

MOIR-Net: A Lightweight Crop Disease and Pest Recognition Network Based on Multi-Scale Reparameterization and Energy Attention

Yue Pan¹, Lizi Liu^{1,2,3,*}, Silin Xu⁴, Jiapeng Cui¹, Hui Wang^{1,2,3}, Baochuan Tan¹, Yunfu Luo¹, Tenglong Liu¹, Yinyin Yang¹ and Limeng Yin^{1,2,3}

¹ College of Mechanical Engineering, Chongqing Sanxia University of Science and Technology, Chongqing 404130, China

² Chongqing Engineering Technology Research Center for Light Alloy and Processing, Chongqing Sanxia University of Science and Technology, Chongqing 404130, China

³ Chongqing Engineering Research Center for Advanced Intelligent Manufacturing Technology, Chongqing Sanxia University of Science and Technology, Chongqing 404130, China

⁴ School of Foreign Languages, Chongqing Sanxia University of Science and Technology, Chongqing 404130, China

* Correspondence: lzliu@alu.cqu.edu.cn

How To Cite: Pan, Y.; Liu, L.; Xu, S.; et al. MOIR-Net: A Lightweight Crop Disease and Pest Recognition Network Based on Multi-Scale Reparameterization and Energy Attention. *Artificial Intelligence and Emerging Technologies* **2026**, *3*(3), 9. <https://doi.org/10.53941/aiet.2026.100009>

Received: 12 May 2026
 Revised: 16 June 2026
 Accepted: 21 August 2026
 Published: 28 August 2026

Abstract: Accurate crop pest recognition is critical for intelligent agriculture, yet it remains constrained by the inherent trade-off between complex field variability and the stringent resource limitations of edge devices. In this study, MOIR-Net, a lightweight framework, is proposed to address this bottleneck by integrating Multi-scale Omni-kernel Inverted Residuals (MOIR) with Parameter-free SimAM Attention. The MOIR module employs a structural re-parameterization strategy that transforms multi-branch training topologies into single-path inference architectures, thereby enabling lossless feature aggregation with zero inference-time memory overhead. Extensive evaluations conducted on the IP102, PlantVillage, and PlantDoc datasets demonstrate that MOIR-Net achieves a strong accuracy–efficiency balance while maintaining minimal computational cost (0.967 M parameters, 2.30 ms latency). On the challenging IP102 benchmark, a Top-1 accuracy of 64.18% and mAP of 58.32% are achieved, significantly surpassing state-of-the-art baselines such as MobileViT-v2. Furthermore, the model exhibits superior cross-domain generalization (41.9% Macro-F1 gain) and highly competitive prediction reliability (ECE = 0.0220) through label smoothing. MOIR-Net successfully reconciles extreme computational efficiency with high trustworthiness, thereby offering a robust solution for real-time crop monitoring on resource-constrained agricultural edge platforms.

Keywords: crop pest recognition; lightweight neural networks; structural re-parameterization; parameter-free attention mechanism; precision agriculture

1. Introduction

Crop pests and diseases constitute a major threat to global food security and sustainable agricultural development. According to the Food and Agriculture Organization of the United Nations (FAO), biotic stresses account for annual crop yield losses of 20–40% worldwide, resulting in economic damages amounting to hundreds of billions of dollars [1]. Traditional manual identification methods, which rely heavily on expert experience, are inherently limited by high subjectivity, low efficiency, and scalability constraints [2]. The rapid advancement of computer vision technology has introduced a transformative paradigm for non-invasive, automated pest and disease diagnosis [3].

