Research on citrus leaf disease detection based on improved SEDS-YOLOv8 model with efficient multi-scale attention and multi-level channel compression

Houbin Wang[1,2,], Yongwei Wang[3], Junyi Liu[4], Jianing Chang[1]*

[1] School of Resources and Environmental Engineering, Ludong University, Yantai, Shandong Province, China

[2] Ruiluweijun (Yantai) Information Technology Co., Ltd., Yantai, Shandong, China

[3] Wenchang county Yuanzhuang town people's government, Wenshang County, Jining, Shandong Province, China

[4] School of Mechanical Engineering, Liaoning Technical University, Fuxin, Liaoning Province, China

ABSTRACT

Citrus is one of the important economic crops, with a vast planting area and complex terrain and environmental conditions. The growth cycle of citrus is long, and it is prone to a wide variety of diseases and pests. If the types of diseases and pests cannot be accurately identified in a timely manner to take corresponding control measures, it will seriously affect the yield and quality of citrus. This study aims to improve the detection accuracy of leaf diseases and pests, reduce the computational scale of the model, and enhance its deployability. A lightweight disease and pest detection model based on the improved YOLOv8 is proposed, and a disease and pest dataset considering different environmental conditions is established. Firstly, the convolutional module (Conv) in the neck network of YOLOv8 is replaced by GSConv, and the C2f module is replaced by VoV-GSCSP, forming a Slim-neck architecture, which reduces the computational complexity of the model while maintaining high recognition accuracy. At the same time, the C2f module in the backbone network is replaced by the C2f_EMA module that integrates the EMA efficient multi-scale attention mechanism, enhancing the model's feature extraction ability for leaf diseases and pests in complex environments. Additionally, the original detection head is improved through multi-level channel compression to reduce features along the channel dimension. The SEDS-YOLOv8 model is designed through the above methods. Experimental results show that the model's parameters, computational cost, and memory usage are reduced by 63.5%, 72.83%, and 61.9% respectively. The model's precision, recall, and mean average precision are 97.5%, 96.2%, and 98.5% respectively. In terms of performance, the detection frame rate on mobile devices reaches 358.5 frames per second, and the average inference time for a single leaf disease and pest image is 4.4 ms. This proves that the algorithm can significantly reduce the computational load of the network while maintaining high detection performance, meeting the deployment requirements of mobile and embedded devices.

Keywords: Pest detection; Slim-neck architecture; EMA efficient multi-scale attention; Multilevel channel compression; Channel pruning

1 INTRODUCTION

Citrus fruits are among the most common fruits in people's daily diets, with a wide variety of species. Their rich nutritional content and sweet, refreshing taste make them highly popular. As a widely cultivated crop globally, citrus brings substantial income to producing regions and farmers. In many rural areas, citrus cultivation has become a key pillar industry, playing an indispensable role in rural revitalization [1]. With the expansion of planting areas, ensuring citrus yield has become a major concern for farmers. Due to the long growth cycle of citrus and its susceptibility to climatic conditions and inherent growth characteristics, the crop is highly prone to various pests and diseases. Common citrus pests include rust mites, leafminers, aphids, and scale insects [2]. These pests tend to occur sporadically, making it difficult to predict their outbreaks in advance, and they can cause severe damage to citrus trees. Additionally, they may trigger concurrent infections with diseases such as Huanglongbing (citrus greening disease) and gummosis. These pests and diseases can lead to citrus tree decline, reduced fruit yield, and lower fruit quality. In severe cases, they may even cause plant death, resulting in significant economic losses for farmers [3].

There are numerous pest and disease control techniques for citrus, among which pesticide spraying is one of the most effective and commonly used methods. However, due to the wide variety of citrus pests and diseases, different diseases require specific treatment approaches, and different pests correspond to different pesticides [4]. Incorrect pesticide application not only fails to effectively control diseases, but may also cause additional harm to citrus trees, further reducing yield. Moreover, excessive fertilization and improper pesticide use can lead to soil and water pollution, posing risks to human health [5]. Therefore, accurately detecting the type and location of pests and diseases not only enables timely prevention and control but also reduces production costs. By ensuring precise pesticide application, environmental pollution can be mitigated, and both citrus yield and quality can be improved [6].

Citrus pest and disease identification primarily relies on recognizing pathological characteristics on leaves, fruits, and other plant parts. Typically, by the time disease symptoms appear on the fruit, the infection has already reached an advanced stage [3]. To ensure timely control, leaves should be the primary focus of observation. However, citrus leaf symptoms often exhibit similarities, and infected areas tend to be densely clustered, making identification challenging. Currently, citrus disease and pest classification mainly depends on manual inspection, which relies on expert diagnosis or farmers' personal experience [7]. Since citrus orchards are often located in remote areas with complex environments, this manual identification method is labor-intensive, inefficient, and highly subjective, leading to inconsistent diagnostic accuracy. Misdiagnosis can delay optimal treatment timing, exacerbating losses. Developing an accurate and efficient method for detecting citrus pests and diseases would help growers quickly identify disease types and implement effective control measures in a timely manner, thereby minimizing losses caused by pests and diseases.

2 RELEATED WORK

With the rise of artificial intelligence and the rapid development of image recognition technology, many related algorithms have been applied to the detection of agricultural pests and diseases [8]. By collecting data and building models, these technologies help or replace manual decision-making. The inherent characteristics of citrus pests and diseases—such as similarities in interspecific features, complex natural environments, and occlusion by branches and leaves—make it easy for existing target detection models to suffer from issues like missed detections and false detections [9]. Therefore, they cannot meet the requirements for detecting various types and targets of pests and diseases in natural environments. Early research on image recognition of agricultural pests and diseases primarily combined traditional machine learning with image processing. Through image preprocessing, segmentation, and feature extraction techniques, features were manually selected and classified using machine learning algorithms (such as Support Vector Machines, K-Nearest Neighbors, linear regression analysis, Principal Component Analysis, etc.) to effectively identify pest images [10].

