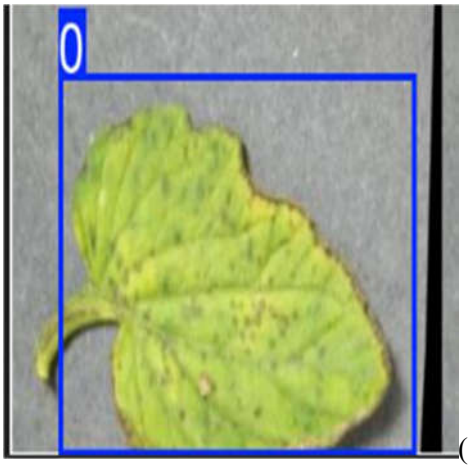
2.3 Image preprocessing process

The original image undergoes the following preprocessing processes: (a) Manual screening, eliminating low-quality images with cluttered backgrounds and shaky lenses, and obtaining effective images. (b) Geometric normalization, bilinear interpolation to scale the long and short edges according to the original aspect ratio.

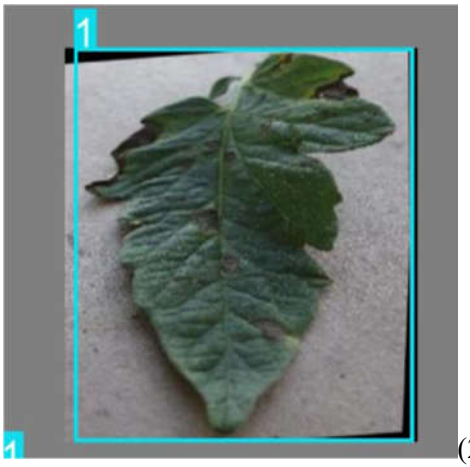
2.4 Data set construction

LabelImg is a commonly used tool in the field of annotation and detection[11][12][13]. Therefore, based on the preprocessed images, we performed manual bounding box annotations using the LabelImg tool. The annotation standards are shown in Figure 1:

(1): Bacterial_Spot
(2): Early_Blight
(3): Healthy
(4): Late_Blight
(5): Leaf_Mold
(6): Leaf_Miner
(7): Mosaic_Virus
(8): Septoria
(9): Spider_Mites
(10): Yellow_Leaf_Curl_Virus

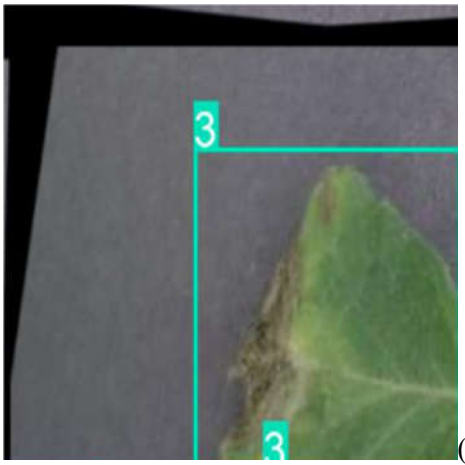


(1)



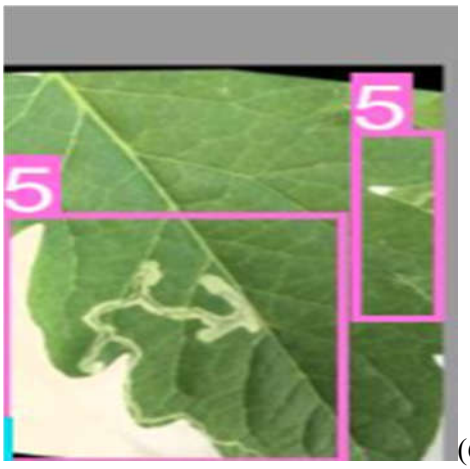
(2)

(3)

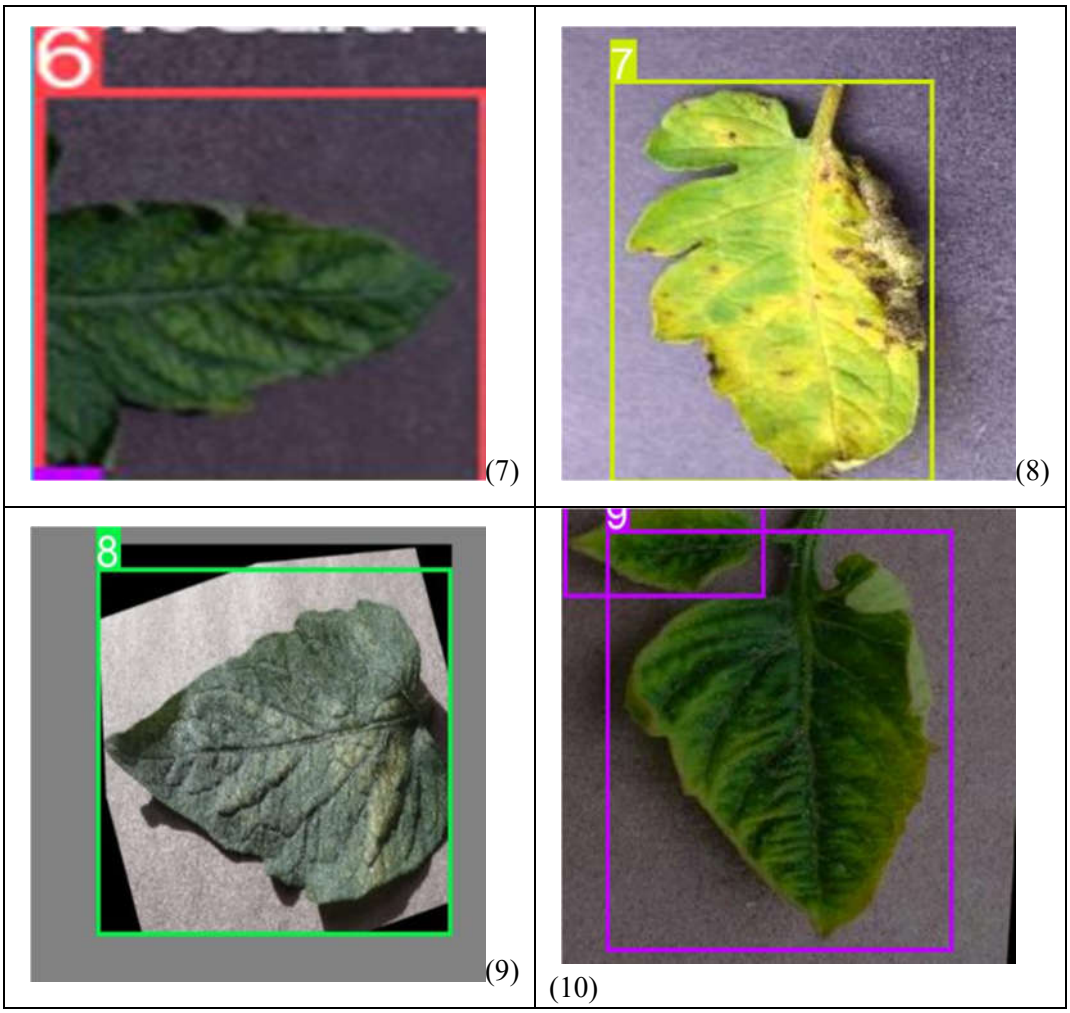


(4)

(5)



(6)



(7) (8) (9) (10)

Figure 1. Labeling diagram for the self-compiled tomato leaf disease dataset.

(1):Bacterial_Spot,(2):Early_Blight,(3):Healthy,(4):Late_Blight,(5):Leaf_Mold,(6):Leaf_Miner,(7):Mosaic_Virus,(8):Septoria,(9):Spider_Mites,(10): Yellow_Leaf_Curl_Virus.

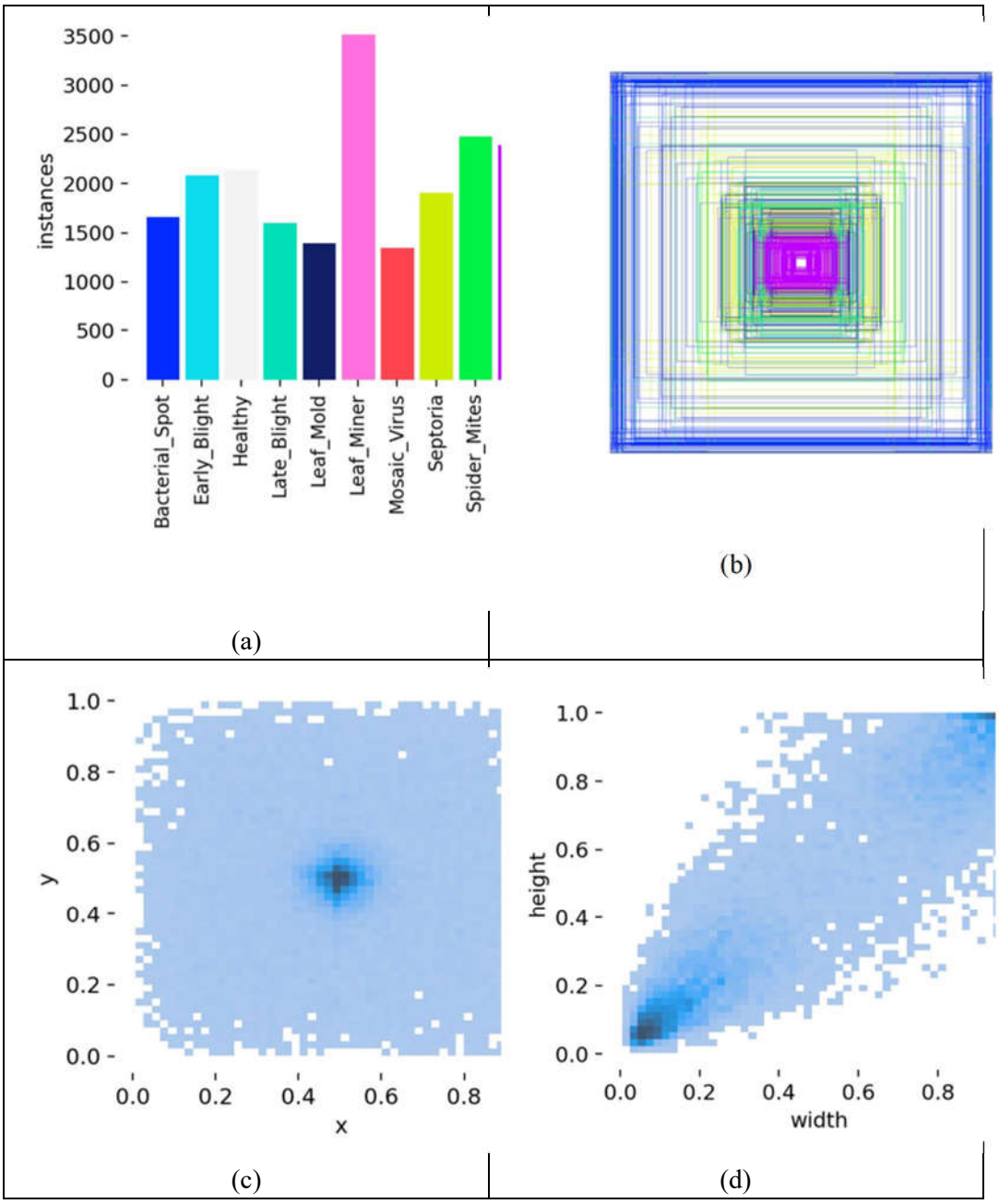


Figure 2. Distribution of labels in the self-made dataset.