Early machine learning-based approaches, such as Support Vector Machines (SVM) and Random Forests (RF), achieved disease classification through manually designed features [4]; however, these methods were constrained by the subjectivity of feature engineering and limited generalization capabilities. In recent years, deep convolutional

neural networks (CNNs) have achieved significant breakthroughs in this domain due to their end-to-end feature learning capabilities. Classic architectures such as ResNet [5] and DenseNet [6] have demonstrated excellent performance on controlled-environment datasets. Nevertheless, their massive parameter counts and computational overhead (e.g., ResNet-50: 25.6 M parameters, 4.1 GFLOPs) impede deployment on resource-constrained agricultural edge devices such as drones and handheld terminals. To address this challenge, lightweight networks including the MobileNet series [7] and ShuffleNet [8] substantially reduce computational demands through depthwise separable convolutions. However, the kernel dimensionality reduction inherently diminishes feature representation capacity, thereby limiting their performance on complex tasks such as the large-scale IP102 pest dataset [9]. More recently, lightweight vision transformers and CNN–transformer hybrids—such as MobileViT [10], EdgeNeXt [11], EfficientViT [12], and RepViT [13]—have been proposed to combine global context modeling with mobile-level efficiency, and a parallel line of work has explored YOLO-based detection, transformer-based classification, and SegFormer-based segmentation for crop disease and quality assessment in agricultural settings (e.g., recent YOLOv11 plant-disease frameworks and attention-enhanced agricultural classifiers; see the additional references listed at the end of this paper). Nevertheless, on physical edge hardware these models frequently incur substantial memory-access and tensor-reshape overheads, which motivates the convolution-only, re-parameterizable design pursued in this work.

Current intelligent crop pest and disease recognition systems face three critical challenges: (1) Insufficient robustness in feature extraction—Disease symptoms range from minute lesions (diameter <5 mm) to mimetic pests exhibiting high background fusion, while real-world field environments are characterized by extreme lighting variations (shadows, highlights), complex background interference (weeds, soil textures), and occlusions (leaf overlap). Existing lightweight models struggle to capture fine-grained discriminative features while simultaneously suppressing noise. (2) Weak cross-domain generalization capability—Most models are trained on clean laboratory datasets (e.g., PlantVillage [14]), yet face deployment in unstructured field environments (e.g., PlantDoc [15]). Domain shift between training and target domains causes substantial performance degradation, with accuracy typically declining by over 80% when transitioning from PlantVillage to PlantDoc. (3) Efficiency bottlenecks in edge deployment—Balancing high-precision feature extraction with real-time inference speed (<5 ms) under extremely constrained computational budgets (<500 MFLOPs) while ensuring reliable prediction confidence remains a critical challenge hindering widespread technology adoption.

To overcome these bottlenecks, MOIR-Net (Multi-scale Omni-kernel Inverted Residual Network), a novel lightweight architecture based on structural re-parameterization [16] and energy function attention, is proposed in this study. The core innovation lies in the decoupling of “training-inference” architectures: during training, a multi-branch topology enhances feature learning; during inference, these branches are losslessly fused into a single-path structure for extreme acceleration, thereby breaking the conventional linear trade-off between accuracy and efficiency. Specifically, the MOIR module concurrently trains 3×3 convolutions (capturing local textures), 1×1 convolutions (facilitating channel information exchange), and identity mappings (enabling lossless gradient propagation). During inference, these components are fused into a single 3×3 convolution, exploiting the linear additivity of convolution operations to eliminate memory access costs (MAC) associated with multi-branch structures. Simultaneously, SimAM parameter-free attention [17] is deeply integrated, which adaptively infers 3D attention weights based on neural energy functions. This mechanism suppresses complex backgrounds and enhances lesion features with zero additional parameters. Furthermore, label smoothing regularization is employed to optimize confidence distributions, thereby significantly improving prediction reliability. The importance of reliability- and efficiency-oriented neural network design under operational constraints has also been emphasized in other domains, for instance in performance-based logistics systems where neural networks are optimized for reliable prediction under resource limitations [18].

