

Tomato Leaf Disease Classification Using EfficientNet Enhanced with RefConv Structural Reparameterization, Efficient Channel Attention, and Lion Optimizer

Lijuan Shi^{1,3}, Haitang Li², Shiwen Zhang^{1,3}

¹College of Information Engineering, Zhengzhou University of Technology, Zhengzhou 450052, China

²College of Information Engineering, Zhoukou Vocational College of Arts and Sciences, Zhoukou 466000, China

³Henan Province Engineering Research Center for Large-scale AI Models and Applications

*Corresponding author

Keywords: Tomato disease recognition, EfficientNet, structural reparameterization, efficient channel attention, lion optimizer, agricultural image classification

Received: Feb 26, 2026

Fine-grained classification of tomato leaf diseases faces challenges due to high inter-class similarity and insufficient model generalization ability. To address these issues, this paper proposes an improved EfficientNet architecture that fuses structural reparameterization, a lightweight channel attention mechanism, and an efficient optimization strategy. Specifically, the reparameterized refocus convolution (RefConv) module is introduced to enhance the extraction ability of subtle lesion textures. The original Squeeze-and-Excitation (SE) module is replaced with the efficient channel attention (ECA) mechanism to achieve lightweight channel feature recalibration. Meanwhile, the Lion optimizer is adopted to accelerate convergence and improve generalization performance on complex agricultural images. Experimental results show that the proposed method achieves 98.44% accuracy on the self-built tomato disease dataset, which is 4.22% higher than that of the baseline model EfficientNet-B0. Cross-species generalization tests further obtain 98.43% and 99.51% accuracy on apple and grape leaf disease datasets, respectively, verifying the robustness and transferability of the model.

Povzetek: Raziskava predstavlja izboljšan model EfficientNet za zelo natančno in robustno prepoznavanje bolezni na listih paradižnika ter drugih rastlin.

1 Introduction

Tomato, as one of the most widely cultivated economic crops worldwide, is susceptible to a variety of common diseases and insect pests, which can lead to yield losses of 20%–100%, severely restricting the sustainable development of agriculture and food security. Traditional manual identification methods rely on expert experience and suffer from low efficiency, strong subjectivity, high cost, and limited coverage, making it difficult to meet the needs of large-scale modern agricultural production.

In recent years, deep learning technology has shown great potential in the intelligent identification of crop diseases and insect pests, becoming a core driving force for the development of precision agriculture. It has been shown that many advances have also been made in intelligent identification of tomato diseases and pests. S. P. Mohanty et al. first demonstrated the superiority of deep convolutional neural networks (CNNs) in plant disease detection and released the PlantVillage dataset, significantly promoting the rise of this field [1]; J. Eunice et al. utilized pre-trained CNN models and focused on fine-tuning hyperparameters for efficient plant disease identification, achieving high classification accuracy [2]; O. Attallah proposed a lightweight scheme based on transfer learning, optimized the network structure through feature selection, and demonstrated high classification

accuracy [3]. However, most of these early studies relied on laboratory-controlled datasets, leaving a gap in real-field applicability. M. Jelali focused on the application of real field datasets and reviewed the adaptability and detection performance of state-of-the-art models including YOLOv8 and Swin Transformer in actual crop environments [4]; L. K. Ndovie et al. applied MobileNetV3 to an in-field tomato disease detection project in Malawi, but found that models trained on PlantVillage data generalized poorly to real-world field conditions [5].

Meanwhile, to improve feature extraction and model efficiency, several lightweight and attention-based enhancements have been proposed. M. Qabulio et al. proposed a tomato leaf disease identification model based on MobileNetV3Small, achieving 99.43% accuracy on a public dataset, demonstrating the effectiveness of lightweight CNNs for tomato disease classification [6]; R. Rashid, W. Aslam, et al. enhanced feature extraction ability by introducing inverted residuals and a channel attention mechanism, improving the recognition accuracy of small lesions under complex backgrounds [7]; H. Sun et al. introduced a multi-scale feature diffusion interaction module into the Neck structure of YOLOv8, effectively improving the detection ability for lesions of different sizes [8]; W. Hou et al. combined super-resolution technology with YOLOv8 to improve the quality of

original images, enabling the model to detect finer lesion features [9]; S. U. Islam et al. integrated object detection of YOLOv8 with the segmentation capability of Meta AI's Segment Anything Model (SAM), realizing the leap from detection to precise segmentation [10]. These YOLOv8-based improvements focus on enhancing multi-scale feature fusion and detection precision. On the other hand, several studies have explored Transformer architectures for tomato disease recognition. Y. Cao improved the structure of Swin Transformer and utilized the advantages of hierarchical feature extraction to improve recognition accuracy in complex texture images [11]; Y. Sun proposed a hybrid architecture of EfficientNetV2 and Swin Transformer, combining the local feature advantages of CNNs and the global modeling ability of Transformers to achieve high-precision classification [12]; S. Yu added Convolutional Feature Interactions inside Swin Transformer, solving the ambiguity problem that may occur when Transformers process detailed features [13]. More recent 2026 hybrid architectures such as EcoTomHybridNet reduce inference cost via dynamic branch allocation, yet still rely on dual-branch designs [14]. However, these Transformer-based methods still suffer from high computational overhead, while earlier approaches such as HFFIncep-Net[15] also lack dedicated mechanisms for fine-grained texture modeling.