(a) Label categories, (b) Object bounding box size, (c)Object bounding box centroid, (d) Corresponding width and height of object bounding boxes.

Figure 2 illustrates the distribution of labels across categories in the self-compiled tomato leaf disease dataset: (a) The bar chart shows the number of instances for each disease category, with “Leaf_Mold” having the highest count exceeding 3,500 instances; (b)The bounding box size distribution plot illustrates the dimensions of differently colored bounding boxes, indicating the presence of targets of various sizes within the dataset; (c)The centroid coordinate distribution plot reveals the spatial distribution of target centroids within images, showing a relatively concentrated distribution in the central region; (d) The width-height scatter plot depicts the relationship between target width and height, with the high-density area in the lower left corner indicating that small-sized targets are

relatively common in the dataset.

To accurately evaluate a model's generalization capability, the original dataset must be divided into mutually exclusive training (Train), validation (Val), and test (Test) sets. Images are allocated to the training, validation, and test sets in a 7:2:1 ratio.

3. Experimental environment

The experimental platform runs on a Windows 11 64-bit operating system, equipped with an AMD Ryzen 7 7735H CPU @ 3.2GHz, 24GB RAM, and an NVIDIA GeForce RTX 4060 graphics card @ 144Hz. Neural networks were trained within an Anaconda3 virtual environment using Python 3.11 with CUDA version 11.8. Specific environment parameters follow the official default settings for YOLOv12n.

3.1 Evaluation indicators

In the YOLO series of models, the following metrics are used to evaluate network performance: Precision (P), Recall (R), Average Precision (AP), and Mean Average Precision (mAP). The specific formulas for each metric are as follows:

$$P = \frac{TP}{TP+FP} \tag{1}$$

$$R = \frac{TP}{TP+FN} \tag{2}$$

$$AP = \int_0^1 PRdR \tag{3}$$

$$mAP = \frac{1}{N}\sum_{i=0}^{N} AP_i \tag{4}$$

In the above formula, TP (True Positive): A positive sample correctly predicted as positive by the model. TN (True Negative): A negative sample correctly predicted as negative by the model. FP (False Positive): A negative sample incorrectly predicted as positive by the model. FN (False Negative): A positive sample is incorrectly predicted as negative by the model.

Therefore, this experiment employs P, R mAP@0.5, and mAP@0.5:0.9 as model performance evaluation metrics. mAP@0.5 indicates the mAP value at an IoU threshold of 0.5, and mAP@0.5:0.95 represents the average mAP values as IoU increases from 0.5 to 0.95 in increments of 0.05. Larger mAP@0.5 and mAP@0.5:0.95 values indicate higher overall model accuracy.

3.2 YOLOv12 model

The YOLO series of algorithms is a classic single-stage object detection algorithm [14]. YOLOv12 is the twelfth iteration of the YOLO family released by Ultralytics, comprising four main components: the input layer, backbone network, neck network, and head output layer[15]. The network architecture of YOLOv12 is shown in Figure 3.

Based on the number of model parameters, the YOLOv12 models are categorized into five types: YOLOv12n, YOLOv12s, YOLOv12m, YOLOv12l, and YOLOv12x. Considering the requirement for real-time detection of tomato leaf diseases while ensuring

detection accuracy, this study selected the YOLOv12n model as the foundation for improvement. First, the SE attention module was introduced into the backbone network to enhance the model's ability to represent disease features in cluttered backgrounds. Second, GhostConv replaced standard convolutions in the neck network to reduce computational load while preserving feature discriminative power. Finally, a scale-adaptive WIoU_v2 loss function was adopted to optimize gradient distribution across data of varying quality.

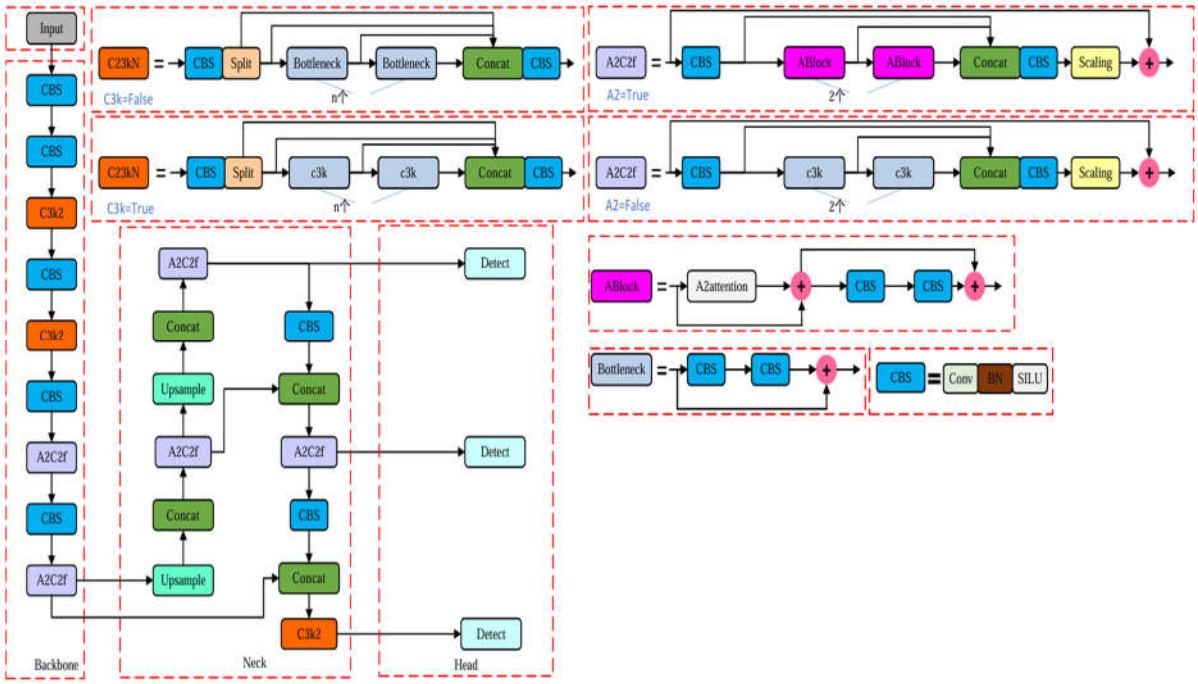


Figure 3. YOLOv12 network architecture

3.3 Backbone improvement: introduction of SE attention mechanism

As the feature extractor for object detection networks, the performance of the backbone directly determines the upper limit of detection capabilities. To enhance the adaptive representation ability of the YOLOv12 backbone network for tomato leaf diseases, this study introduces SE (Squeeze-and-Excitation) attention modules into various parts of its network backbone[16]. SE explicitly models dependencies between feature channels, adaptively adjusting the importance of different semantics, enabling the network backbone to focus on target-relevant regions and features.

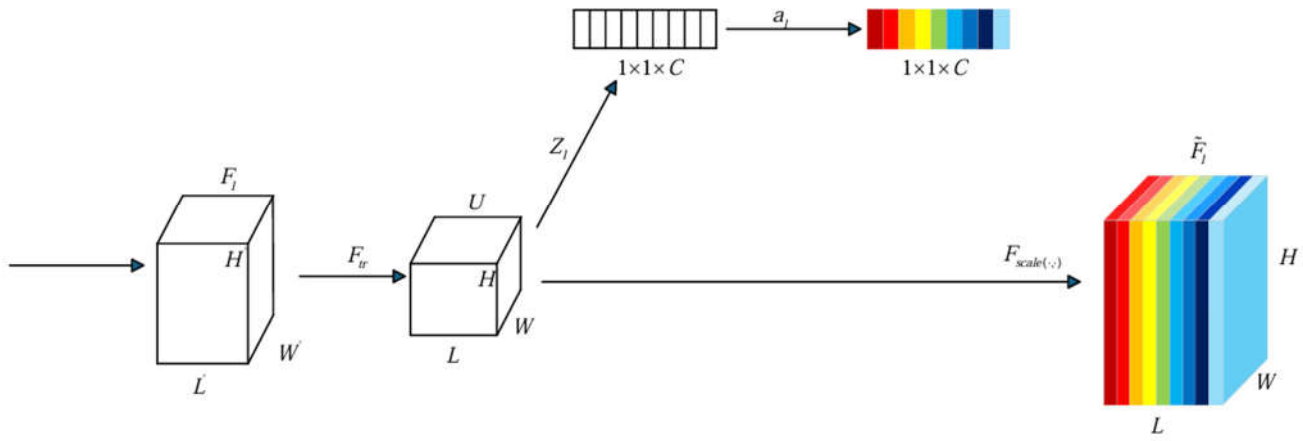


Figure 4. Schematic diagram of the structure of the SE attention module.