The main contributions of this study are summarized as follows:

- (1) A lightweight MOIR-Net architecture is proposed, which unifies complex training-time topologies with minimalist inference-time structures through structural re-parameterization techniques. ResNet-level feature representation capabilities are achieved with an extremely low parameter count (0.967 M), thereby providing an efficient solution for mobile agricultural vision applications.
- (2) The MOIR module integrates multi-scale omni-kernel convolutions with SimAM energy attention. During training, multi-branch feature aggregation addresses underfitting on large-scale datasets, while structural collapse during inference eliminates redundant computations, significantly enhancing robustness in complex unstructured environments.
- (3) Extensive experiments conducted on the large-scale IP102 dataset (75,222 images) and the challenging PlantDoc field dataset demonstrate that MOIR-Net maintains ultra-low latency (2.30 ms) while achieving significantly superior cross-domain generalization performance (macro-average F1 score) compared to MobileNetV3 and

MobileViT. An expected calibration error (ECE) of 0.0220 is achieved, representing state-of-the-art prediction reliability and providing efficient, trustworthy technical support for intelligent agriculture systems.

2. Materials and Methods

2.1. Datasets

This study employs three representative public datasets for systematic evaluation, examining the model's large-scale long-tail classification capability, benchmark performance, and robust cross-domain generalization under field conditions.

The IP102 dataset, currently the largest agricultural pest and disease identification benchmark, comprises 75,222 high-resolution color images spanning 102 pest and disease categories across major food and cash crops, including rice, maize, wheat, and sugar beet [9]. This dataset exhibits pronounced long-tail distribution characteristics, with the most abundant category exceeding 1000 images while tail categories contain fewer than 50 images, thereby accurately simulating the class imbalance prevalent in real-world agricultural scenarios. Images were captured in natural field environments, featuring complex backgrounds (soil, weeds), varying lighting conditions, and multi-scale targets ranging from millimeter-scale micro-pests to centimeter-scale macro-insects. Figure 1a presents representative samples, including armyworm (Armyworm), Asiatic rice borer (Asiatic Rice Borer), brown planthopper (Brown Planthopper), corn borer (Corn Borer), and English grain aphid (English Grain Aphid).