Consequently, existing models face three major limitations: they cannot capture subtle lesion textures without extra inference cost, they rely on computationally expensive attention mechanisms, and they lack systematic solutions for training stability and convergence. To address these limitations, the objectives of this paper are as follows: (1) to enhance fine-grained texture modeling of lesion features without increasing inference cost; (2) to reduce attention mechanism redundancy while maintaining channel recalibration efficiency; (3) to improve training stability and convergence speed for agricultural image classification.

Therefore, the objective of this study is to improve the fine-grained classification accuracy of tomato leaf diseases while maintaining computational efficiency and cross-species generalization capability. To achieve this objective, this paper proposes an improved EfficientNet framework that integrates three key technologies:

A RefConv module based on structural reparameterization to strengthen texture modeling;

An Efficient Channel Attention (ECA) mechanism to achieve channel feature recalibration with extremely low computational overhead;

The Lion optimizer to stabilize the training process and accelerate convergence.

The constructed model maintains high accuracy while exhibiting excellent cross-species generalization ability.

2 Dataset construction and preprocessing

2.1 Dataset construction

In this study, a comprehensive dataset containing 11 categories of tomato leaf images was constructed for fine-grained disease classification. The categories include healthy leaves and 10 common diseases: bacterial spot, early blight, late blight, leaf mold, powdery mildew, target spot, two-spotted spider mite damage, Septoria leaf spot, tomato mosaic virus, and tomato yellow leaf curl virus. All images were obtained from the standardized PlantVillage dataset, with a total of 28,753 samples.

2.2 Stratified dataset partitioning

To alleviate the problem of class imbalance during training, a stratified sampling strategy was adopted. The dataset was divided into training, validation, and test sets with a fixed ratio of 7:1:2, ensuring a consistent distribution of each class across all subsets. Specifically, the training set contains 19,430 images for parameter optimization, the validation set contains 2,709 images for hyperparameter tuning and overfitting monitoring, and the test set contains 6,614 images for final performance evaluation.

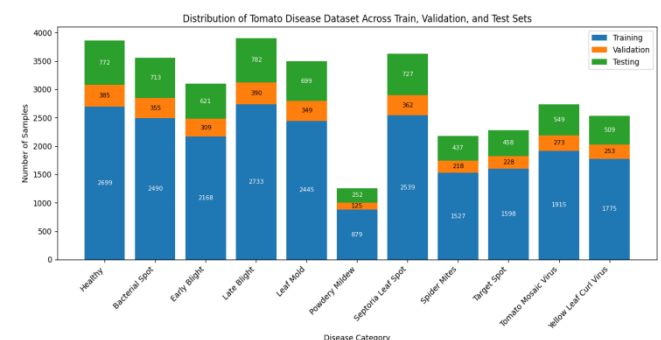


Figure 1: Distribution characteristics of sample sizes across categories in different subsets.

Among them, the training set contains 19,430 samples and is used for model parameter optimization; the validation set contains 2,709 samples and is used to monitor overfitting during training and adjust hyperparameters; the test set contains 6,614 samples and is used to independently evaluate the final generalization performance of the model.

2.3 Data preprocessing

To improve feature extraction capability and model generalization, multi-stage preprocessing was implemented:

Image Standardization: All images were uniformly resized to 224×224 pixels, and pixel intensities were normalized to the range $[0, 1]$ to reduce the influence of illumination and scale variations.

Data Augmentation: Random transformations were applied to the training set, including horizontal flip (probability = 0.5), rotation ($\pm 15^\circ$), brightness adjustment (± 0.2), and contrast adjustment (± 0.2). These operations effectively enlarged the effective sample size of minority classes (e.g., powdery mildew, two-spotted spider mite damage) and alleviated overfitting.

Noise and Background Removal: For field-collected images, Gaussian filtering ($\sigma = 1.5$) was used to suppress environmental noise, and a background segmentation algorithm was combined to eliminate irrelevant regions (e.g., soil, weeds), thereby improving the visibility of lesion areas.

3 Model structure and improvements

3.1 Overview of the EfficientNet baseline model

EfficientNet [16] is a series of convolutional neural networks that achieve an optimal balance between accuracy and efficiency through systematic model scaling. Its core contribution lies in the proposed Compound Scaling principle. Traditional methods usually increase network depth, width, or input resolution individually. However, the authors of EfficientNet demonstrated through extensive experiments that these three factors are interdependent.

Blindly scaling a single dimension leads to rapid saturation of performance gains, whereas coordinated scaling can bring significant performance improvements. Specifically, compound scaling uses a unified compound coefficient ϕ to coordinate the scaling ratios of network width (number of channels), depth (number of layers), and resolution (image size). The method is formalized mathematically as follows: under given computational resource constraints (e.g., FLOPs), the goal is to achieve optimal model accuracy. EfficientNet formulates the scaling problem as:

$$\begin{aligned} & \max_{d,w,r} \text{Accuracy}(N(d,w,r)) \\ & \text{s.t. } \text{FLOPs}(N) \approx 2^\phi \cdot \text{FLOPs}_{\text{base}} \quad (1) \\ & d = \alpha^\phi, \quad w = \beta^\phi, \quad r = \gamma^\phi \\ & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \end{aligned}$$

where (d,w,r) are the scaling coefficients for depth, width, and resolution, respectively; (α,β,γ) are constants determined via grid search on a small baseline network; and ϕ is a user-specified global scaling coefficient that controls the overall model size. This principle ensures that the structural dimensions of the model grow cooperatively at an optimal ratio during scaling.