Traditional agricultural pest and disease detection based on machine learning has achieved substantial progress. Compared to manual identification of crop pests and diseases, machine learning methods offer higher efficiency and precision [8]. However, significant issues remain. The images of crop pests and diseases are often complex and irregular, making it difficult for traditional methods to extract optimal features for detection. Furthermore, these methods rely on manual feature extraction, leading to certain subjectivity and limitations, requiring considerable time, efficiency, and human resources for data preprocessing, which may lead to lower recognition accuracy [11]. The requirements for the detection environment are high, making it vulnerable to interference from other factors, and thus failing to adequately meet the practical requirements for pest and disease detection in the field [12].

Compared to machine learning, deep learning can automatically extract feature information from images, which prevents issues such as feature loss or errors that can arise from manual feature extraction [13]. Deep learning exhibits superior speed and accuracy over traditional algorithms, along with excellent generalization capabilities [14]. The advent of Convolutional Neural Networks (CNNs) has shown outstanding performance in the fields of image processing and classification and has begun to be extensively utilized in the detection of agricultural pests and diseases. With the continuous development of deep learning technology, various object detection algorithms based on deep learning have made the detection of agricultural pests and diseases more efficient [15]. These algorithms can provide accurate information regarding the location, category, and size of objects during the image detection process, better meeting the detection needs in practical applications [16]. Object detection algorithms are mainly divided into two categories: two-stage algorithms and one-stage algorithms [17].

Given the diversity of pests and diseases, significant variation in appearance and scale, and the similarity of relevant features to surrounding environments, detection remains challenging. This paper constructs a lightweight detection model dedicated to pest and disease detection based on the YOLOv8 object detection algorithm. Additionally, it studies the model's deployment performance on edge computing devices such as Jetson Xavier NX and Raspberry Pi 4B, aiming to provide technical support for efficient intelligent detection of pests and diseases and the development of intelligent pest control equipment.

3 MODEL ESTABLISHMENT AND SOLUTION

3.1 Network structure of YOLOv8 deep learning model

The YOLOv8 model network structure is shown in Figure 1, and it consists of four main parts: the input (Input), backbone (Backbone), neck (Neck), and detection head (Head) [18].

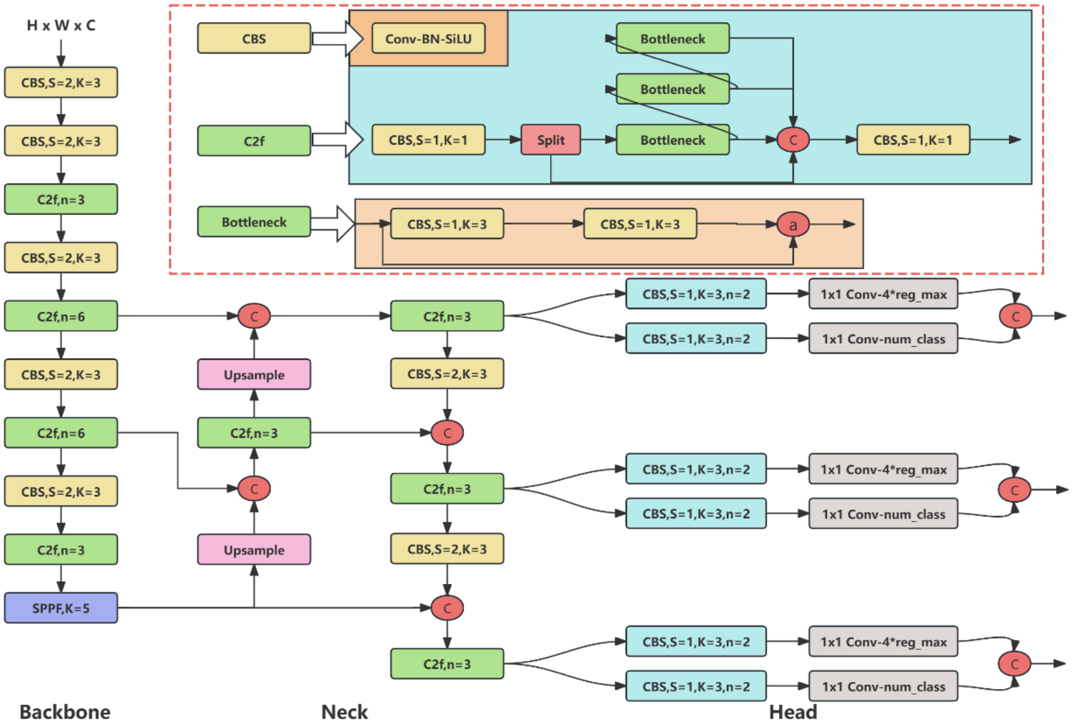


Fig. 1: Model network structure of YOLOv8

The Input preprocesses the input image to the network by performing adaptive resizing, which improves the image processing efficiency. The Backbone consists of Conv, C2f, and SPPF modules, which are used to extract feature information from the input image. The C2f module includes 2 Conv layers and *n* Bottleneck layers, which help to extract richer gradient information. The SPPF module includes three consecutive max-pooling operations to capture object information at different scales, thus enhancing the accuracy of object detection. The Neck part is composed of a combination of PAN (Path Aggregation Network) and FPN (Feature Pyramid Network) structures, which fuse feature maps from different layers and scales, ensuring the model's ability to extract features in multi-scale scenarios. The Head adopts a decoupled head structure and an anchor-free strategy, using 3 decoupled heads to perform image detection and classification tasks at different scales. Finally, the model outputs the target class and detection box location information.