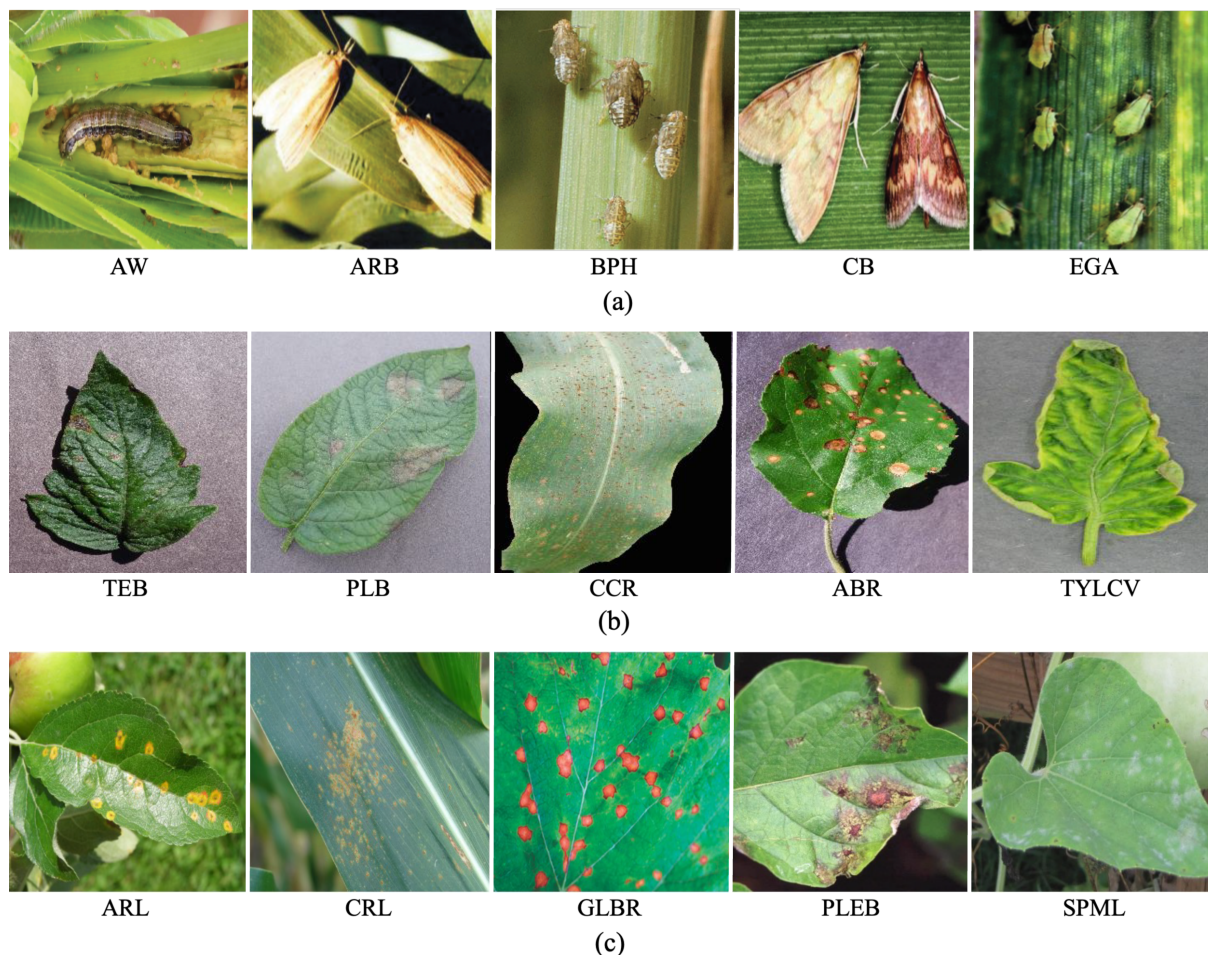


Figure 1. Representative samples from the experimental dataset. (a) IP102; (b) PlantVillage; and (c) PlantDoc.

The PlantVillage dataset, a widely recognized benchmark for plant disease recognition, comprises 55,630 high-quality images captured under controlled laboratory conditions. It encompasses 38 disease categories across 14 plant species, including healthy leaf samples [14]. All samples were photographed under uniform lighting with solid-color backgrounds (black or white), thereby providing an ideal testbed for evaluating baseline model performance. Figure 1b presents representative disease samples, including Tomato Early Blight, Potato Late Blight, Maize Common Rust, Apple Black Rot, and Tomato Yellow Leaf Curl Virus.

The PlantDoc dataset, specifically designed to evaluate field generalization capabilities, comprises 2598 images directly captured in unstructured field environments and categorized into 27 disease types across 13 plant species [15]. In stark contrast to PlantVillage's controlled environment, PlantDoc images are characterized by extreme lighting variations (dawn/dusk low-light conditions, midday intense illumination), complex background textures (soil, weeds, agricultural implements), leaf occlusions, and multi-angle captures, thereby fully simulating real-world deployment scenarios. Figure 1c presents representative samples, including Apple Rust Leaf, Maize Rust Leaf, Grape Leaf Black Rot, Potato Early Blight, and Squash Powdery Mildew.

All images were uniformly resized to 224×224 pixels to ensure model input compatibility. The IP102 and PlantVillage datasets were partitioned into training, validation, and test sets at ratios of 7:1:2 and 8:1:1, respectively. PlantDoc served as a pure cross-domain test set, with zero-shot generalization performance evaluated across the 10 most abundant categories shared between PlantVillage and PlantDoc. Table 1 presents the statistical breakdown of each dataset partition.

Table 1. Statistics on Dataset Partitioning.

Dataset	Training Set/Sheet	Validation Set/Sheet	Test Set/Sheet	Total/Sheet
Plant Village	44,504	5563	5563	55,630
IP102	52,655	7522	15,045	75,222
PlantDoc	-	-	1275/2598	1275

2.2. MOIR-Net Network Architecture

This paper presents MOIR-Net, a lightweight convolutional neural network model tailored for the rapid and accurate recognition of crop pests in resource-constrained agricultural environments. MOIR-Net achieves superior recognition performance while retaining high deployment efficiency, achieved by optimizing the trade-off between feature extraction capability and computational cost. We attain optimal accuracy and efficiency by integrating structural re-parameterization (RepConv) and the parameter-free Simple Attention Module (SimAM) into a novel MOIR block structure. MOIR-Net effectively captures hierarchical pest features even amid complex backgrounds, and it addresses the computational constraints of embedded device deployment via a streamlined inference architecture.