At the micro-structural level, the baseline network of EfficientNet (such as EfficientNet-B0) adopts the MBConv (Mobile Inverted Bottleneck Convolution) module as its core building block. As shown in Figure 1, each MBConv module consists of three stages: first, a 1×1 expansion convolution to increase the feature channel dimension (usually expanded by a factor of 6); second, a

$k \times k$ depthwise separable convolution (k is typically 3 or 5) for lightweight spatial feature extraction; and finally, a 1×1 projection convolution to compress the channel dimension back to the output size. The module is usually also integrated with a Squeeze-and-Excitation (SE) attention module to enhance feature representation through channel-wise weight recalibration.

This paper uses EfficientNet-B0 as the baseline model. Its original design goal is to balance high performance and high efficiency on mobile devices, which is highly compatible with the limited edge computing resources often encountered in agricultural disease recognition scenarios. However, when directly applying this general-purpose model to the fine-grained classification task of tomato leaf diseases with high inter-class similarity, there are still three main limitations:

Standard convolutional layers have limited ability to extract features from subtle lesion textures;

The number of parameters and computational overhead of the SE attention module can be further optimized;

The training convergence speed and generalization ability of the model on complex agricultural image data need to be improved.

To address these challenges, this paper proposes three improvements, aiming to fully exploit the disease classification potential without significantly increasing the inference complexity of the model.

3.2 RefConv convolution module

To further enhance the model's ability to extract fine texture features of tomato leaf diseases without increasing inference latency, this study adopts Reparameterized Refocusing Convolution (RefConv) [17] to replace several key convolutional layers in EfficientNet. Specifically, except for all 1×1 pointwise convolutions, the original 3×3 expansion convolution, $k \times k$ depthwise convolution in the MBConv block, and the initial stem convolution of the network are all replaced with the RefConv module.

The core idea of RefConv is to refocus the convolution kernels of the pre-trained model rather than adding extra training branches. For a pre-trained convolutional layer, its weights are frozen and used as the base weight (W_b). RefConv applies a learnable refocusing transformation ($T(\cdot)$) on (W_b) to generate the transformed weight (W_r) for forward propagation. The process can be expressed as:

$$W_t = T(W_b; W_r) = W_b * W_r + W_b \quad (2)$$

where (W_r) denotes the learnable parameters introduced in the transformation, and $(*)$ denotes the convolution operation. A $k \times k$ convolution (default $k=3$) is used as the transformation function to establish cross-channel connections in the weight space. This design enables each output kernel channel to fuse information from all input kernel channels, thereby refocusing on the multi-channel representations learned by the pre-trained model.

In the training phase (referred to as refocus learning), the base weight (W_b) is frozen, and only the transformation parameter (W_r) is optimized. Forward propagation is performed using (W_t):

$$\mathbf{Y}=\mathbf{X}*\mathbf{W}_t \quad (3)$$

After training, only the final fused ($\mathbf{W}_t$) is saved and deployed. Therefore, the model structure during inference is identical to that of the original model, without introducing any extra parameters or computational overhead. By reducing channel redundancy and enhancing the diversity of weight representations, this module effectively improves the model's ability to model lesion edges and complex textures, and demonstrates stable performance gains in subsequent ablation experiments.

3.3 ECA attention mechanism

In fine-grained agricultural disease classification, discriminative features are often concentrated in specific feature channels. To guide the model to focus on these channels more efficiently, this study replaces the SE module in the original MBConv block with the Efficient Channel Attention (ECA) module [18]. The core idea of ECA is to avoid channel dimension reduction and implement local cross-channel interaction via one-dimensional convolution, thereby significantly reducing the number of parameters while maintaining or even improving performance.

Given an input feature ($\mathbf{X} \in \mathbb{R}^{W \times H \times C}$), the ECA module first aggregates spatial information through Global Average Pooling (GAP) to obtain a channel descriptor vector ($\mathbf{y} \in \mathbb{R}^C$):

$$y = \text{GAP}(\mathbf{X}), \quad y_c = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H x_{i,j,c} \quad (4)$$

Where (y_c) denotes the global average value of the (C) channel.

Different from the SE module, which uses fully connected layers with dimension reduction, ECA operates directly on ($\mathbf{y}$). It captures local cross-channel dependencies through a fast 1D convolution and generates the final channel weight vector ($\omega \in \mathbb{R}^C$) via the Sigmoid function:

$$\omega = \sigma(\text{C1D}_k(\mathbf{y})) \quad (5)$$

Here, (C1D_k) denotes the 1D convolution with kernel size (k), and (σ) is the Sigmoid activation function. The key advantage of this 1D convolution lies in its local interaction property: for the weight (ω_i) of the (i) channel, its computation only depends on the (k) adjacent channels centered at (y_i) (i.e., ($y_{i-|k/2|}, \dots, y_i, \dots, y_{i+|k/2|}$)). This strategy is proven to effectively model the interactions among channels while drastically reducing the number of parameters from ($O(C^2)$) in the SE module to only (k).