3.2 Slim-neck module

In detecting leaf pests and diseases in natural environments, high accuracy and speed performance are essential for the model, as they directly affect its operational capability and deployability. Typically, the larger the number of model parameters, the larger the model size, which increases the difficulty and cost of deployment on actual mobile devices. Therefore, to maintain detection accuracy while effectively reducing the model's complexity, a lightweight network structure, Slim-neck, composed of GSConv and VoV-GSCSP modules, was built to improve the YOLOv8 model. In Slim-neck, the lightweight hybrid convolution GSConv (Group-Shuffle Convolution) was first used to replace all Conv modules in the original YOLOv8 neck network. The GSConv structure is shown in Figure 2 and consists of standard convolution (SC), depthwise separable convolution (DSC), and shuffle operations. The Slim-neck using the GSConv module maximizes the advantages of DSC and eliminates the negative effects that arise from channel information separation, such as reduced feature extraction capability. GSConv first performs SC on the feature map, followed by DSC, then concatenates the feature maps produced by both operations, and finally uses the shuffle operation to recombine the channels. This method fully utilizes the advantages of both convolutions, enabling the model to maintain detection performance while significantly reducing computational complexity.

Correspondingly, based on GSConv, the VoV-GSCSP (Variety of View Group Shuffle Cross Stage Partial Network) module was introduced to further reduce computational costs and balance detection accuracy. Its structure is shown in Figure 2. This module aggregates cross-stage partial networks in a one-time manner, optimizing the trade-off between efficiency and performance.

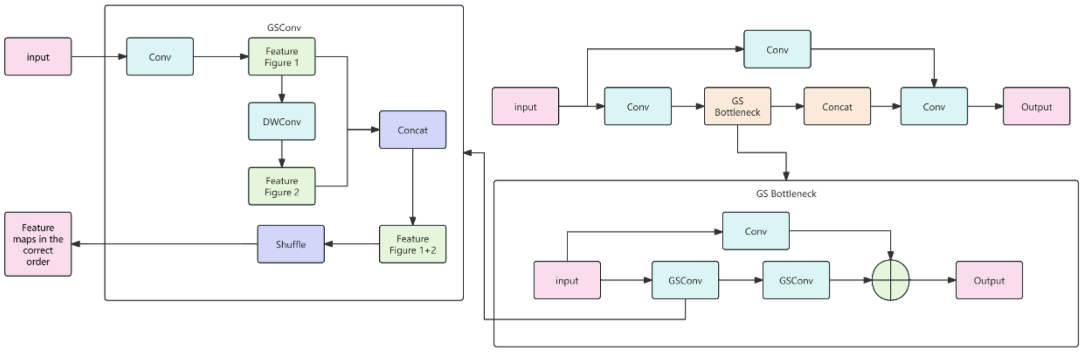


Fig. 2: Slim-neck Model structure

The feature extraction process of the VoV-GSCSP module is divided into two paths: Path One involves extracting features through conventional convolution on the input feature map; Path Two first processes the input feature map with conventional convolution, then performs feature extraction using the GSBottleneck structure designed based on GSConv, and finally concatenates the feature maps from both paths for output. The VoV-GSCSP module fully leverages the advantages of GSConv and GSBottleneck, enhancing the model's feature extraction capability, reducing the number of model parameters, and further improving the model's lightweight performance while maintaining detection accuracy. In this study, the VoV-

GSCSP module replaced all C2f modules in the Neck part of the original YOLOv8 model.

3.3 Efficient multi-scale attention mechanism of EMA

In the process of detecting leaf pests and diseases, issues such as partial occlusion between leaves and variations in environmental lighting can lead to missed detections. Additionally, there is a certain degree of feature similarity between different types of pests and diseases, as well as between pests and surrounding objects, which can cause false detections. These problems are related to the model's limited ability to extract effective leaf pest and disease feature information or its insufficient ability to filter the extracted features. To enhance the model's detection capability for leaf pest and disease images, this study introduces the EMA (Efficient Multi-Scale Attention) mechanism into the backbone network of YOLOv8. The EMA attention mechanism is an efficient multi-scale attention mechanism based on cross-spatial learning. It reshapes certain channels into the batch dimension and groups the channel dimensions, without requiring dimensionality reduction. This approach helps prevent the loss of channel feature information and reduces computational overhead, offering high accuracy and a low number of parameters.

The structure of the EMA attention mechanism module is illustrated in Figure 3. Its operational workflow is as follows: First, for any input $X \in \mathbb{R}^{(C \times H \times W)}$, the EMA divides it along the channel dimension into G sub-features, represented as $X = [X_0, X_1, \dots, X_{G-1}]$, where $X \in \mathbb{R}^{(C/G \times H \times W)}$, in order to capture different semantic information. Subsequently, the EMA employs three pathways to extract attention weight descriptors from the grouped feature maps.