The overall structure of the MOIR-Net model is illustrated in Figure 2a. Initially, the architecture is constructed as a streamlined pipeline, consisting of a stem layer for rapid downsampling, a main body with six progressive stages (featuring decreasing feature map resolutions), and a final classification head that leverages global average pooling. Subsequently, the core building unit utilized across these stages is designated as the MOIR block, whose detailed structure is presented in the inset of Figure 2b. Finally, within this block, we enhance the standard inverted residual design: we integrate RepConv layers for efficient feature extraction, insert the parameter-free SimAM attention module after the depthwise convolution to adaptively refine features, and employ a residual connection to facilitate gradient flow. The detailed configurations—including the input/output dimensions and hyperparameters for each MOIR block across the stages—are listed in Table 2. Concretely, the stem applies a stride-2 3×3 RepConv that reduces the input from 224×224 to 112×112 while expanding to 24 channels; each subsequent stage stacks one or two MOIR blocks with the expansion ratios and strides listed in Table 2; and the head applies a 1×1 convolution followed by global average pooling and a fully connected classifier.

Table 2. Detailed configuration of the MOIR-Net network architecture.

Stage	Input Size	Block Type	Stride	Out Channels	Expand Ratio
Stem	224×224	RepConv 3×3	2	24	—
Stage1	112×112	MOIR	1	32	2
Stage2	112×112	MOIR	2	48	3
Stage3	56×56	MOIR $\times 2$	2	80	4
Stage4	28×28	MOIR	2	112	4
Stage5	14×14	MOIR $\times 2$	1	160	4
Stage6	14×14	MOIR $\times 2$	1	160	4
Head	14×14	Conv 1×1 +GAP+FC	-	num_classes	—