As shown in Figure 4, Ftr is regarded as a standard convolution operator, which transforms the input feature map F_l into l output feature maps U through convolution operations.Given the feature map $F_l \in R^{H_l \times W_l \times C_l}$of the first layer,The SE module first performs global average pooling in the spatial dimension to obtain the channel descriptor $Z_l \in R^{C_l}$（as shown in Equation (5)）.

$$z_l = \frac{1}{H_l W_l} \sum_{i=1}^{H_l} \sum_{j=1}^{W_l} F_l(i,j) \tag{5}$$

Where (i, j) represents spatial coordinates.z_l is a feature vector encompassing the global receptive field, integrating contextual information across the entire feature map. Subsequently, the self-attention mechanism applies nonlinear transformations to z_l through two fully connected layers and a Sigmoid function, generating channel attention weights $\alpha_l \in R^{C_l}$ (as shown in Equation (6)).

$$\alpha_l = \sigma(W_2 \delta(W_{1z_l})) \tag{6}$$

Where $W_1 \in R^{\frac{C_l}{r} \times C_l}$, $W_2 \in R^{C_l \times \frac{C_l}{r}}$ represent the weight matrices of the two fully connected layers, respectively. Δ denotes the ReLU function,σ denotes the Sigmoid function, and r is the scaling factor (typically set to 16). These two fully connected layers function as bottleneck layers, introducing nonlinearity while reducing the number of parameters. Finally, α_l undergoes a channel−wise multiplication with F_l to yield the attention-weighted feature map $\hat{F_l}$ as shown in Equation (7).

$$\hat{F_l}(i,j,c) = \alpha_l(c) \cdot F_l(i,j,c) \tag{7}$$

3.4 Locating the SE Embedding Position

Research integrating SE attention modules based on the YOLOv12 architecture demonstrates that their positional selection significantly impacts detection performance. We systematically evaluate three distinct attention mechanism layout schemes in Figure 5:

(a) SPPF Position in the Backbone Network

The SE module is integrated at the spatial pyramid feature output layer (Layer 9) of the backbone network, focusing on enhancing the representation capability of multi-scale features and improving cross-scale fusion efficiency.

(b) Neck C2f Module Optimization

Positioned at the output end of the critical C2f module connecting the detection head (layers 15/18/21), strategically placing the SE module within the feature propagation path effectively suppresses feature noise while amplifying discriminative feature responses.

(c) Hybrid Integration Approach

Combining the advantages of approaches (a) and (b), the SE attention mechanism is simultaneously integrated into the backbone network SPPF and multiple neck C2f modules, achieving comprehensive feature enhancement.

Through systematic comparative experiments (results shown in Table 1), we draw the following key conclusions: Based on comprehensive performance evaluation, we ultimately selected scheme (a) as the optimal SE integration position for YOLOv12. Adding the SE module at the end of the backbone network (layer 9) ensures computational efficiency while significantly enhancing feature quality. This configuration achieves the optimal balance of detection accuracy while maintaining model lightness.

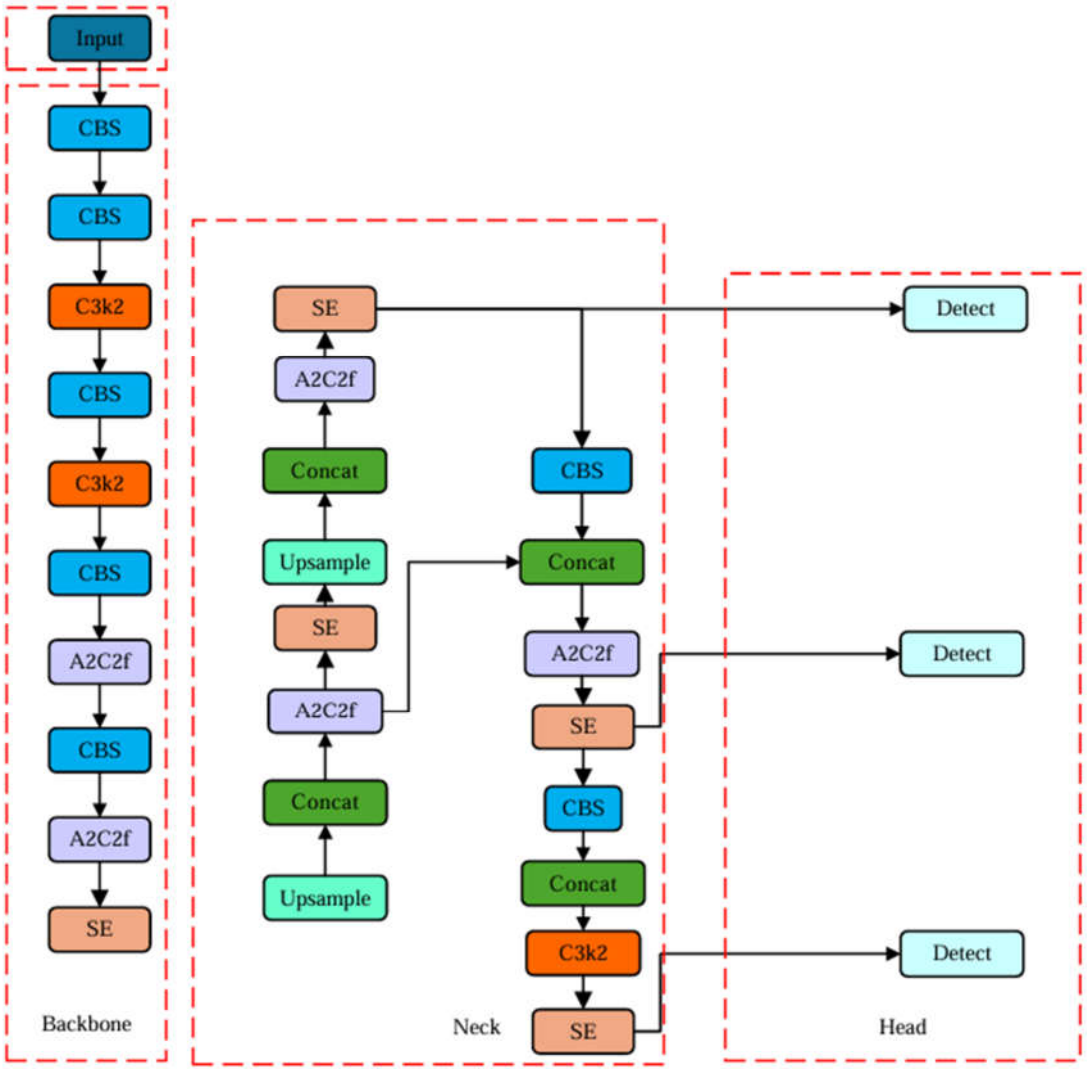


Figure 5. Location of attention mechanism additions.

Table 1. Experimental table of comparison of channels added by the attention mechanism.

Embedded Solution	mAP@0.5	mAP@0.5:0.95	Precision	Recall
Solution A	0.730	0.639	0.766	0.668
Solution B	0.709	0.623	0.740	0.667
Solution C	0.726	0.633	0.759	0.662
Origin	0.709	0.613	0.760	0.664

This study provides crucial design guidance for the optimal deployment of attention mechanisms within the YOLOv12 architecture, ensuring that the Supervision Enhancement (SE) module maximizes its feature enhancement capabilities.

3.5 Neck Improvement: Lightweight GhostConv Feature Fusion

YOLOv12 constructs a more robust feature pyramid through top-down multi-scale feature fusion, achieving complementary integration of high-level semantic information and low-level detail. However, the feature fusion process requires extensive convolutional operations, leading to redundant model parameters and reduced inference speed. Therefore, lightweight improvements to the Neck structure can effectively enhance YOLOv12's real-time performance.

To this end, this study proposes replacing the standard convolutions in Neck with lightweight GhostConv[17] (as shown in Figure 7). This method generates more “ghost” feature maps through low-cost operations, enriching feature diversity without increasing computational overhead. As illustrated in Figure 6, GhostConv consists of two steps:

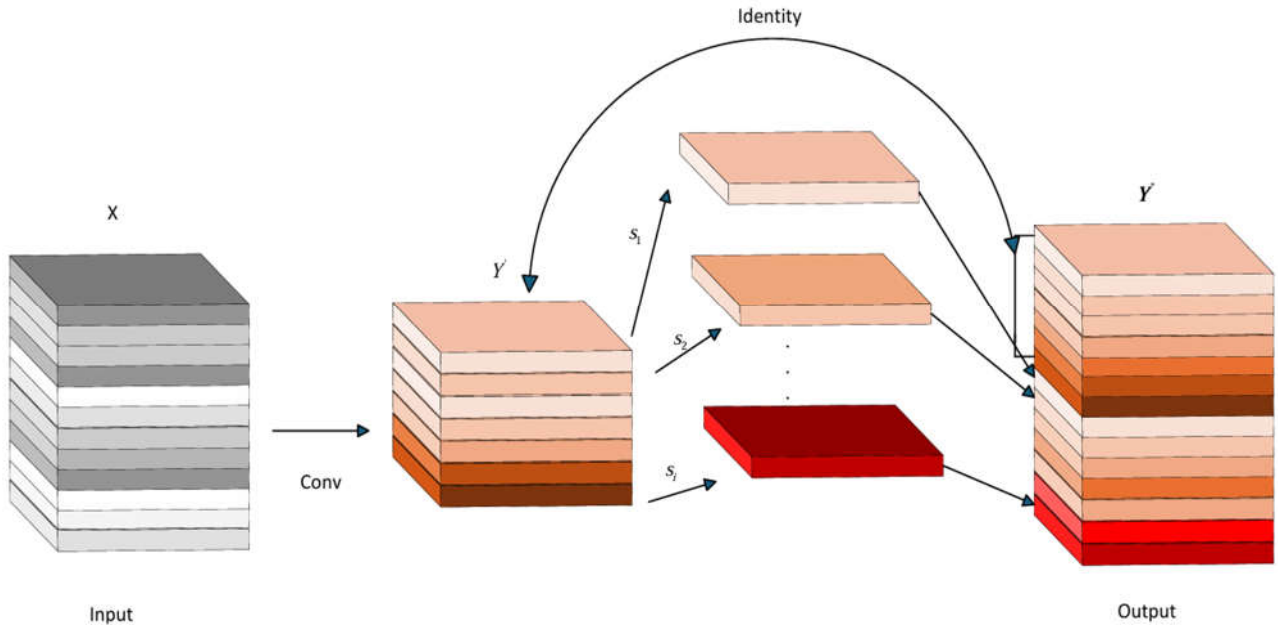


Figure 6. Schematic diagram of GhostConv.