To enable the interaction range to adapt to convolutional blocks with different numbers of channels (C), ECA uses a nonlinear mapping function to determine the optimal convolution kernel size (k):

$$k = \psi(C) = \left\lfloor \frac{\log_2(C)}{\gamma} + b \right\rfloor \text{ odd} \quad (6)$$

where $\psi(C)$ is a function for adaptively determining the 1D convolution kernel size (k), ($\lfloor t \rfloor \text{ odd}$) denotes the nearest odd number to (t), and (γ) and (b) are hyperparameters (set as ($\gamma=2, b=1$) by default in the original paper). This design ensures that a larger channel number (C) leads to a wider

adaptively selected interaction range (k), thus guaranteeing the universality and effectiveness of the module across different network layers.

Finally, the original feature map ($\mathbf{X}$) completes feature recalibration by element-wise multiplication with the channel weight vector (ω):

$$\tilde{\mathbf{X}} = \mathbf{X} \otimes \omega \quad (7)$$

where ($\otimes$) denotes the channel-wise multiplication operation, and ($\tilde{\mathbf{X}}$) is the output feature after recalibration. In the model proposed in this paper, the ECA module is placed after the depthwise convolution in each MBConv block. This position allows it to directly perform channel-level importance evaluation on spatially filtered features. Since it avoids dimension reduction and only introduces (k) learnable parameters, ECA achieves better channel attention modeling than the SE module with almost no increase in model complexity.

3.4 Lion optimizer

The choice of optimization algorithm is critical to the final model performance. To address the common challenges in agricultural images, such as class imbalance, complex backgrounds, and weak early disease features, this study adopts the Lion optimizer [19] for model training. Lion is an efficient symbol-function-based optimizer whose update rule combines momentum and weight decay, with a simple form and low memory usage.

In each iteration, Lion first updates the exponential moving average of gradient momentum ($\mathbf{m}_t$):

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t \quad (8)$$

where ($\mathbf{g}_t$) is the current gradient, and (β_1) is the momentum decay factor.

The parameter update is jointly determined by the sign function and the weight decay term:

$$\theta_t = \theta_{t-1} - \eta_t \cdot [\text{sign}(\mathbf{m}_t) + \lambda \theta_{t-1}] \quad (9)$$

where (η_t) is the learning rate, (λ) is the weight decay coefficient, and ($\text{sign}(\cdot)$) binarizes the update direction.

This design provides an implicit gradient clipping effect, improving training stability while avoiding the memory cost of maintaining second-order moments in Adam-like optimizers. Subsequent ablation experiments also show that, compared with traditional optimizers, Lion enables the improved model to converge faster to better generalization performance and significantly increases the accuracy of disease recognition.

4 Experiments

4.1 Experimental configuration

The experiments were conducted on a Windows 11 operating system, equipped with an NVIDIA RTX A4000 GPU (16 GB memory), an Intel(R) Core (TM) i9-14900K CPU (3.20 GHz), and 128 GB of RAM. The model was implemented based on the PyTorch 2.3.0 framework with Python 3.12 as the programming language.

All images of tomato diseased leaves were resized to 224×224 pixels. The dataset was randomly partitioned into training, validation, and test sets at a ratio of 7:1:2, which

were used for model training, validation, and testing, respectively.

Considering the specific conditions of the experiment, the Lion optimization algorithm was adopted with a batch size of 96, a total training epoch of 60, an initial learning rate of 0.0001, and a weight decay coefficient set to 0.005.

4.2 Experimental evaluation metrics

To comprehensively evaluate the classification performance of the model in tomato leaf disease recognition, in addition to accuracy, this study further introduces Precision, Recall, and F1-score as evaluation metrics to reflect the discriminative ability of the model on each category from multiple perspectives, ensuring scientific and comparable evaluation results. Meanwhile, the number of Parameters and floating-point operations (FLOPs) are used to measure model complexity.

The definitions of each metric are given as follows:

Precision: measures the proportion of actual positive samples among those predicted as positive by the model, reflecting the precision of the model. The calculation formula is:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

where (TP)(True Positive) denotes the number of samples correctly predicted as positive, and (FP)(False Positive) denotes the number of samples incorrectly predicted as positive.

Recall: measures the proportion of actual positive samples that are correctly predicted as positive by the model, reflecting the completeness of detection. The calculation formula is:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (11)$$

where (FN) (False Negative) denotes the number of positive samples incorrectly predicted as negative.

F1-score: the harmonic means of precision and recall, used to comprehensively balance precision and recall performance. It is especially valuable when the category distribution is imbalanced. The calculation formula is:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Accuracy: reflects the overall proportion of samples correctly predicted by the model, serving as a basic metric for evaluating the overall performance of classification models. The calculation formula is:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

where (TN) (True Negative) denotes the number of samples correctly predicted as negative.