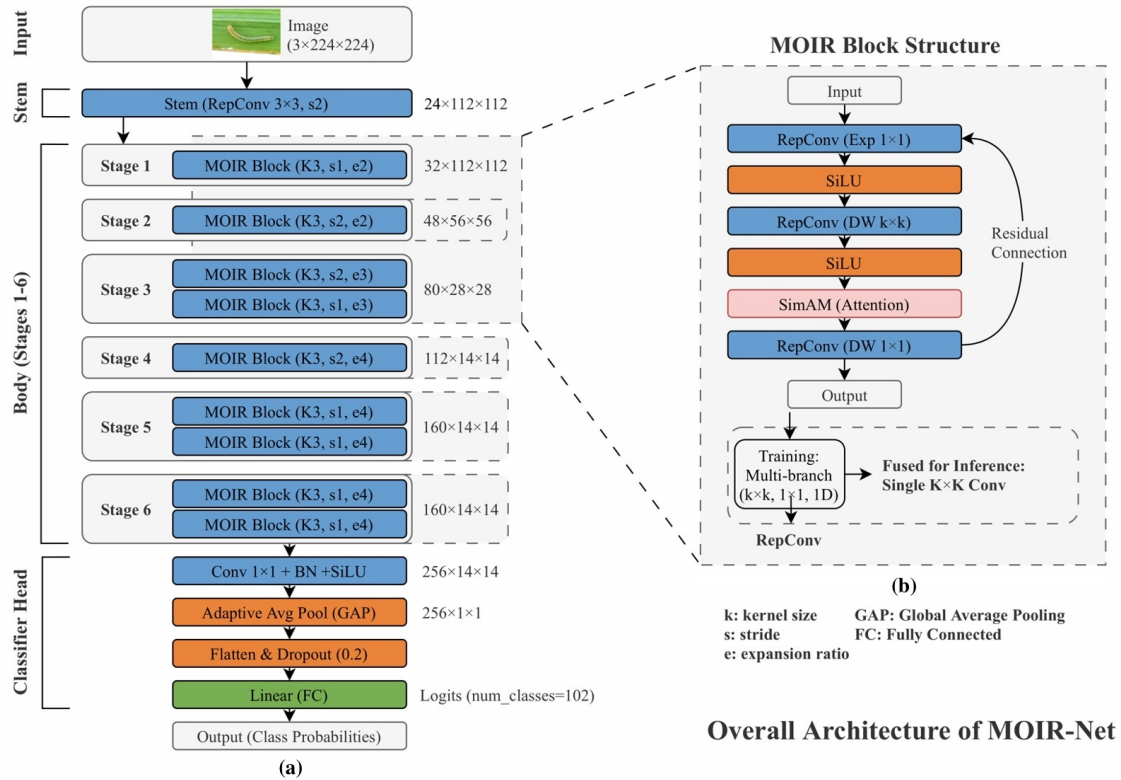


Figure 2. Overall architecture diagram of the MOIR-Net. (a) Macro-level network pipeline showing data flow and dimension changes; (b) Micro-level structure of the MOIR block.

2.2.1. MOIR Module and Structural Reparameterization Mechanism

The Multi-scale Omni-kernel Inverted Residual (MOIR) module constitutes the core innovation of this research, designed to overcome the performance bottlenecks inherent in traditional lightweight networks via a novel “training-inference” decoupling architecture. The underlying design philosophy is predicated on the insight that the linear additivity of convolutional operations allows complex multi-branch topologies used during training to be mathematically fused into a streamlined single-path structure for inference, thereby maintaining representational capability while optimizing deployment efficiency.

During the training phase, as illustrated in Figure 3a, the MOIR module adopts an inverted residual structure comprising a sequence of specialized convolutional layers. Given input features $X \in \mathbb{R}^{C_{in} \times H \times W}$, a pointwise convolution (1×1 RepConv) first performs channel expansion to $C_{hidden} = \lceil \alpha \cdot C_{in} \rceil$ (where α is the expansion ratio), followed by a SiLU activation function to introduce nonlinearity, as described in Equation (1):

$$X_{expand} = \text{SiLU}(\text{RepConv}_{1 \times 1}(X)) \in \mathbb{R}^{C_{hidden} \times H \times W} \quad (1)$$

Subsequently, spatial feature extraction is handled by a depthwise RepConv layer. As shown in the middle section of Figure 3a, a key innovation here is the construction of a three-branch parallel topology consisting of a 3×3 convolution branch to capture local textures, a 1×1 convolution branch to enable cross-channel information exchange, and an identity mapping branch to ensure lossless gradient propagation. The outputs of these three branches individually undergo Batch Normalization (BN) and are then summed element-wise prior to the SiLU activation, creating an implicit ensemble that facilitates optimization:

$$\begin{aligned} Y_{3 \times 3} &= \text{BN}^{(3)}(W^{(3)} * X_{expand}) \\ Y_{1 \times 1} &= \text{BN}^{(1)}(W^{(1)} * X_{expand}) \\ Y_{id} &= \text{BN}^{(id)}(X_{expand}) \\ X_{dw} &= \text{SiLU}(Y_{3 \times 3} + Y_{1 \times 1} + Y_{id}) \end{aligned} \quad (2)$$