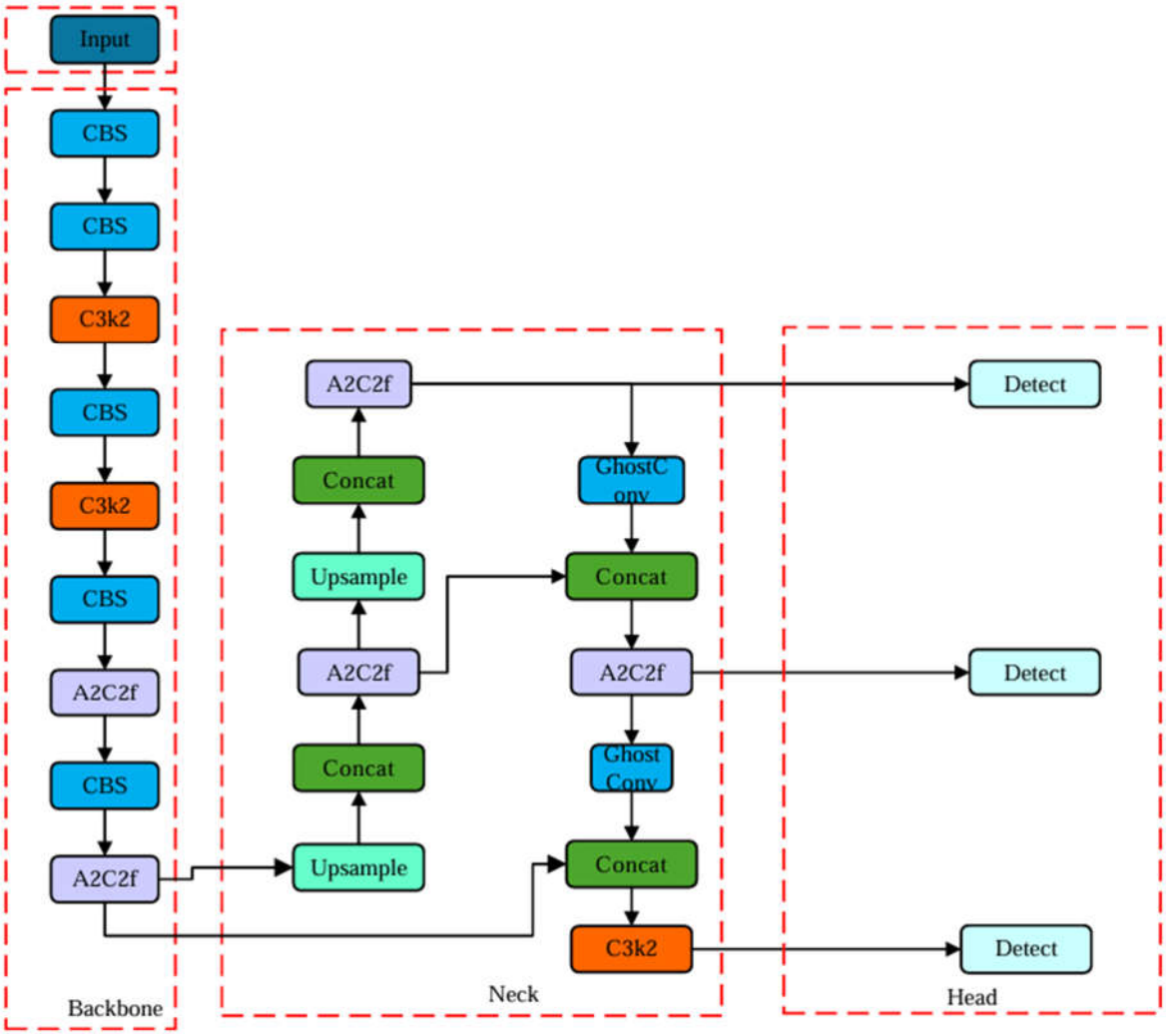


Figure 7. Structural diagram of GhostConv partially replacing the standard convolutions.

(a) Convolutional Layer: First, generate several basic feature maps through conventional convolution. Let the input feature map be $X \in R^{c\times h\times w}$, and the convolution kernel for standard convolution be $W \in R^{c'\times c\times k\times k}$, Then the basic feature map $Y' \in R^{c'\times h'\times w'}$ is given by Equation (8). Here,*denotes the convolution operation.

$$Y' = W * X \tag{8}$$

(b) Phantom generation:Generate phantom feature maps $Y'' \in R^{sc'\times h'\times w'}$ from Y' via low-cost operations.Specifically, uniformly partition Y'into s subsets along the channel dimension, apply linear transformations to each subset, and concatenate them by channel to obtain Y'' as shown in Equation (9).$Y'_i \in R^{\frac{c'}{s}\times h'\times w'}$ denotes feature subset i, $B_i \in R^{\frac{c'}{s}\times\frac{c'}{s}\times 1\times 1}$ represents the parameters of the i−th linear transformation, and s is the phantom scale. The final output feature map $Y \in R^{(s+1)c'\times h'\times w'}$ is the concatenation of $Y'$and Y'' along the channel dimension, as shown in Equation (10):

$$Y'' = Concat(B_i * Y'_1, \ldots, B_s * Y'_s) \tag{9}$$

$$Y = Concat(Y', Y'') \quad (10)$$

3.6 Head improvement: the wise IoU loss function

The overall loss function of YOLOv12 continues the design philosophy of multi-task Q-cooperative optimization, comprising localization loss (CloU-Pro), confidence loss (DynamicFocalLoss), and classification loss (Meta-Contrastive Loss):

(a) Localization Loss (CloU-Pro): As shown in Equation (11), YOLOv12 introduces dynamic aspect ratio penalties and probability distribution modeling on top of the traditional CloU (Complete Intersection over Union) [18] to address localization bias in occlusion scenarios.

$$L_{Ioc} = 1 - CIoU + \alpha \cdot \frac{p^2(b_{pred}, b_{gt})}{c^2} + \beta \cdot \frac{v}{1 - IoU + v} \quad (11)$$

In equation (11), p denotes the Euclidean distance to the center point, c represents the diagonal length of the minimum bounding box, and v is the aspect ratio difference term, defined as $v = \frac{4}{\pi^2}(arctan\frac{w_{gt}}{h_{gt}} - arctan\frac{w_{pred}}{h_{ored}})^2$ (dynamically adjusting the weight β of the aspect ratio difference term).

Probability Distribution Modeling: The coordinates (x, y, w, h) of the prediction box are modeled as a Gaussian distribution. The expected overlap area is calculated via Monte Carlo sampling to enhance the stability of small object detection.

(b) Confidence Loss (Dynamic Focal Loss): In YOLOv12, confidence loss addresses the positive-negative sample imbalance by incorporating a dynamic label assignment strategy (as shown in Equation 12), thereby reducing interference from low-quality negative samples.

$$L_{conf} = -\sum[y_{obj} \cdot (1-p)^{\gamma} \log(p) + (1 - y_{obj}) \cdot p^{\gamma} \log(1-p)] \quad (12)$$

$(1-p)^{\gamma}$ is dynamically adjusted based on the IoU between the predicted and ground−truth labels. For samples with high IoU,$(1-p)^{\gamma}$ is reduced to suppress overfitting on simple samples.

(c) Classification Loss (Meta-Contrastive Loss): Categorical loss dynamically adjusts class weights for long-tail data distributions through meta-learning (as shown in Equation 13), thereby enhancing recognition capabilities for rare classes.

$$L_{cls} = -\sum_c [\ \gamma^c \cdot y_c \log(p_c) + (1 - \gamma^c) \cdot (1 - y_c) \log(1 - p_c)] + \lambda \cdot L_{cont} \quad (13)$$

γ^c is dynamically generated by the meta-network based on category feature embeddings, reinforcing the learning weights for rare classes. The final total loss is the weighted sum of the three loss terms in Equation (14).

$$L = \lambda_{Ioc} L_{Ioc} + \lambda_{conf} L_{conf} + \lambda_{cls} L_{cls} \quad (14)$$

3.7 The loss function adopted by YOLOv12 exhibits three major critical flaws in small object detection

(a) Scale-insensitive distribution learning bias: The original Distribution Focal Loss (DFL) loss function employs a uniform discretization mechanism for objects of varying sizes, causing small objects to suffer severe quantization accuracy loss within the limited discrete

intervals (reg_max=16). Large targets, occupying more pixels, receive richer gradient signals that dominate the network's optimization direction. This significantly degrades the bounding box regression accuracy for small targets, leading to severe missed detections and localization errors in dense small-object scenarios.

(b) Lack of Shape-Adaptive Geometric Constraints: While the CIoU loss function accounts for center point distance and aspect ratio, it lacks specialized optimization for the unique shape characteristics of small objects. In detecting slender small objects (e.g., traffic sign poles, plant stems), traditional rectangular priors severely mismatch actual shapes, causing systematic bias in IoU calculations and significantly reducing shape-sensitive object recall rates.

(c) Gradient Instability with Low-Quality Samples: The combination of a strict Task-Aligned Assigner and CIoU exhibits significant vulnerability to low-quality small-object samples (e.g., blurred, occluded, incomplete). The scarcity of high-quality positive samples results in weak gradient signals, where minor positional deviations cause drastic fluctuations in IoU values. This leads to gradient oscillations and unstable convergence during training.