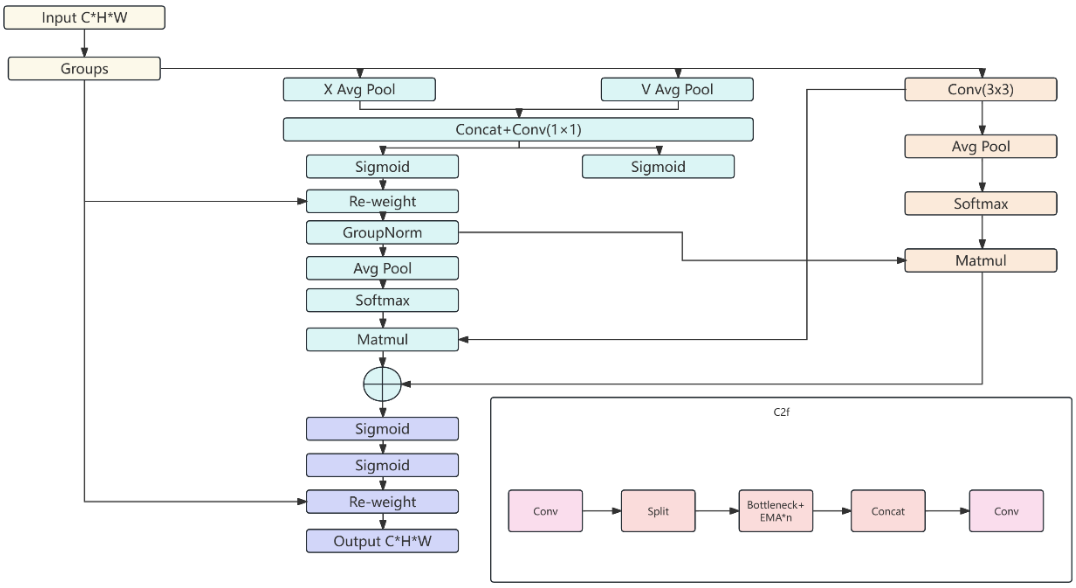


Fig. 3: Network structure of the EMA mechanism module

The first two pathways utilize a 1 × 1 branch that incorporates 1 × 1 convolution operations, while the third pathway employs a 3 × 3 branch that utilizes 3 × 3 convolution operations. In the 1 × 1 branch, two one-dimensional global average pooling operations are applied in different directions to encode the channels, facilitating cross-channel information interaction. Conversely, the 3 × 3 branch omits the one-dimensional global average pooling operation and Group Normalization to achieve multi-scale feature representation. Subsequently, a two-dimensional global average pooling operation is applied to encode global spatial information from the outputs of both the 1×1 and 3×3 branches. The formula for the twodimensional global average pooling operation is as follows:

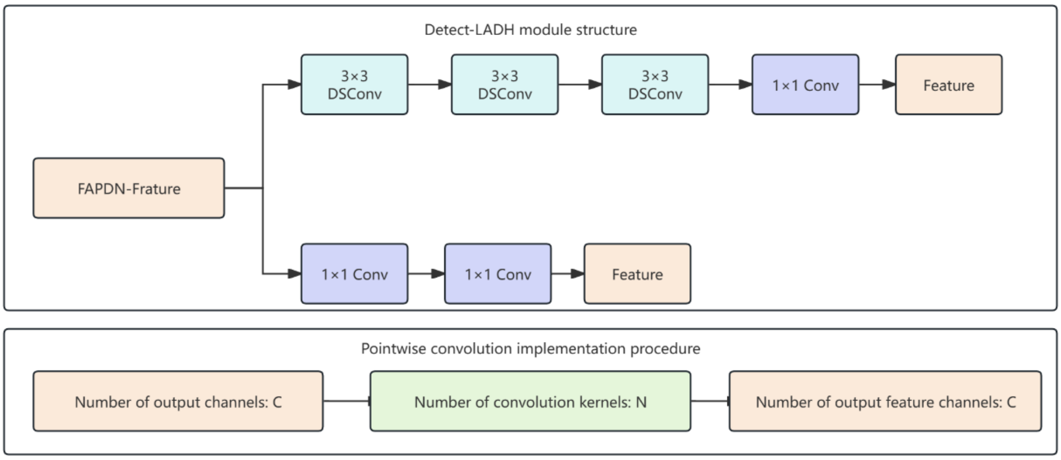
$$Z_C = \frac{1}{H \times W} \sum_j^H \sum_i^W X_C(i,j) \tag{1}$$

In the equation, Z_C represents the output value of the C-th channel after pooling, where H and W denote the spatial dimensions of the input features. The variable C indicates the number of channels, and $X_C(i,j)$ represents the input of the C-th channel at width i and height j.

The output feature maps within each group are obtained by aggregating two generated spatial attention weight values. Finally, a Sigmoid activation function is employed to capture pixel-level paired relationships, thereby acquiring global contextual information. This study integrates the EMA attention mechanism into the Bottleneck of the C2f module, forming the C2f_EMA module. This integration enhances the capability of the C2f module to capture multi-scale feature information by optimizing the Bottleneck structure. Overall, based on prior experience, the second and fourth C2f modules in the YOLOv8 backbone network are replaced with the C2f_EMA module. With the incorporation of the EMA attention mechanism, the model can utilize 1×1 and 3×3 convolutions to connect more contextual information in the intermediate feature maps, further refining and screening the characteristic information of disease-affected leaves. This approach to cross-spatial information aggregation in different spatial dimensions enables the model to effectively address the issues of missed and false detections of leaf pests and diseases.

3.4 Lightweight asymmetric detection head Detect-LADH

In this study, the number of pest and disease categories in the dataset directly affects the probability of false detections among different classes. The original YOLO algorithm employed a coupled head for detection tasks, primarily utilizing the same convolutional layers at the top of the network for classification and regression. However, these tasks have distinct focuses, which can lead to conflicts during the detection process. In subsequent improvements, while the introduction of a decoupled head method significantly enhanced the model's detection capabilities, it also greatly increased the network's parameter count, resulting in reduced inference speed. To address these issues, this paper draws upon the idea of improving the asymmetric decoupling head (ADH) proposed by Chollet and employs multi-level channel compression to enhance the detection head in the YOLOv8 network. The modified module is called Detect-LADH, with its structure illustrated in Figure 4. By isolating tasks within the network, two different channels are utilized to perform related tasks. To expand the receptive field and increase the task parameters for the regression branch, three depthwise separable convolutions (DSConvs) are employed to reduce features along the channel dimension, replacing the traditional 3×3 convolutions in each branch. The advantage of DSConv over standard convolutions lies in its significant reduction in parameter count by decomposing the convolution operation into depthwise convolutions and pointwise convolutions (PWConv). The operational flow for PWConv is shown in Figure 4, where the convolution operation weights and combines the channels C from the previous output along the depth direction to generate new feature channels C. The number of new channels generated equals the number of convolution kernels used. This module effectively resolves the conflicts introduced by the original coupled head between classification and regression tasks, thereby reducing the probability of false detections among different diseases. P3, P4, and P5 are multi system object detection heads, whereP3 corresponds to a detection map size of 80 × 80, P4 corresponds to a detectionfeature map size of 40 × 40, and P5 corresponds to a detection feature map sizeof 20 × 20; Conv_Seg is the regression channel, and Conv_Cls is theclassification channel.