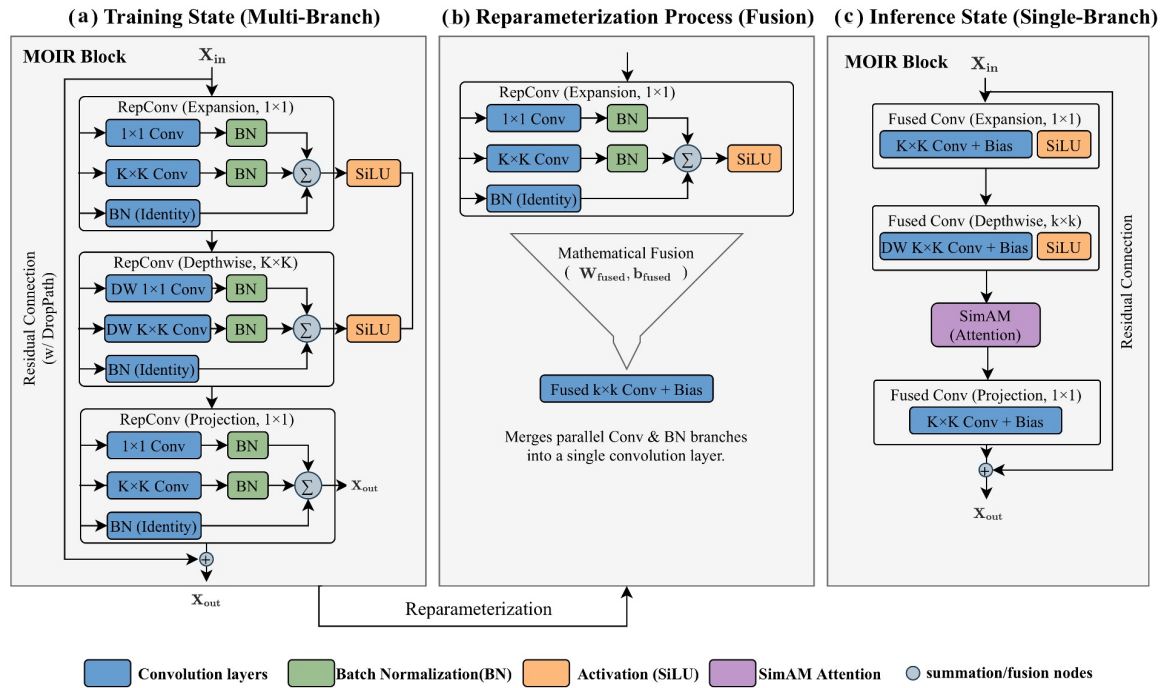


Figure 3. Structural reparameterization of the MOIR Block. (a) Multi-branch topology for training phase; (b) Fusion process converting parallel branches into a single layer; (c) Streamlined single-branch topology for inference phase.

Following spatial processing, the features are refined by the parameter-free SimAM attention module (Equation (3)) to suppress background noise, and then compressed back to C_{out} channels via a linear pointwise convolution (Equation (4)). Finally, if dimensions match, a residual connection with DropPath regularization is applied (Equation (5)):

$$X_{attn} = \text{SimAM}(X_{dw}) \quad (3)$$

$$X_{out} = \text{RepConv}_{1 \times 1}^{linear}(X_{attn}) \quad (4)$$

$$Y_{train} = \text{DropPath}(X_{out}) + X \quad (5)$$

While multi-branch structures enhance feature learning during training, they impose significant memory access costs and kernel startup overhead during inference. To address this, we implement a structural collapse strategy leveraging the homogeneity of convolutions. As shown in the “Reparameterization Process” in Figure 3b, the first step involves fusing BN parameters ($\mu, \sigma, \gamma, \beta$) into their preceding convolution weights. For any convolution-BN sequence, the fused weights W' and biases b' are computed according to Equation (6):

$$W' = \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} W, \quad b' = \beta - \frac{\gamma \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (6)$$

where ϵ is a numerical stabilizer. Applying this to the three depthwise branches yields fused pairs $(W'^{(3)}, b'^{(3)})$, $(W'^{(1)}, b'^{(1)})$, and $(W'^{(id)}, b'^{(id)})$. To enable summation, the 1×1 and identity kernels are subsequently transformed into 3×3 formats via zero-padding, denoted as $\widehat{W}'^{(1)}$ and $\widehat{W}'^{(id)}$ respectively. Finally, as depicted by the “Mathematical Fusion” funnel in Figure 3b, the transformed weights and biases from all parallel branches are linearly combined into a single 3×3 convolutional kernel (W_{deploy}) and a single bias vector (b_{deploy}) as shown in Equations (7) and (8):