4.3 Result analysis

4.3.1 Comparative experiments

To verify the effectiveness and superiority of the improved model proposed in this paper, comparative experiments were conducted on the unified tomato leaf disease dataset with several mainstream deep learning models, such as GoogLeNet [20], ShuffleNetV2 [21], Res2Net50 [22], DenseNet121 [23], MobileNetV2 [24], RegNet [25], GhostNet [26], and the original EfficientNet [16]. Accuracy, Precision, Recall, and F1-score were adopted as evaluation metrics. The detailed results are shown in Table 1.

Table 1: Accuracy comparison among different models.

Model	Precision	Recall	F1-score	Accuracy	Param(M)	FLOPs(G)
GoogLeNet	95.92%	95.76%	95.81%	95.93%	10.33	1.58
ShuffleNetV2	83.48%	81.25%	82.04%	82.64%	0.35	0.04
Res2Net50	94.99%	94.99%	94.69%	94.82%	90.30	182.33
DenseNet121	95.10%	95.01%	94.99%	95.29%	6.96	2.91
MobileNetV2	90.66%	90.25%	90.34%	90.61%	2.23	0.33
RegNet	93.41%	92.57%	92.94%	93.10%	2.32	0.21
GhostNet	94.33%	93.16%	93.66%	93.73%	3.91	0.15
EfficientNet	94.40%	94.03%	94.18%	94.22%	4.02	0.42
Ours	98.55%	98.59%	98.57%	98.44%	70.89	1.84

Overall Performance Analysis: The proposed method (Ours) outperforms all other comparison models significantly on all evaluation metrics. Specifically, the accuracy reaches 98.44%, while precision, recall, and F1-score achieve 98.55%, 98.59%, and 98.57%, respectively, demonstrating excellent comprehensive classification capability.

Group-wise Comparative Analysis: From the experimental results:

Classic models such as GoogLeNet and Res2Net-50 achieve approximately 95% accuracy due to their deep structures, showing robust feature extraction ability.

Lightweight models including MobileNetV2 and GhostNet have obvious advantages in parameters and computational cost, but their accuracy is relatively low

(90.61%–93.73%), indicating limited representation ability in complex fine-grained classification tasks.

The original EfficientNet, as an efficient architecture, obtains 94.22% accuracy, yet its performance still has room for improvement.

The results show that the proposed model achieves the best recognition performance while maintaining reasonable parameters (70.89M) and computational cost (1.84G FLOPs). Compared with the original EfficientNet-B0, the accuracy is improved by 4.22%. This validates the effectiveness of the improved strategies proposed in this paper, which can better exploit the model potential in fine-grained agricultural disease classification tasks.

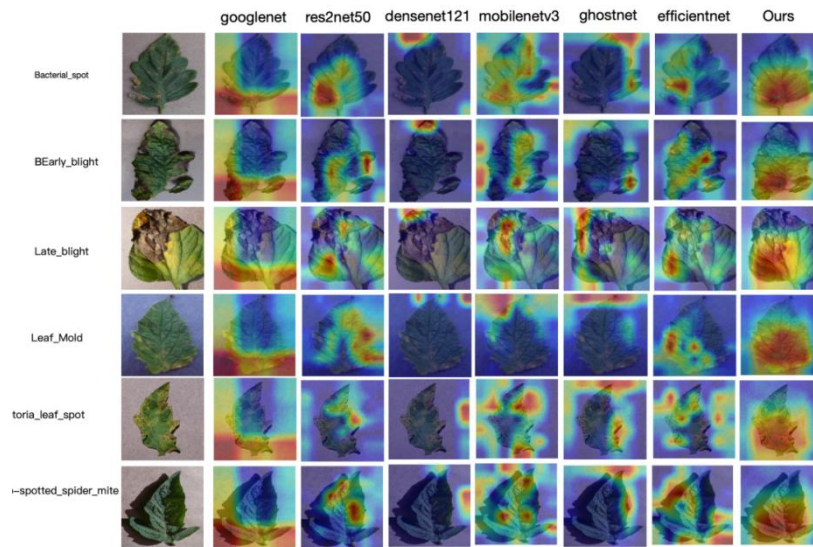


Figure 2: Comparison of heatmaps among different models

The feature activation heatmaps generated using the Grad-CAM method, as shown in Figure 2, reveal that compared with classic models such as GoogLeNet and ResNet34, the proposed model focuses its activation regions more intensively and accurately on discriminative feature areas such as lesion edges and texture variations for multiple disease categories including bacterial spot, early blight, and late blight, while significantly reducing attention to irrelevant regions such as leaf backgrounds or healthy tissues. Although lightweight models such as

MobileNetV3 and GhostNet possess a certain degree of feature localization capability, their heatmap activation regions are relatively scattered or poorly localized for diseases with complex details such as leaf mold and Septoria leaf spot, indicating limitations in model capacity or feature extraction ability when facing subtle lesions. This comparison intuitively verifies the effectiveness of the RefConv and ECA modules introduced in this paper in enhancing the model's ability to extract and focus on fine disease features.