Detect-LADH module structure

Pointwise convolution implementation procedure

Number of output channels: C

Number of convolution kernels: N

Number of output feature channels: C

Fig. 4: Detect-LADH module structure and Pointwise convolution implementation procedure

3.5 Model pruning

The improved methods used in the YOLOv8n model for leaf disease detection retain detection accuracy while compressing model size and computational load. However, the presence of numerous convolutional structures in the model still results in redundancy, which poses a significant resource burden for future deployment on embedded devices. To further lighten the model and accelerate inference speed, the Slim pruning method from structured pruning is employed for additional model compression.

In channel pruning, sparse training is first conducted on the Batch Normalization (BN) layers of the network model to filter out some unimportant channels. Currently, batch normalization is commonly used to expedite model convergence. During channel pruning, the BN layer normalizes the internal activations using small batch statistics. Let z_{in} and z_{out} represent the input and output of the BN layer, respectively, B denote the current mini-batch, and ϵ be a small positive constant to avoid division by zero. A scaling factor γ and a bias β are introduced for each channel of the BN layer. The normalization process of channel data in the BN layer is expressed as follows:

$$\hat{z} = \frac{(z_{in} - \mu_B)}{\sqrt{\sigma_B^2 + \epsilon}} \tag{2}$$

$$z_{out} = \gamma \hat{z} + \beta \tag{3}$$

Here, μ_B and σ_B are the mean and standard deviation of the input activations in batch

B. In CNNs, a common practice is to insert a BN layer after the convolution layer and utilize the channel scaling parameters for adjustment. In this regard, Slim introduces a scaling factor γ for each channel, multiplying this factor with the channel's output, after which the network weights are combined with these scaling factors, applying sparse regularization to γ. The loss function L is defined as follows:

$$L = \sum_{(x,y)} l(f(x,W),y) + \lambda \sum_{\gamma \in \Gamma} g \cdot (\gamma) \tag{4}$$

In this equation, $\sum_{(x,y)} l(f(x,W),y)$ represents the normal training loss function, where x and y denote the training input and output, and W represents the trainable weights. The term $\lambda \sum_{\gamma \in \Gamma} g \cdot (\gamma)$ is the regularization term, where $g(\gamma)$ is the penalty function for the scaling factors, with the L 1 norm selected: $g(\gamma) = |\gamma|$. Here, the weight coefficient λ acts as a balance factor, controlling sparsity.

When setting different sparse regularization values λ, the weights and mean accuracy of the model's BN layers exhibit corresponding changes. If λ is too small, the sparsity process is slow, failing to distinguish channel importance. Conversely, if λ is too large, the accuracy tends to drop too quickly. To maintain good recognition performance while conducting sparse training, the optimal sparsity rate is determined. Based on preliminary experimental data, λ was set to 0.0005, 0.001 ,0.005, and 0.01, and the original model underwent sparse training. The distribution changes of the BN layer scaling factors under different coefficients were visualized. It can be observed that the γ coefficients gradually approach zero as training progresses, and the larger the sparsity rate, the faster γ approaches zero.

After sparse training, the model becomes more compact, with many scaling factors approaching zero. Subsequently, for those channels with scaling factors near zero, all input and output connections and their corresponding weights are removed. The model is pruned based on different ratios. As the number of deleted channels increases (i.e., the percentage increases), it is necessary to choose an appropriate pruning rate during the pruning process.

After conducting 10 experiments with different pruning rates, it was observed that as the pruning rate increased, the number of model parameters gradually decreased. This decrease was particularly noticeable when the pruning rate was between 20% and 60%. When the pruning rate was below 50%, the average model accuracy remained around 97% with slight fluctuations. However, when the pruning rate exceeded 50%, the model's accuracy significantly dropped, making it unsuitable for real-time detection of leaf diseases. Therefore, to balance model accuracy with the degree of memory reduction, this paper selects 50% as the experimental pruning rate. After the experiments, it was found that, apart from a few critical channels that could not be pruned, most of the remaining channels underwent a certain degree of compression. This result indicates that the pruning algorithm is effective for this model.

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Data sources

The citrus leaf dataset in this study contains five categories of leaves: Black Spot, Melanose, Canker, Greening, and Healthy leaves. To reduce resource consumption during the network training process, all images in the dataset were compressed and uniformly adjusted to a resolution of 640×640 pixels. Since most of the acquired datasets were not annotated, this paper used Label Img software to manually label the pixel regions of the targets on RGB images. The diseased areas were marked with the maximum enclosing rectangle, ensuring that the rectangle closely adhered to the edges of the target pixels, and the labeling format adopted the YOLO format.

To enhance the adaptability and robustness of the training model and to prevent it from overfitting on a specific dataset, random noise, local cropping, and random rotations were applied to the original image collection, ultimately expanding the dataset to 4,558 images.