Therefore, addressing these limitations, this study deeply optimizes the detection loss function of YOLOv12. It adopts the scale-adaptive Wise IoU_v2 loss function to comprehensively enhance small object detection performance:

3.8 Adapted wise IoU loss

The training datasets for object detection tasks using adaptive Wise IoU loss functions are predominantly of high quality, designed to enhance the fitting capability of the bounding box loss. However, datasets like the purchased tomato dataset used in this paper may contain numerous occlusion-affected features and exhibit subpar quality. Persisting with bounding box regression would jeopardize model performance. Therefore, this paper adopts the Wise IoU Loss (WIoU)[19]. This function employs a gradient gain allocation strategy to mitigate the detrimental influence of low-quality samples while reducing the competitiveness of high-quality anchor boxes. This enables WIoU to focus on medium-quality anchor boxes, thereby enhancing the detector's overall performance. The loss function currently exists in three versions: v1, v2, and v3. Since low-quality data inevitably exists in training datasets, geometric factors like distance and aspect ratio exacerbate penalties on low-quality samples, thereby weakening model generalization. Thus, version v1 employs a dual-attention mechanism: when anchor frames align well with target frames, geometric penalties are reduced, and minimizing training interference enhances model generalization. WIoU_v1 is defined as follows:

$$L_{WIoUv1} = R_{WIoU} L_{IoU} \tag{14}$$

$$R_{WIoU} = \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}\right) \tag{15}$$

Here, W_g and H_g denote the dimensions of the minimum bounding box.(x, y) and (x_{gt}, y_{gt})represent the coordinates of the anchor frame and target frame center of mass, respectively. R_{WIoU} significantly amplifies the L_{IoU} value of ordinary quality anchor frames, whilevalue of ordinary quality anchor frames, while L_{IoU} markedly reduces the R_{WIoU} value of high-quality anchor frames.

v2 builds upon v1 by incorporating a monotonic focus mechanism, $L_{IoU}^{\gamma^*}$ for cross-entropy calculation. This mechanism enables the model to concentrate on complex samples, effectively reducing the contribution of low-quality samples to the loss value, thereby enhancing classification performance, as demonstrated by WIoU_v2 in Equation (17).

$$L_{WIoUv2} = L_{IoU}^{\gamma^*} L_{WIoUv1}, \gamma > 0 \tag{16}$$

Due to the increase in the focusing factor, the gradient of WIoU_v2 backpropagation similarly undergoes the following changes：

$$\frac{\partial L_{WIoUv2}}{\partial L_{IoU}} = L_{IoU}^{\gamma^*} \frac{\partial L_{WIoUv1}}{\partial L_{IoU}}, \gamma > 0 \tag{17}$$

$$L_{WIoUv2} = (\frac{L_{IoU}^*}{\overline{L_{IoU}}})^{\gamma} L_{WIoUv1}, \gamma > 0 \tag{18}$$

L_{IoU} is the normalization factor,$\overline{L_{IoU}}$ is the exponential moving average of momentum, and $(\frac{L_{IoU}^*}{\overline{L_{IoU}}})^{\gamma}$is the gradient gain. This parameter dynamically updates to maintain a higher gradient gain, addressing slow convergence during late training stages. By establishing a dynamic non-monotonic focus mechanism, version v3 assigns smaller gradient gains to anchor frames with larger outlier values.This effectively prevents low-quality samples from generating larger harmful gradients. The outlier value of an anchor frame is denoted as $\beta = \frac{L_{IoU}^*}{\overline{L_{IoU}}}$, where a smaller outlier value indicates higher anchor frame quality. The non-monotonic focus coefficient is derived based on the anomaly level, combined with the v1 version to obtain WIoU_v3 in Equation (20).

$$L_{WIoUv3} = rL_{WIoUv1}, r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{20}$$

Among these,αandδare hyperparameters that control the outlier threshold βand gradient gain r. WIoU has three versions, each with distinct advantages. Since this training dataset was purchased and contains a significant number of low-quality samples, this paper selects version v2 as the loss function to optimize YOLOv12 and enhance the model's classification performance.

3.9 Combination and Synergistic Optimization of Improvement Strategies

In the preceding three subsections, this study proposes improvements targeting the backbone network, neck structure, and head model of YOLOv8, respectively, along with three strategies: a spatial attention mechanism, GhostConv feature fusion, and the WIoU_v2 function. Figure 8 illustrates the deployment locations of these three enhancement modules within YOLOv12-WGS and the overall network architecture. The following sections will explore how to combine these strategies into a synergistically optimized improvement scheme.

The YOLOv12-WGS architecture incorporates three strategic enhancements: (1) Integrating the SE attention module at the spatial pyramid feature output of the backbone network; (2) Partial replacement of standard convolutions with GhostConv modules in the neck structure; (3) Implementation of the WIoU_v2 loss function at the final head output.

These modifications establish a hierarchical cascade of improvements spanning shallow, intermediate, and deep layers, collectively addressing spatial attention, channel-level feature refinement, and scale-adaptive optimization throughout the detection pipeline.

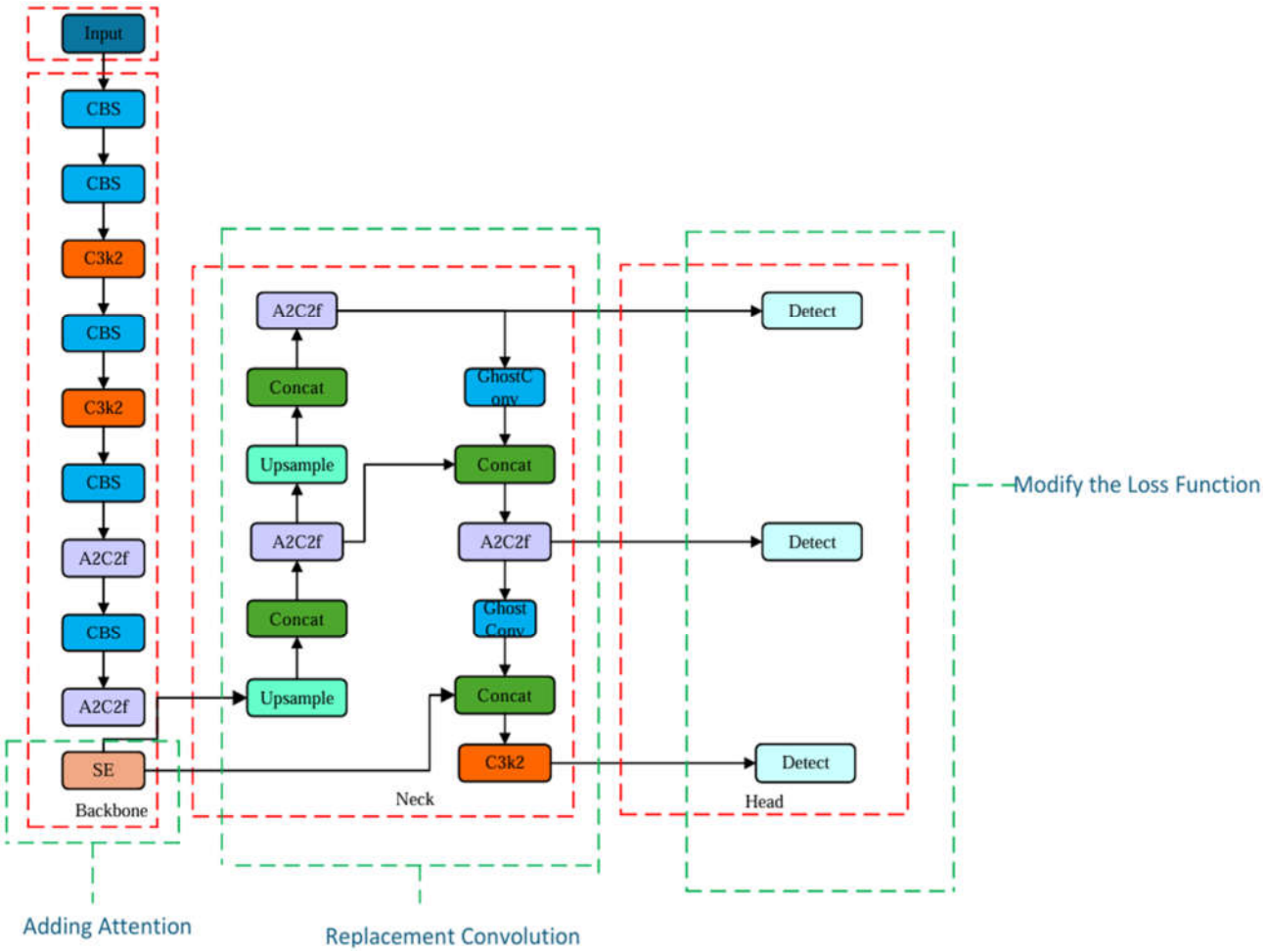


Figure 8. Schematic diagram of the overall network structure of the improved YOLOv12-WGS.

(a) The SE module processes different semantic features through adaptive weighting, highlighting target-relevant regions during the backbone network stage to establish a robust feature foundation for subsequent detection. Simultaneously, the introduction of SE aids in capturing detailed target information on high-resolution feature maps, further enhancing detection accuracy for small objects.

(b) The GhostConv module expands feature channel counts via low-cost operations, minimizing computational overhead while maximizing semantic information retention. When applied to feature fusion for neck structures, it enhances the discrimination capability between high- and low-level features, improves network robustness, and meets real-time requirements.

(c) The WIoU_v2 function adaptively adjusts regression loss based on target scale and shape during the Head stage, increasing the weight for small targets. It also correlates confidence prediction with detection frame quality, forming a more balanced loss function that simultaneously improves localization accuracy and classification performance.

These three modules—from the backbone network to the neck network to the head network—form a progressively enhanced optimization pathway: first amplifying local features, then optimizing global fusion, and finally guiding the loss function. This creates a closed-loop system of interconnected, synergistically enhanced components. The computational overhead introduced by these three modules remains relatively limited, ensuring the practicality of the optimization scheme. This combined strategy is named YOLOv12-WGS (WIoU_v2+GhostConv+SE Attention) to highlight its technical characteristics.

4. Results

4.1 Analysis of the Effectiveness of Attention Mechanism Improvements

To evaluate the impact of different attention mechanisms on YOLOv12 detection performance, this study selected three mainstream attention modules—CBAM, ECA, and S—and conducted validation experiments on the constructed tomato growth point bud multi-environment dataset. The training process lasted 300 epochs, with data collected from the test set. After integrating the three attention mechanisms into the YOLOv12 backbone network, their mAP, Precision, and Recall are summarized in Table 2. Adding CBAM and ECA attention mechanisms resulted in When model training is complete, mAP@0.5 and mAP@0.5:0.95, and Precision all show a downward trend, while Recall remained largely unchanged compared to the YOLOv12n baseline model and SE. Therefore, CBAM and ECA are not suitable for this dataset. Adding SE improved training accuracy, while Precision and Recall both increased.