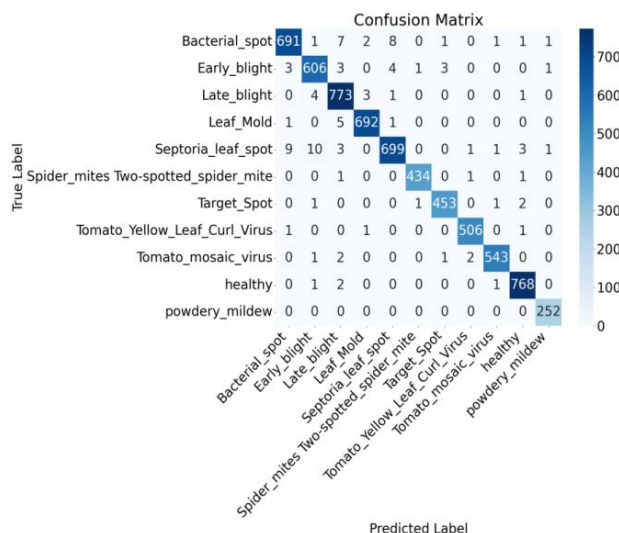


Figure 3: Results of the confusion matrix

As can be seen from Figure 3, the confusion matrix of the model on the test set clearly reveals its classification details and main error patterns. Overall, the model achieves high classification accuracy for most categories. For instance, late blight, leaf mold, and early blight were correctly identified 708, 677, and 606 times, respectively, and healthy leaves were also accurately distinguished.

However, limited confusion exists between some categories with similar visual characteristics. For example, a small number of samples of bacterial spot were misclassified as early blight or late blight; many samples of Septoria leaf spot were misclassified into categories such as early blight, which may be due to the relatively small sample size of this category or insufficient feature discrimination in the dataset.

These results demonstrate that the proposed model has highly reliable recognition ability for major disease categories and robust overall classification performance. For a few easily confused categories, further optimization can be carried out in the future by enhancing sample balance or introducing more targeted local feature learning mechanisms.

4.3.2 Ablation study

To verify the specific contributions of each improved module introduced in this paper (ECA channel attention mechanism, RefConv improved convolution module, and Lion optimizer) to model performance, as well as to explore their synergistic mechanism, a systematic ablation study is designed in this section using EfficientNet as the baseline model. The experimental results are shown in Table 3. By gradually adding different modules, the influence of each component on the recognition accuracy of tomato leaf diseases is quantitatively analyzed.

Table 2: Accuracy comparison among improved models.

Model	eca	refconv	lion	Param(M)	FLOPs(G)	Accuracy
EfficientNet				4.02	0.42	94.22%
EfficientNet RefConv		√		71.53	1.84	95.24%
EfficientNet Lion			√	4.02	0.42	95.63%
EfficientNet ECA	√			3.38	0.42	95.00%
EfficientNet RefConv ECA	√	√		70.89	1.84	95.47%
EfficientNet RefConv ECA Lion	√	√	√	70.89	1.84	98.44%

The results of the ablation study are shown in Table 2. Taking EfficientNet as the baseline model (94.22%), its number of parameters is 4.02 M and the computational cost is 0.42 GFLOPs. When the RefConv module is introduced alone, the accuracy reaches 95.24%, while the number of parameters and computational cost increase significantly to 71.53 M and 1.84 GFLOPs, indicating that this structure cannot be fully adapted to the baseline when used independently. Adding the ECA module alone can improve the accuracy to 95.00% with a slight decrease in parameters (3.38 M), demonstrating that it can effectively enhance feature representation efficiency. Using the Lion optimizer alone increases the accuracy to 95.63%, highlighting its role in training optimization. The results of the ablation study are shown in Table 2. Taking EfficientNet as the baseline model (94.22%), its number of parameters is 4.02 M and the computational cost is 0.42 GFLOPs. When the RefConv module is introduced alone, the accuracy reaches 95.24%, while the number of parameters and computational cost increase significantly to 71.53 M and 1.84 GFLOPs, indicating that this structure cannot be fully adapted to the baseline when used independently. Adding the ECA module alone can improve the accuracy to 95.00% with a slight decrease in parameters (3.38 M), demonstrating that it can effectively enhance feature representation efficiency. Using the Lion optimizer alone increases the accuracy to 95.63%, highlighting its role in training optimization.

Combining ECA and RefConv further improves the model accuracy to 95.47%, reflecting their positive synergy in feature extraction. Finally, the complete model integrating ECA, RefConv, and Lion achieves the best accuracy of 98.44% with 70.89 M parameters and 1.84 GFLOPs computational cost, which is significantly superior to other combinations.

The results show that the improved strategies adopted in this paper realize the synergistic enhancement of model representation ability and convergence performance through the deep integration of attention mechanism, convolutional structure optimization, and training strategy, thereby achieving significant and balanced performance improvement in tomato leaf disease recognition.

4.3.3 Generalization experiment

To verify the generalization ability and adaptability of the proposed model in other plant leaf disease recognition tasks, a generalization experiment was conducted on a public apple leaf disease dataset. This dataset contains four categories: apple scab, black rot, cedar apple rust, and healthy leaves. Several mainstream models, including GoogLeNet, ResNet50, DenseNet121, and EfficientNet, were selected for comparison. The experimental results were obtained by retraining the proposed model on the public apple leaf disease dataset.

Table 3: Performance comparison of different models on the apple leaf disease dataset.