The hardware configuration for the model training and testing platform in this study includes a 14th generation Intel Core i9-14900K CPU, an NVIDIA GeForce RTX 2080TI GPU, and 22GB of video memory. The software configuration consists of Windows 11 operating system, PyTorch 2.0.1 deep learning framework, CUDA version 11.8, Python 3.8 programming language, and PyCharm integrated development environment. During training, the input image size for the model is set to 640 pixels × 640 pixels, with a batch size of 16 and a total of 100 epochs. The initial learning rate is set to 0.01, while the remaining parameters use their default values.

The edge computing devices used to test the model's actual deployment performance include a Jetson Xavier NX (manufacturer: NVIDIA, USA) and a Raspberry Pi 4B development board. The Jetson Xavier NX is equipped with a 6-core NVIDIA Arm 64-bit CPU, a 48-core NVIDIA Volta architecture GPU, and 8GB of RAM. The software environment includes Ubuntu 18.04 operating system, PyTorch 1.8 deep learning framework, and Python 3.6.9 programming language. The Raspberry Pi 4B development board features a 4-core 64-bit CPU and 4GB of RAM. Its operating environment includes Raspberry Pi OS, PyTorch 1.8.1 deep learning framework, and Python 3.9 programming language.

4.2 Comparison of detection performance of different attention mechanisms

To investigate the rationale for introducing the EMA attention mechanism, other representative attention mechanisms were selected for comparative experiments. During the experiments, each attention mechanism module was placed in the same position within the YOLOv8 network structure and tested on the validation set. Table 1 lists the results of the comparative experiments.

Table 1: Test results of the models with different attentionmechanisms

Models	P/%	R/%	mAP/%	Model size / MB
YOLOv8	88.6	80.7	87.7	6.0
C2fEMA—YOLOv8	90.6	80.9	87.9	6.0
CBAM—YOLOv8	85.7	77.8	86.0	6.0
SE—YOLOv8	91.0	79.8	87.5	6.0
SEAM—YOLOv8	90.8	77.3	87.5	6.1

When the EMA attention mechanism was introduced, the model's mean Average Precision (mAP) improved by 0.2 percentage points compared to the baseline. In contrast, when the CBAM, SE, and SEAM attention mechanisms were introduced, the model's mAP decreased by varying degrees compared to the baseline, specifically by 1.7, 0.2, and 0.2 percentage points, respectively.

Compared to EMA, although the CBAM attention mechanism considers both channel and spatial attention, it only focuses on local information without establishing long-range dependencies. This limitation affects its ability to balance local pest and disease features with the overall leaf context, thereby hindering the detection of diseased leaves in complex environments. The SE attention mechanism focuses solely on channel attention, which somewhat restricts the positioning and extraction capabilities for citrus pest and disease features. The SEAM attention mechanism may not reliably extract effective targets under complex scenes and extreme occlusion conditions.

Moreover, results indicate that adding attention mechanisms did not significantly increase the model size. In summary, the introduction of the EMA attention mechanism enhances feature extraction capability without enlarging the model size. The findings demonstrate that introducing the EMA attention mechanism is an effective approach to improving the model's accuracy in detecting citrus pests and diseases.

4.3 Ablation test

To investigate the impact of introducing the Slim-neck module and EMA attention mechanism on the performance of the developed citrus pest and disease detection model, ablation experiments were conducted. The test set results are shown in Table 2.

Table 2: Results of the ablation test

Models	P/%	R/%	mAP/%	mAP_{50-95}/%	Parameters /M
YOLOv8	88.6	80.7	87.7	75.3	3.0
YOLOv8＋Slim－neck	89.5	81.4	88.8	75.0	2.8
YOLOv8＋C2f__EMA	90.6	80.9	87.9	75.6	3.0
YOLOv8＋Slim－neck＋C2f __EMA	92.4	81.8	89.3	75.3	2.8

Analyzing the results in Table 2, it can be observed that after adding the Slim-neck module, the model's precision increased by 0.9 percentage points, the recall rate increased by 0.7 percentage points, and the mean Average Precision (mAP) increased by 1.1 percentage points, while the number of parameters decreased by 0.2 million. This improvement is attributed to the model reducing computational complexity by optimizing the connection methods of feature maps and minimizing redundant calculations. Simultaneously, the VoV-GSCSP module can extract features more effectively, thereby maintaining detection accuracy.

After introducing the EMA attention mechanism, the model's precision increased by 2 percentage points, and recall and mAP improved by 0.2 percentage points each. This enhancement occurs because the model can integrate features from different scales, paying more attention to the citrus targets with pests and diseases in complex backgrounds.

When both the Slim-neck module and EMA attention mechanism are added to the

YOLOv8 model, the precision, recall, and mAP of the model increased by 3.8, 1.1, and 1.6 percentage points, respectively, with the number of parameters decreasing by 0.2 million. The improved YOLOv8 model demonstrates significantly better object detection capabilities than the aforementioned models, with reduced instances of missed and false detections, while also having a substantially smaller parameter count compared to the baseline model. The implemented improvements effectively enhance the model's detection performance while reducing its size, allowing each modification to play its intended role.

4.4 Performance comparison of mainstream models

In order to study the difference between the improved YOLOv8 model and other models in the detection performance of citrus diseases and pests, this section selects several mainstream object detection models: Faster R-CNN, SSD, YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8 and YOLOv9 to carry out performance comparison tests. The test results are shown in Table III. Compared with other models, the accuracy and mAP results of the improved YOLOv8 model are higher, and the number of parameters and model size are smaller. Compared with Faster R-CNN, SSD, YOLOv3, YOLOv4, YOLOv5, YOLOv7 and YOLOv9 models, the mAP is increased by 3.8, 10.3, 13.5, 30.7, 5.3, 6.3 and 4.9 percentage points, respectively. The number of parameters is reduced by 38.5, 21.0, 58.7, 61.1, 4.2, 33.7 and 6.8M, respectively. The model size is reduced by 102.4, 86.5, 229.4, 238.4, 8.0, 65.7 and 13.7 MB, respectively.