Table 2. Comparison of the detection performance of different attention mechanisms.

Method	mAP@0.5	mAP@0.5:0.95	Precision	Recall
SE	0.730	0.639	0.766	0.668
CBAM	0.729	0.635	0.743	0.672
ECA	0.724	0.631	0.760	0.672
Origin	0.709	0.613	0.760	0.664

This study compared the effectiveness of three attention mechanisms (CBAM, ECA, SE) in enhancing the detection performance of YOLOv1 for tomato leaf diseases. Results indicate that the SE mechanism not only improves detection accuracy but also maintains model compactness, with its advantages primarily stemming from its efficient channel attention modeling capability[20]. The SE module learns inter-channel dependencies through global average pooling and a two-layer fully connected network, enabling adaptive recalibration of feature channels[21]. This mechanism is particularly well-suited for complex environmental changes in agricultural settings, dynamically adjusting channel weights based on feature responses under varying conditions to enhance key information while suppressing noise. The SE module features a streamlined design that not only reduces the risk of overfitting[22] but also effectively integrates multi-scale features within the feature pyramid structure[23]. This is particularly crucial for detecting growth points and flower

buds of tomatoes of varying sizes. In contrast, CBAM and ECA performed poorly in this study, likely due to their overly complex spatial attention calculations, which struggle to meet the demands of tomato organ recognition. Such tasks may rely more on channel-level features like color and texture rather than precise spatial localization information. Additionally, both mechanisms are susceptible to interference from complex backgrounds, leading to attention dispersion. In summary, the SE attention mechanism was ultimately selected for integration. This mechanism not only enhances model training accuracy but also meets lightweight deployment requirements. Figure 9 shows the training results of the YOLOv12 model after incorporating SE.

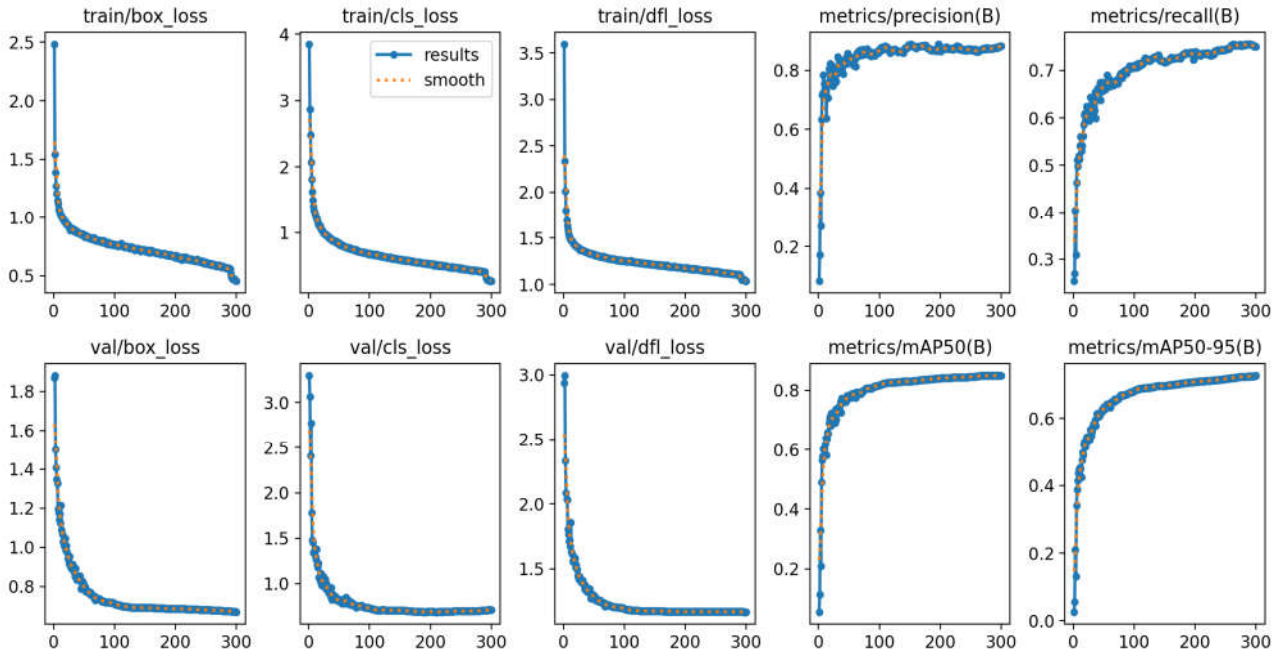


Figure 9. Training results of the YOLOv12 model with SE added.

4.2 Effect Evaluation of Lightweight Convolution Modules

Traditional convolutions often introduce redundant parameters during feature extraction, leading to increased model storage overhead and inference latency. To enhance the efficiency of the YOLOv12 neck feature fusion module, this study conducted training cycles of 300 epochs using test set data. Additionally, three representative lightweight convolution structures—Focus Convolution, Reparameterized Convolution (RepConv), and Ghost Convolution (GhostConv)—were selected for comparative analysis. Focus achieves efficient downsampling through hierarchical spatial information processing, while RepConv enhances model expressiveness via structural reparameterization, making it suitable for scenarios requiring a balance between accuracy and speed. Table 3 summarizes the improvements of the four convolutional architectures in terms of mAP@0.5, mAP@0.5:0.95, Precision, Recall, Params, and GFLOPs.

Table 3. Comparison of the detection performance of different convolutional

structures.

Method	mAP@0.5	mAP@0.5:0.95	Precision	Recall	Params/M	GFLOPs/G
GhostConv	0.706	0.614	0.761	0.674	2.5	6.0
Focus	0.711	0.618	0.762	0.672	3.1	6.9
RepConv	0.704	0.607	0.794	0.653	2.6	6.2
Origin	0.709	0.613	0.760	0.664	2.6	6.2

The purpose of improving the convolutional module is to reduce the number of parameters and the number of floating-point operations during model training. However, a reduction in the number of parameters often leads to a decrease in accuracy[24]. Therefore, this experiment uses the number of parameters and the number of floating-point operations as primary evaluation metrics, with mAP@0.5 and mAP@0.5:0.95 as a secondary metric. After modifying the model to the Focus convolution module, the number of floating-point operations increased by 0.7G, and the number of parameters increased by 0.5M. Thus, this approach was first excluded. After modifying the RepConv convolution module, neither the model's training parameters nor its floating-point operations decreased, failing to meet lightweight design standards. However, model training showed improvements: Precision increased by 3.4, mAP@0.5 precision decreased by 0.5, mAP@0.5:0.95 precision decreased by 0.6, while Recall decreased by 1.1. After modifying the GhostConv convolution module, both the number of training parameters and the number of floating-point operations were significantly reduced: And it's only mAP@0.5 precision decreased by 0.3, while other metrics improved. With substantial reductions in both training parameters and floating-point operations without significant accuracy loss, the GhostConv module outperformed the RepConv module. Therefore, this paper selects the GhostConv convolution module to improve the Neck convolution module. Parameter count decreased by 0.1M, floating-point operations reduced by 0.2G, with no significant accuracy loss. GhostConv achieves high parameter compression while maximally preserving feature discriminative power, striking an excellent balance between accuracy and parameters that meets lightweight design standards.

This study compared the performance of four lightweight convolutional structures in the YOLOv12 model, ultimately selecting GhostConv as the optimization solution. This approach significantly reduces computational complexity while maintaining high detection accuracy, validating GhostConv's effectiveness in object detection tasks. (The number of parameters is reduced by 3.8%, the computational load is decreased by 3.2%, and mAP@0.5 is only reduced by 0.3%.) This stems from its unique “feature reuse” mechanism. By generating a small number of ‘seed’ features and applying linear transformations to produce “ghost” features, GhostConv effectively minimizes redundant computations[25]. Unlike traditional convolution methods that extract a large number of redundant features[26], GhostConv achieves a “fewer but better” feature representation through its ingenious feature generation strategy, demonstrating that feature quality outweighs quantity in complex object detection tasks—a finding that will inform future lightweight model design. In contrast, other lightweight approaches exhibit limitations. While RepConv incurs only

minor accuracy loss, it fails to significantly reduce computational complexity, which contradicts the lightweight objective of this study[27]. Although the Focus module reduces computational load, the increase in parameters runs counter to the research objectives[28]. Figure 10 shows the training results of the YOLOv12 model after replacing standard convolutions with GhostConv.

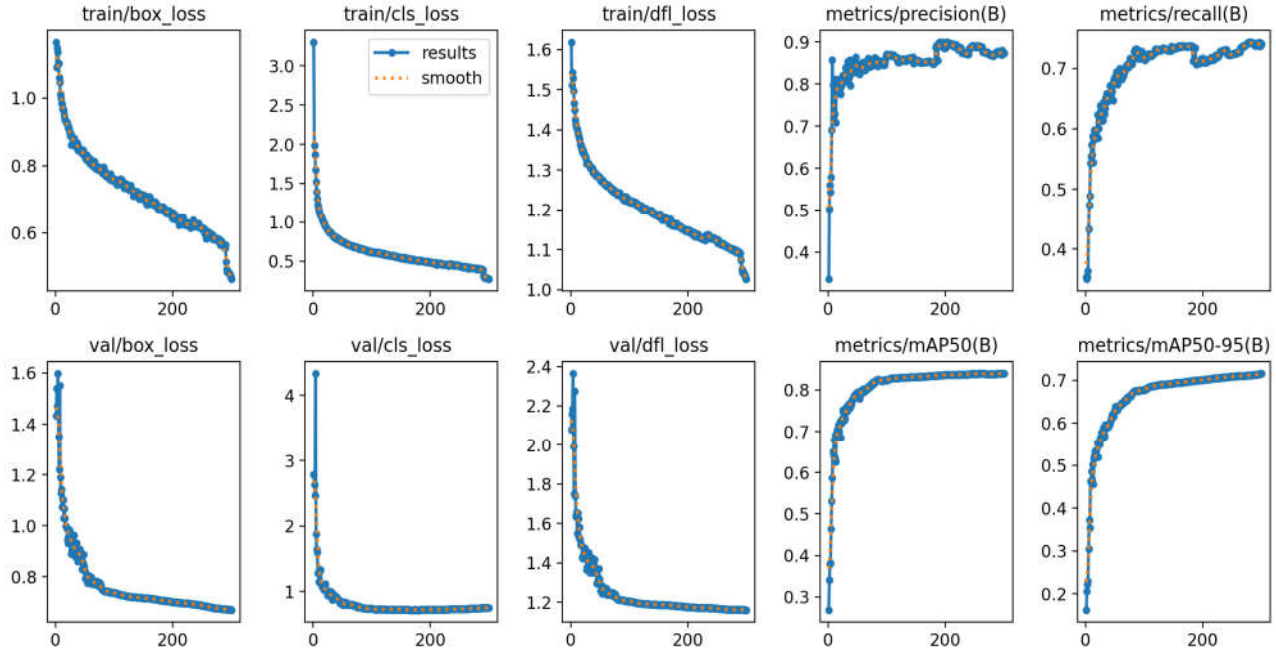


Figure 10.Training results of the YOLOv12 model after GhostConv embedding.