Model	Precision	Recall	F1-score	Accuracy
GoogLeNet	98.53%	96.90%	97.63%	97.65%
ShuffleNetV2	86.97%	83.81%	84.59%	86.34%
Res2Net50	96.95%	95.87%	96.39%	96.55%
DenseNet121	97.46%	97.63%	97.52%	97.65%
MobileNetV2	94.40%	89.90%	91.96%	92.62%
RegNet	94.86%	92.37%	93.39%	93.88%
GhostNet	95.49%	93.19%	94.22%	94.35%
EfficientNet	97.27%	96.32%	96.78%	96.70%
Ours	98.49%	97.89%	98.18%	98.43%

As shown in Table 3, the proposed model (Ours) still exhibits excellent comprehensive performance in the apple disease recognition task, with an accuracy of 98.43% and an F1-score of 98.18%, both outperforming all other comparison models. In particular, the recall rate reaches 97.89%, indicating that the model has high

recognition completeness for disease samples. Classic models such as GoogLeNet and DenseNet121 also achieve an accuracy of 97.65%, showing strong feature transfer ability. In contrast, lightweight models including ShuffleNetV2 and MobileNetV3 perform relatively poorly, with accuracies of 86.34% and 92.62%

respectively, demonstrating their limited generalization ability in cross-domain recognition tasks. The experimental results illustrate that the improved method proposed in this paper not only performs prominently in tomato leaf disease recognition, but also achieves superior recognition performance and favorable generalization ability on the apple leaf disease dataset, further verifying the effectiveness and robustness of the model in agricultural disease recognition tasks.

To further evaluate the generalization and transfer ability of the proposed model in disease recognition tasks of different crops, a generalization experiment was conducted on a public grape leaf disease dataset in this section. The dataset consists of four categories: grape black rot, Esca, leaf blight, and healthy leaves. Performance comparisons were carried out with mainstream models such as GoogLeNet, Res2Net50, DenseNet121, and EfficientNet. The experimental results were obtained by retraining the proposed model on the public grape leaf disease dataset.

As can be seen from Table 4, the proposed model (Ours) achieves the best performance in grape disease

recognition, with an accuracy of 99.51%. Precision, recall, and F1-score reach 99.40%, 99.59%, and 99.49%. Respectively, and all indicators are significantly superior to those of other comparison models. Traditional models such as GoogLeNet also perform well, with an accuracy of 98.89%, indicating a certain feature transfer ability. However, the lightweight model ShuffleNetV2 only achieves an accuracy of 79.12%, demonstrating its limited generalization ability in cross-domain recognition.

The proposed model performs particularly well in recall, showing a low miss rate for disease samples and high recognition completeness. The experimental results illustrate that the recognition model based on the improved EfficientNet not only exhibits superior performance in tomato disease classification, but also maintains excellent recognition accuracy and stable generalization ability on the grape leaf disease dataset with large structural differences, further verifying the effectiveness and robustness of the model in feature extraction and cross-domain recognition.

Table 4: Performance comparison of different models on the grape leaf disease dataset.

Model	Precision	Recall	F1-score	Accuracy
GoogLeNet	99.16%	98.87%	99.01%	98.89%
ShuffleNetV2	85.44%	79.18%	80.00%	79.12%
Res2Net50	96.74%	95.83%	96.26%	96.07%
DenseNet121	97.11%	97.18%	97.07%	96.93%
MobileNetV2	94.70%	94.21%	94.25%	93.49%
RegNet	96.58%	96.06%	96.27%	95.82%
GhostNet	95.51%	93.49%	94.40%	94.23%
EfficientNet	97.86%	97.12%	97.48%	97.42%
Ours	99.40%	99.59%	99.49%	99.51%

5 Conclusion

In this paper, an improved model based on EfficientNet is proposed for tomato leaf disease recognition. By introducing the RefConv module, ECA attention mechanism, and Lion optimizer, the model performance in fine-grained disease classification is significantly enhanced. Experimental results demonstrate that on the tomato disease dataset, the proposed method outperforms mainstream classification models in various metrics such as accuracy and F1-score. Meanwhile, the training process is stable without overfitting.

Ablation experiments verify the effectiveness and synergistic effect of each improved module. In particular, the combined use of ECA and Lion greatly improves the model's feature selection and convergence ability. Cross-species generalization experiments show that the model still maintains superior performance on apple and grape leaf disease datasets, indicating its favorable feature transfer ability and scene adaptability.

Future work will further explore lightweight designs to reduce model parameters and computational costs, and attempt to introduce temporal information or multimodal data to improve real-time recognition capability in practical agricultural environments.

Acknowledgement

This work was supported by the Key Research Project Plan of Henan Universities under Grant 26B520050, Science and Technology Research Project of Henan Province 262102210073.

Supported by the Key Discipline Construction Project of Intelligent Science and Technology, Henan Province.

References

- [1] S. P. Mohanty, D. P. Hughes, and M. Salathé (2016). Using deep learning for image-based plant disease detection, *Frontiers in plant science*, 7, 1419, <https://doi.org/10.3389/fpls.2016.01419>.
- [2] J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth (2022). Deep learning-based leaf disease detection in crops using images for agricultural applications, *Agronomy*, 12(10), 2395, <https://doi.org/10.3390/agronomy12102395>.
- [3] O. Attallah (2023). Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection, *Horticulturae*, 9(2), 149, <https://doi.org/10.3390/horticulturae9020149>.