Table 3: Detection performance comparison between differentmainstream object detection models on the test dataset

Models	*P*/%	*R*/%	mAP/%	Parameters / M	Model size / MB
Faster R－CNN	64.3	85.1	85.5	41.3	108.0
SSD	88.5	80.1	79.0	23.8	92.1
YOLOv3	82.7	66.4	75.8	61.5	235.0
YOLOv4	80.9	40.5	58.6	63.9	244.0
YOLOv5	86.9	76.5	84.0	7.0	13.6
YOLOv7	81.1	80.5	83.0	36.5	71.3
YOLOv8	88.6	80.7	87.7	3.0	6.0
YOLOv9	84.2	79.8	84.4	9.6	19.3
SEDS-YOLOv8	92.4	81.8	89.3	2.8	5.6

In order to verify the actual detection effect before and after the improvement of the model, multiple separately taken images of citrus diseases and pests are selected for detection, and the visualization results are shown in Figure 5.

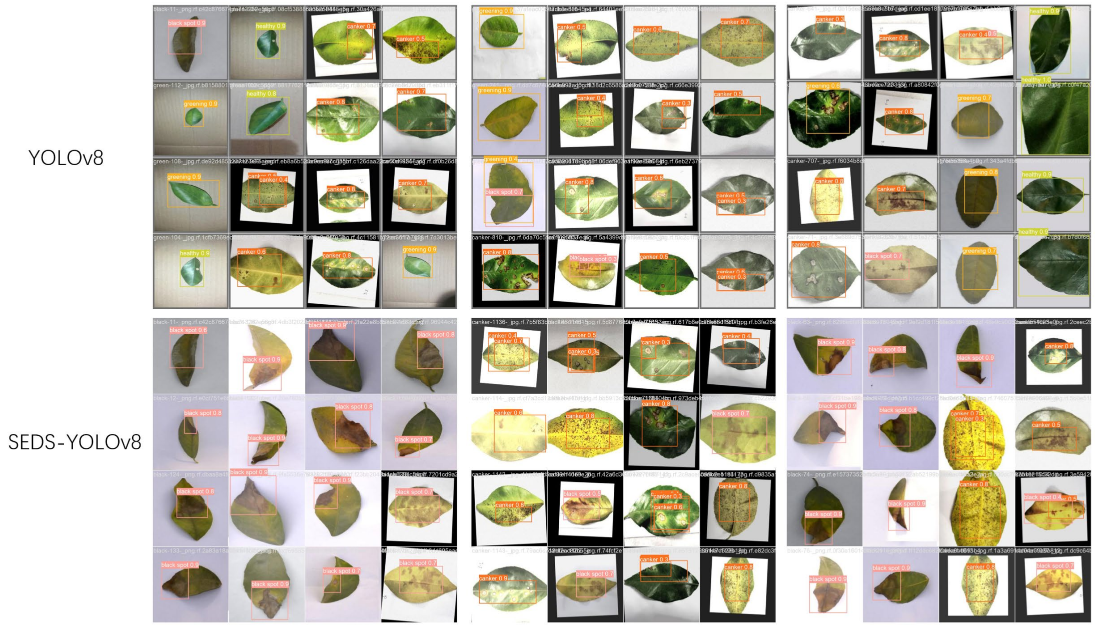


Fig. 5: Effect of citrus leaf detection

In the detection and recognition task of citrus diseases and pests, compared with the original model, the improved YOLOv8 model not only achieves good detection accuracy of citrus diseases and pests, but also has a good lightweight effect. In addition, the model shows a stronger ability to detect pests and diseases in the field.

4.5 Edge computing device deployment test analysis

In order to verify the actual deployment performance of the established improved YOLOv8 model on the mobile terminal in the field, the model, the baseline model and the YOLOv5 model with more field deployment reports were deployed on the edge computing devices Jetson XavierNX and Raspberry PI 4B, and the test experiment was carried out. The model checking frame rate results are shown in Table IV.

Table 4: Detection frame rate of the model implemented on theedge computing devices

Device	YOLOv5	YOLOv8	Improved YOLOv8
Jetson Xavier NX	18.1	23.9	27.0
Raspberry PI 4B	0.4	0.7	0.7

The detection frame rate achieved by the improved YOLOv8 model on Jetson Xavier NX and Raspberry PI 4B is 27.0 and 0.7 frames /s, respectively. The detection frame rate of YOLOv8 model implemented on Jetson Xavier NX and Raspberry PI 4B is 23.9 and 0.7 frames /s, respectively. The detection frame rates of YOLOv5 model implemented on Jetson Xavier NX and Raspberry PI 4B are 18.1 and 0.4 frames /s, respectively. Compared with the YOLOv5 model, the frame rate of the improved YOLOv8 model is increased by 8.9 and 0.3 frames /s on Jetson XavierNX and Raspberry PI 4B, respectively. The above results show that the improved YOLOv8 model has good prospects for mobile terminal deployment, which can provide technical support for the research and development of efficient intelligent detection and intelligent control equipment of citrus diseases and pests.