4.3 Impact Analysis of Loss Function Improvement

To evaluate the impact of different loss functions on the output results of the YOLOv12 detector, this study conducted training cycles of 300 epochs using data from the test set. Additionally, three widely used loss functions in the object detection field were selected: Effective Intersection over Union Loss (EIoU Loss), Generalized Intersection over Union Loss (GIoU Loss), and Generalized Intersection over Union Loss v2 (WIoU_v2). We summarized the accuracy, recall, and mAP metrics on the tomato leaf disease dataset (see Table 4), demonstrating the experimental effectiveness of the three convolutional architectures in enhancing the YOLOv12 model.

Table 4. Comparison of detection performance with different loss functions.

Method	mAP@0.5	mAP@0.5:0.95	Precision	Recall
WIoU_v2	0.723	0.618	0.757	0.672
EIOU	0.707	0.614	0.757	0.664
GIOU	0.710	0.618	0.763	0.665
Origin	0.709	0.613	0.760	0.664

Experimental tests indicate that modifying different loss functions does not affect the number of training parameters or the number of floating-point operations in the model.

Therefore, precision, recall, and mAP are adopted as evaluation metrics. The default loss function for YOLOv12 is CIoU. Results show that changing the loss function to EIoU or GIoU causes minimal variation in model training accuracy. However, after optimizing WIoU, all four metrics exhibit an upward trend. Model training accuracy remained largely unchanged, but optimizing WIOU_v2 resulted in an upward trend across all four metrics. Given the presence of numerous complex samples in this dataset, precision was prioritized as the evaluation metric to enhance model recognition accuracy. In summary, the selected loss function improvement module is the WIOU_v2 version. Based on the YOLOv12n baseline model, the recall rate increased by 0.8%, mAP@0.5 improved by 1.4%, and mAP@0.5:0.95 rose by 0.5%, while only a slight degradation in precision was observed. This effectively enhances object detection accuracy on the experimental dataset. In the tomato leaf disease detection task, the WIOU_v2 loss function significantly outperforms traditional loss functions, with advantages primarily manifested in the following aspects: (a) Multi-dimensional weighting mechanism: The introduction of multiple weighting factors effectively balances the contribution of different samples in complex agricultural scenarios[29], enhancing detection capabilities for concealed lesions or difficult-to-identify diseases. (b) Geometric information fusion: Comprehensive consideration of bounding box overlap, centroid distance, and aspect ratio[30]. Significantly enhances localization accuracy in key areas of tomato leaves while adapting to morphological size variations. (c) Adaptive Learning: The weight adaptation mechanism enhances the model's ability to handle changes in lighting conditions and occlusion issues[31], thereby improving detection robustness. (d) Scale Invariance: mAP@0.5:0.950.95 increase of 0.8%, indicating the model's robust performance across different IoU thresholds and suitability for detecting plant parts at various growth stages. Compared to EIoU and GIoU, WIoU_v2 demonstrates superior metric performance, primarily attributed to its enhanced capability in handling complex backgrounds and target diversity in agricultural scenarios. Figure 11 shows the training results of the YOLOv12 model after adopting WIoU_v2.

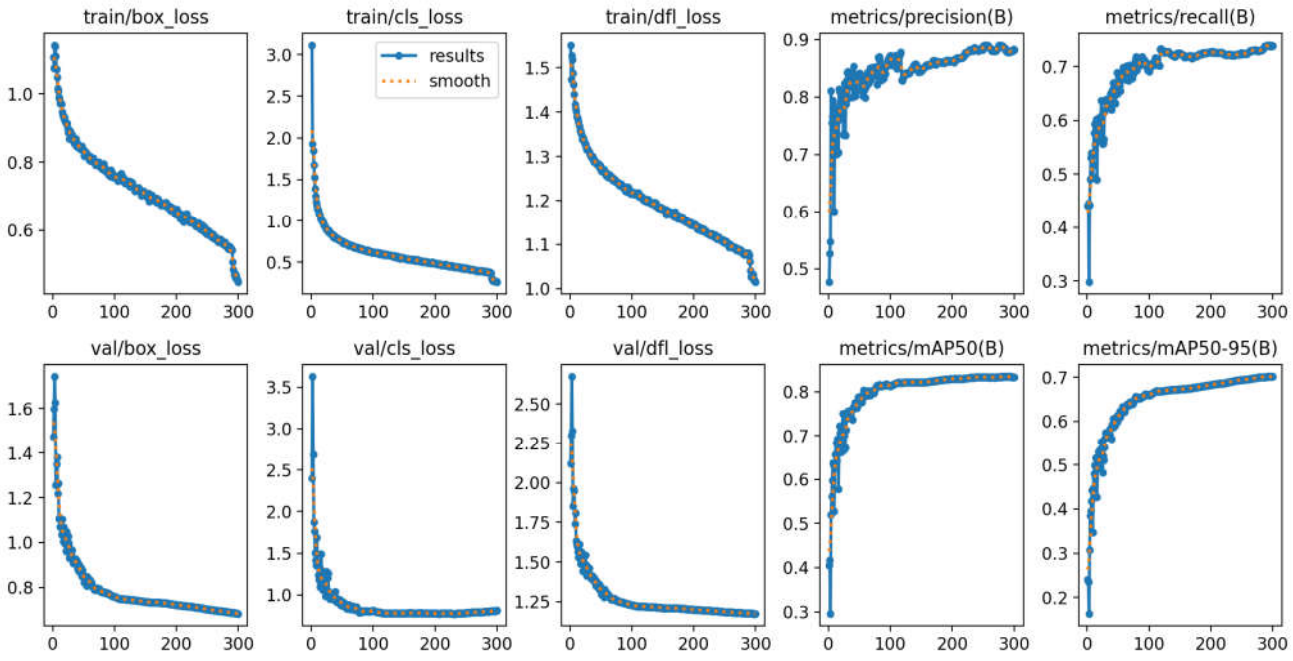


Figure 11.Training results of the YOLOv12 model after using WIoU_v2

4.4 Comparative analysis with other object detection models

In this study, the training process was conducted in cycles of 300 epochs using data from the test set to identify the optimal improvement scheme. To highlight the timeliness and efficiency of the improved algorithm and validate the effectiveness of the proposed enhancement method, its detection performance was compared with seven YOLO variants. As shown in Table 5, although YOLOv9t slightly outperforms YOLOv12-WGS in terms of precision and parameter count, it is inferior to YOLOv12-WGS in mAP@0.5, mAP@0.5:0.95, and recall rate. Furthermore, YOLOv9t requires significantly higher computational overhead, as reflected by its notably larger number of floating-point operations.YOLOv5, YOLOv6, YOLOv8, YOLOv10n, YOLO11, and YOLOv12 all underperform YOLOv12-WGS in mAP0.5, mAP0.5:0.95, accuracy, and recall. Therefore, the improved YOLOv12-WGS model proposed in this paper is the optimal algorithm for identifying tomato leaf diseases. Figure 11 shows the training results of the YOLOv12-WGS model.

Table 5. Comparative experiments in other YOLO variants.

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	Params/M	GFLOPs/G
YOLOv5	0.694	0.566	0.730	0.628	2.7	7.8
YOLOv6	0.682	0.584	0.673	0.618	4.5	13.0
YOLOv8	0.707	0.595	0.757	0.638	3.2	8.9
YOLOv9t	0.683	0.585	0.778	0.609	2.1	8.5
YOLOv10n	0.699	0.602	0.696	0.650	2.8	8.7
YOLO11	0.698	0.591	0.759	0.634	2.6	6.6
YOLOv12	0.709	0.613	0.760	0.664	2.6	6.2
YOLOv12-WGS	0.718	0.632	0.773	0.670	2.5	6.1

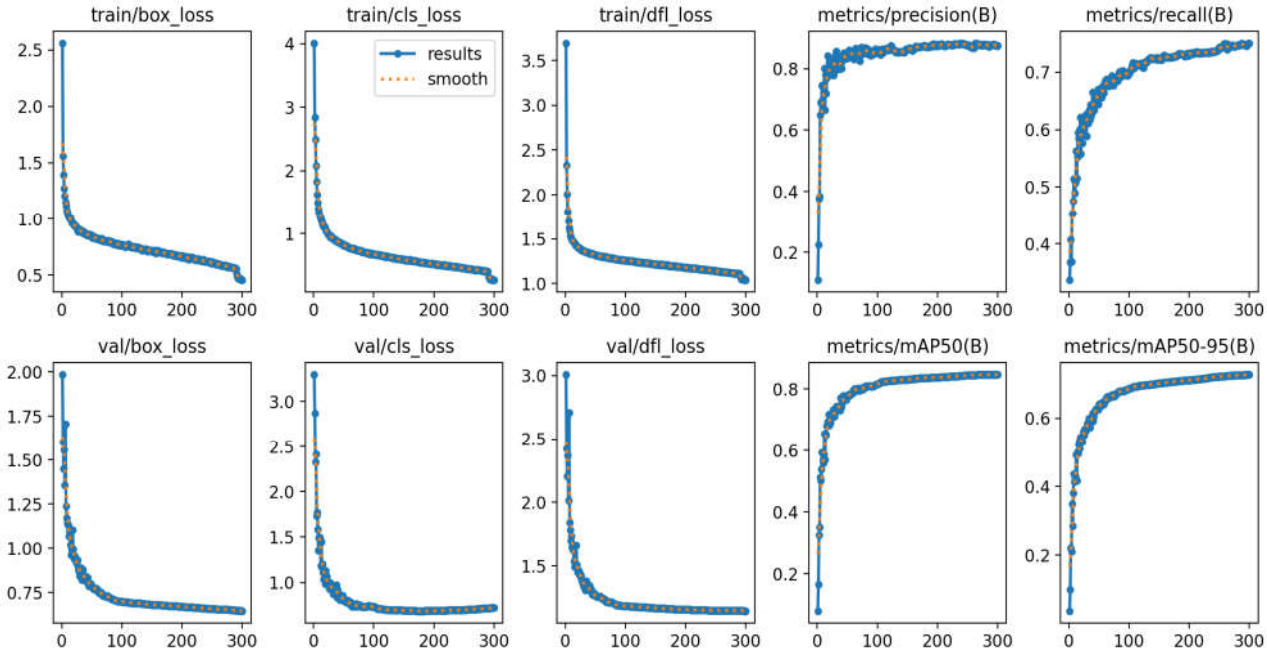


Figure 12. YOLOv12-WGS Model Training Results Diagram

5. DISCUSSIONS

Although this study has made significant progress in detecting tomato leaf diseases, future research must address the following limitations:(a) Research Limitations