- [4] M. Jelali (2024). Deep learning networks-based tomato disease and pest detection: a first review of research studies using real field datasets, *Frontiers in Plant Science*, 15, 1493322, <https://doi.org/10.3389/fpls.2024.1493322>.
- [5] L. K. Ndovie and E. Masabo (2024). Leveraging MobileNetV3 for in-field tomato disease detection in Malawi via CNN, *SAIEE Africa Research Journal*, 115(3), 74–85, <https://doi.org/10.23919/saiee.2024.10551304>.
- [6] M. Qabulio, M. S. Memon, S. Iqbal, P. Kumar, and A. Tsetse (2024). Effective tomato leaf disease identification model using mobilenetv3small, *International Journal of Information Systems and Computer Technologies*, 3(1), 57–72, <https://doi.org/10.58325/ijisct.003.01.0079>.
- [7] R. Rashid, W. Aslam, R. Aziz, and G. Aldehim (2025). A modified mobileNetv3 coupled with inverted residual and channel attention mechanisms for detection of tomato leaf diseases, *IEEE Access*, <https://doi.org/10.1109/ACCESS.2025.3550205>.
- [8] H. Sun, I. T. Nicholas, R. Fu, and D. K. Kang (2024). YOLO-FMDI: A lightweight YOLOv8 focusing on a multi-scale feature diffusion interaction neck for tomato pest and disease detection, *Electronics*, 13(15), 2974, <https://doi.org/10.3390/electronics13152974>.
- [9] W. Hou, C. Gong, W. Cao, et al. (2025). Tomato leaf disease detection method based on super-resolution enhancement and improved YOLOv8, *Transactions of the Chinese Society of Agricultural Engineering*, 41, 211–220, <https://doi.org/10.11975/j.issn.1002-6819.202503209>.
- [10] U. Islam, G. Ferraioli, and V. Pascasio (2025). Tomato leaf detection, segmentation, and extraction in real-time environment for accurate disease detection, *AgriEngineering*, 7(4), 120, <https://doi.org/10.3390/agriengineering7040120>.
- [11] Y. Cao (2025). Accurate Tomato Leaf Disease Identification Method based on Improved Swin Transformer, *Journal of Imaging Science & Technology*, 69(5), 1, <https://doi.org/10.2352/J.ImagingSci.Technol.2025.69.5.050503>.
- [12] Y. Sun, L. Ning, B. Zhao, and J. Yan (2024). Tomato leaf disease classification by combining EfficientNetv2 and a Swin transformer, *Applied Sciences*, 14(7), 7472, <https://doi.org/10.3390/app14177472>.
- [13] S. Yu, L. Xie, and L. Dai (2025). ST-CFI: Swin Transformer with convolutional feature interactions for identifying plant diseases, *Scientific Reports*, 15(1), 25000, <https://doi.org/10.1038/s41598-025-08673-0>.
- [14] O. Nabil and C. Leghris (2026). EcoTomHybridNet: Policy-Guided Adaptive CNN–Transformer Inference for Resource-Aware Edge-Based Tomato Leaf Disease Classification, *Future Internet*, 18(5), 271, <https://doi.org/10.3390/fi18050271>.
- [15] M. E. Qureshi, M. H. Ashraf, M. W. Arshad, A. Khan, and H. Ali (2025). Hierarchical Feature Fusion With Inception V3 for Multiclass Plant Disease Classification, *Informatica*, 49(27), 145–160, <https://doi.org/10.31449/inf.v49i27.8208>.
- [16] M. Tan and Q. Le (2019). Efficientnet: Rethinking model scaling for convolutional neural networks, *International Conference on Machine Learning*, 97, 6105–6114.
- [17] Z. Cai, X. Ding, Q. Shen, and X. Cao (2025). RefConv: Reparameterized Refocusing Convolution for Powerful ConvNets, *IEEE Transactions on Neural Networks and Learning Systems*, 36, 11617–11631, <https://doi.org/10.1109/TNNLS.2025.3552654>.
- [18] Q. Wang, B. Wu, P. Zhu, et al. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11534–11542, <https://doi.org/10.1109/CVPR42600.2020.01155>.
- [19] X. Chen, C. Liang, D. Huang, et al. (2023). Symbolic discovery of optimization algorithms, *Advances in Neural Information Processing Systems*, 36, 49205–49233.
- [20] C. Szegedy, W. Liu, Y. Jia, et al. (2015). Going deeper with convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.
- [21] H. Ran, S. Wen, S. Wang, et al. (2020). Memristor-based edge computing of ShuffleNetV2 for image classification, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(8), 1701–1710, <https://doi.org/10.1109/TCAD.2020.3022970>.
- [22] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang, and P. Torr (2019). Res2net: A new multi-scale backbone architecture, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2), 652–662.
- [23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017). Densely connected convolutional networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.
- [24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen (2018). Mobilenetv2: Inverted residuals and linear bottlenecks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.
- [25] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár (2020). Designing network design spaces, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10428–10436.
- [26] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu (2020). Ghostnet: More features from cheap operations, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1580–1589.