5 CONCLUSION

This study proposes a lightweight citrus pest and disease detection model based on the improved YOLOv8, and establishes a citrus pest and disease dataset considering different environmental conditions, providing a new approach for the intelligent detection of citrus pests and diseases and the on-site deployment of equipment. In the improved YOLOv8 model, lightweight convolution GSConv and VoV-GSCSP modules are introduced to form a Slim-neck lightweight architecture. Additionally, the EMA attention mechanism is incorporated into the model to form the C2f_EMA module, replacing the C2f module in the model's Backbone. This enables the model to maintain high accuracy while reducing computational complexity and enhancing its ability to extract and fuse multi-scale spatial features of citrus pests and diseases. To improve the model's robustness to data, the MLCA attention mechanism is added, which reduces complexity without significantly lowering the model's accuracy. Finally, the model is pruned to further compress it. The precision, recall, and mAP of the model are 92.4%, 81.8%, and 89.3% respectively, enabling accurate and effective identification of citrus pests and diseases. Compared with the current mainstream models such as Faster R-CNN, SSD, YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, and YOLOv9, the improved YOLOv8 model achieves higher mAP, fewer parameters, and a smaller model size. It not only achieves good detection accuracy for citrus pests and diseases but also has a better lightweight effect, making it more advantageous in the detection of citrus pests and diseases in complex backgrounds. The improved YOLOv8 model has been deployed and tested on edge computing devices Jetson Xavier NX and Raspberry Pi 4B, achieving detection frame rates of 27.0 and 0.7 frames per second respectively. Its deployment performance is superior to that of the mainstream YOLOv5 model and the YOLOv8 baseline model, effectively meeting the requirements for lightweight deployment of intelligent pest and disease control equipment.

The improved YOLOv8 model proposed in this study achieves precise identification of citrus pests and diseases, and has a promising prospect for mobile deployment. It can provide technical support for the development of automatic pest and disease control operations and intelligent pest and disease control equipment for citrus. In the future, we will collect more images of citrus leaf pests and diseases under occlusion or different lighting conditions to enrich the dataset, and further analyze the features of the input data and observe the network structure. Additionally, we will compare the use of multiple pruning algorithms to perform secondary compression on the model, so as to achieve further lightweighting of the model while improving the accuracy of object detection.

6 DATA SOURCES

The article includes some data to support the results of this research. The dataset for this article is available at https://github.com/jinmuxige0816/orange

7 ACKNOWLEDGEMENTS

Thanks for the data support provided by National-level Innovation Program Project Fund "Research on Seedling Inspection Robot Technology Based on Multi-source Information Fusion and Deep Network" (No.: 202410451009); Jiangsu Provincial Natural Science Research General Project (No.: 20KJB530008); China Society for Smart Engineering "Research on Intelligent Internet of Things Devices and Control Program Algorithms Based on Multi-source Data Analysis" (No.: ZHGC104432); China Engineering Management Association "Comprehensive Application Research on Intelligent Robots and Intelligent Equipment Based on Big Data and Deep Learning" (No.: GMZY2174); Key Project of National Science and Information Technology Department Research Center National Science and Technology Development Research Plan (No.: KXJS71057); Key Project of National Science and Technology Support Program of Ministry of Agriculture (No.: NYF251050).

REFERENCES

[1] Testa, R., Tudisca, S., Schifani, G., Di Trapani, A. M., & Migliore, G. (2018). Tropical fruits as an opportunity for sustainable development in rural areas: The case of mango in small-sized Sicilian farms. *Sustainability*, *10*(5), 1436.

[2] George, A., Rao, C., & Mani, M. (2022). Pests of citrus and their management. *Trends in Horticultural Entomology*, 551-575.

[3] Panno, S., Davino, S., Caruso, A. G., Bertacca, S., Crnogorac, A., Mandić, A., Noris, E., & Matić, S. (2021). A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the mediterranean basin. *Agronomy*, *11*(11), 2188.

[4] Hajji-Hedfi, L., & Chhipa, H. (2021). Nano-based pesticides: challenges for pest and disease management. *Euro-Mediterranean Journal for Environmental Integration*, *6*(3), 69.

[5] Zahoor, I., & Mushtaq, A. (2023). Water pollution from agricultural activities: A critical global review. *Int. J. Chem. Biochem. Sci*, *23*(1), 164-176.

[6] Antonio, M., Alcaraz, M. R., & Culzoni, M. J. (2024). Advances on multiclass pesticide residue determination in citrus fruits and citrus-derived products–A critical review. *Environmental Science and Pollution Research*, *31*(38), 50012-50035.

[7] Naqvi, S. A. H., Wang, J., Malik, M. T., Umar, U.-U.-D., Hasnain, A., Sohail, M. A., Shakeel, M. T., Nauman, M., Hassan, M. Z., & Fatima, M. (2022). Citrus canker—Distribution, taxonomy, epidemiology, disease cycle, pathogen biology, detection, and management: A critical review and future research agenda. *Agronomy*, *12*(5), 1075.

[8] Domingues, T., Brandão, T., & Ferreira, J. C. (2022). Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey. *Agriculture*, *12*(9), 1350.

[9] Yao, X., Lin, H., Bai, D., & Zhou, H. (2024). A Small Target Tea Leaf Disease Detection Model Combined with Transfer Learning. *Forests*, *15*(4), 591.

[10] Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(1), 113.

[11] Wang, J., & Wang, M. (2021). Review of the emotional feature extraction and classification using EEG signals. *Cognitive robotics*, *1*, 29-40.

[12] Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. *Information processing in agriculture*, *8*(1), 27-51.

[13] Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern recognition letters*, *141*, 61-67.

[14] Gai, R., Liu, Y., & Xu, G. (2024). TL-YOLOv8: A blueberry fruit detection algorithm based on improved YOLOv8 and transfer learning. *Ieee Access*.

[15] Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., Wu, Y., & Zhang, Y. (2022). Deep learning based detector YOLOv5 for identifying insect pests. *Applied Sciences*, *12*(19), 10167.

[16] Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, *8*(1), 66.

[17] Zhang, H., & Cloutier, R. S. (2021). Review on one-stage object detection based on deep learning. *EAI Endorsed Transactions on e-Learning*, *7*(23), e5-e5.

[18] Wang, X., Gao, H., Jia, Z., & Li, Z. (2023). BL-YOLOv8: An improved road defect detection model based on YOLOv8. *Sensors*, *23*(20), 8361.