First, the dataset's representativeness needs improvement. While a dataset covering 10 typical diseases was constructed, it is limited to a single tomato variety, which may affect the model's generalization ability. Second, the model's robustness in complex environments still needs improvement. Finally, comparative experiments were confined to the YOLO series, lacking experimental data from other models for contrast. (b) Future Research Directions. To address these limitations, future research may focus on three aspects: First, constructing diverse, large-scale tomato disease detection datasets. This involves expanding variety coverage and increasing environmental variability. Second, enhance the model's environmental adaptability and detection accuracy by integrating multimodal data fusion techniques, such as combining data from multiple sensors, including spectral imaging and thermal imaging. Third, obtain detection data from other model series like Faster R-CNN and GLIP for comparative analysis to identify shortcomings in the improved model and implement corrective measures.

6. CONCLUSIONS

This study addresses the critical challenge of detecting tomato leaf diseases by constructing a high-quality image dataset covering 10 typical leaf diseases. We systematically enhanced and optimized the YOLOv8 object detection model based on attention mechanisms, feature extraction, and loss functions. Key findings include: (a) The enhanced YOLOv8-WGS model significantly improves detection accuracy by integrating SE attention modules into the backbone network, employing GhostConv convolutions in

the neck region, and incorporating scale-adaptive weighting WIOU_v2 into the loss function—achieving the accuracy rate for detecting tomato leaf disease detection mAP@0.5 improved from 70.9% to 71.8%, and with mAP@0.5 rising from 0.95 to 0.95, and mAP@0.5: 0.95 increasing from 61.3% to 63.2%. (b) The enhanced model demonstrated superior scale adaptability for detecting tomato leaf diseases, with recall improving from 66.4% to 67.0%, showcasing exceptional target recognition and environmental adaptability. In summary, the multi-strategy optimization method presented in this study effectively addresses the challenges in tomato leaf disease detection. It not only significantly enhances detection accuracy and robustness but also achieves lightweight deployment, thereby providing a novel approach and a practical solution for tomato leaf disease detection in complex environments. This approach not only guides precise regulation in greenhouse tomato production to enhance yield and quality but also provides a reference for intelligent detection of other crop organs. It serves as a key driver for advancing greenhouse agriculture toward digitalization, automation, and intelligence, demonstrating broad application prospects.

7. ACKNOWLEDGES

This work was supported by the Graduate Student Innovation Project (SJCX25_XY007) and the Yancheng Key Research and Development Program (Social Development) Project (YCBE202506).

8. REFERENCES

[1] Chunman Yan, Huiling Li. CSPNet: A feature interaction network for tomato leaf d104isease detection in complex scenarios[J]. Computers and Electronics in Agriculture, 2025, 238 110823-110823.

[2] Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, Suneet Gupta. ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network[J]. Procedia Computer Science, 2020, 167 (C): 293-301.

[3] Fahim Mahafuz Ruhad, Md Fahim, Mir Sazzat Hossain, Md. Fahad Monir, Ashraful Islam, M. Ashraful Amin. Beyond classification: Benchmarking object detection models for efficient tomato leaf disease identification on a real-world dataset[J]. Smart Agricultural Technology, 2025, 12 101336-101336.

[4] Jiangjun Yao, Yiming Li, Zhengyan Xia, Pengcheng Nie, Xuehan Li, Zhe Li. WTAD-YOLO: A lightweight tomato leaf disease detection model based on YOLO11[J]. Smart Agricultural Technology, 2025, 12 101349-101349.

[5] Debtanu Ghosh, Subhayu Ghosh, Nanda Dulal Jana, Suparna Biswas, Rammohan Mallipeddi. Designing optimal Vision Transformer architecture using differential evolution for tomato leaf disease classification[J]. Computers and Electronics in Agriculture, 2025, 238 110824-110824.

[6] Vinay Gautam, Anand Muni Mishra, Pabhjot Kaur, Mukund Pratap Singh, Prabhishek Singh, Manoj Diwakar, Indrajeet Gupta. Composite deep learning model for characterization of tomato leaf disease[J]. Multimedia Tools and Applications, 2025, (prepublish): 1-30.

[7] P. Ananthi, K. N. Devi, G. D, P. Shanmugapriya and G. S, "Tomato Leaf Diseases Prediction Using Deep Learning Algorithms," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ADICS58448.2024.10533491.

[8] C. Song et al., "TDPPL-Net: A Lightweight Real-Time Tomato Detection and Picking Point Localization Model for Harvesting Robots,"in IEEE Access,vol 11, pp.37650-37664, 2023, doi:10.1109/ACCESS.2023.3260222.

[9] L. -D. Quach, K. N. Quoc, A. N. Quynh, H. T. Ngoc and N. Thai-Nghe, "Tomato Health Monitoring System: Tomato Classification, Detection, and Counting System Based on YOLOv8 Model With Explainable MobileNet Models Using Grad-CAM++," in IEEE Access, vol. 12, pp. 9719-9737, 2024, doi: 10.1109/ACCESS.2024.3351805.

[10] R. Liu, "YOLOv10 Tomato Ripening Detection Enhanced by Convolutional Neural Network Attention Mechanism," 2024 4th International Conference on Computer Science and Blockchain (CCSB),Shenzhen,China,2024,pp.90-93,doi:10.1109/CCSB63463.2024.10735542.

[11] J. Xue, Y. Zheng, C. Dong-Ye, et al. Improved YOLOv5 network method for remote sensing image-based ground objects recognition, Soft Comput, 26: 10879–10889, 2022.

[12] Kumar S K P ,Kumar V Y ,S. V B , et al. Fish Detection in Underwater Environments Using Deep Learning [J]. National Academy Science Letters, 2023, 46 (5): 407-412. DOI:10.1007/S40009-023-01265-4.

[13] Mathew ,P. M ,Mahesh , et al. Leaf-based disease detection in bell pepper plant using YOLO v5 [J]. Signal, Image and Video Processing, 2021, 16 (3): 1-7. DOI:10.1007/S11760-021-02024-Y.

[14] J. Redmon, S. Divvala, R. Girshick, et al. You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:779-788.

[15] Wen G ,Shuang Q ,Chenyi Z , et al. Defect detection for industrial neutron radiographic images based on modified YOLO network [J]. Nuclear Inst. and Methods in Physics Research, A, 2023, 1056 DOI:10.1016/J.NIMA.2023.168694.

[16] A. G. Roy, N. Navab and C. Wachinger, "Recalibrating Fully Convolutional Networks With Spatial and Channel “Squeeze and Excitation” Blocks," in IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 540-549, Feb. 2019, doi: 10.1109/TMI.2018.2867261.

[17] Wang, X. & Liu, J. Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment. Sci. Rep. 14, 4261 (2024).

[18] Qi, Z. et al. A novel method for tomato stem diameter measurement based on improved YOLOv8-seg and RGB-D data. Comput. Electron. Agric. 226, 109387 (2024).

[19] Tong, Z., Chen, Y., Xu, Z., Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv.org (2023).

[20] Qi, J. et al. An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. Comput. Electron. Agric. 194, 106780 (2022).

[21] Jin, X. et al. Delving deep into spatial pooling for squeeze-and-excitation networks. Pattern Recognit. 121, 108159 (2022).

[22] Islam, M. P. et al. Performance prediction of tomato leaf disease by a series of parallel convolutional neural networks. Smart Agric. Technol. 2, 100054 (2022).

[23] Wang, Y., Zhang, P. & Tian, S. Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion. Front. Plant Sci. 15, 1382802 (2024).

[24] Zhang, Z. et al. A method for counting fish based on improved YOLOv8. Aquac. Eng. 107, 102450 (2024).

[25] Gong, T. & Ma, Y. PSO-based lightweight neural architecture search for object detection. Swarm Evol. Comput. 90, 101684 (2024).

[26] Liu, J. & Wang, X. Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. Front. Plant Sci. 11, 898 (2020).

[27] Li, R., Shao, Z. & Zhang, X. Rep2former: A classification model enhanced via reparameterization and higher-order spatial interactions. J. Electron. Imaging 32, 053002 (2023).

[28] Usman, M., Zaka-Ud-Din, M. & Ling, Q. Enhanced encoder–decoder architecture for visual perception multitasking of autonomous driving. Expert Syst. Appl. 246, 123249 (2024).

[29] Chen, Z., Liu, H., Zhang, L. & Liao, X. Multi-dimensional attention with similarity constraint for weakly-supervised temporal action localization. IEEE Trans. Multimed. 25, 4349–4360 (2023).

[30] Xiong, C., Zayed, T. & Abdelkader, E. M. A novel YOLOv8-GAM-Wise-IoU model for automated detection of bridge surface cracks. Constr. Build. Mater. 414, 135025 (2024).

[31] Shen, L., Lang, B. & Song, Z. DS-YOLOv8-based object detection method for remote sensing images. IEEE Access 11, 125122 125137 (2023).