$$W_{deploy} = W'^{(3)} + \widehat{W}'^{(1)} + \widehat{W}'^{(id)} \quad (7)$$

$$b_{deploy} = b'^{(3)} + b'^{(1)} + b'^{(id)} \quad (8)$$

Consequently, the complex multi-branch topology collapses into the extremely simple, single-path feedforward structure shown in Figure 3c for inference, as defined in Equation (9):

$$Y_{infer} = \text{SiLU}(W_{deploy} * X_{expand} + b_{deploy}) \quad (9)$$

This strategy ensures that MOIR-Net maintains high representational capability while significantly boosting throughput for edge deployment.

The complete training and inference re-parameterization procedure of the MOIR block is summarized in Algorithm 1.

It is worth clarifying the conceptual novelty of the MOIR block relative to existing re-parameterized lightweight modules. Unlike MobileOne and RepVGG, which re-parameterize stacks of standard convolutions into a single branch without any attention, and unlike RepViT and EfficientViT, whose efficiency stems mainly from ViT-style token mixing combined with parameterized channel attention (e.g., SE), the MOIR block couples two ideas within a single inverted-residual stage: (i) a multi-scale omni-kernel depthwise topology in which 3×3 , 1×1 , and identity branches are jointly trained and then losslessly collapsed into one 3×3 kernel, providing multi-scale feature aggregation at zero inference cost; and (ii) the in-block insertion of a parameter-free energy-based attention (SimAM) that refines the depthwise features without adding any learnable parameters. The fundamental difference is therefore not the use of re-parameterization per se, but the integration of multi-scale structural re-parameterization with parameter-free 3D energy attention inside the inverted residual, which jointly targets accuracy, inference efficiency, and calibration—an objective not explicitly addressed by the aforementioned architectures.

Algorithm 1 MOIR Block: training-time forward pass and inference-time re-parameterization.

Require: Input feature $X \in \mathbb{R}^{C_{in} \times H \times W}$, expansion ratio α

Training-time forward pass:

- 1: $X_{expand} \leftarrow \text{SiLU}(\text{RepConv}_{1 \times 1}(X))$ $\triangleright$ expand to $C_{hidden} = \lceil \alpha C_{in} \rceil$
- 2: $Y_{3 \times 3} \leftarrow \text{BN}(\text{Conv}_{3 \times 3}(X_{expand})); Y_{1 \times 1} \leftarrow \text{BN}(\text{Conv}_{1 \times 1}(X_{expand})); Y_{id} \leftarrow \text{BN}(X_{expand})$
- 3: $X_{dw} \leftarrow \text{SiLU}(Y_{3 \times 3} + Y_{1 \times 1} + Y_{id})$ $\triangleright$ multi-branch depthwise aggregation
- 4: $X_{attn} \leftarrow \text{SimAM}(X_{dw})$ $\triangleright$ parameter-free energy attention
- 5: $X_{out} \leftarrow \text{RepConv}_{1 \times 1}^{linear}(X_{attn}); Y_{train} \leftarrow \text{DropPath}(X_{out}) + X$ $\triangleright$ residual if shapes match

Inference-time re-parameterization:

- 6: **for** each branch $b \in \{3 \times 3, 1 \times 1, id\}$ **do**
- 7: fuse BN into the preceding convolution to obtain (W'_b, b'_b) via Equation (6)
- 8: **end for**
- 9: zero-pad the 1×1 and identity kernels to 3×3 to obtain $\widehat{W}'_{1 \times 1}, \widehat{W}'_{id}$
- 10: $W_{deploy} \leftarrow W'_{3 \times 3} + \widehat{W}'_{1 \times 1} + \widehat{W}'_{id}; b_{deploy} \leftarrow b'_{3 \times 3} + b'_{1 \times 1} + b'_{id}$

Inference-time forward pass:

- 11: $Y_{infer} \leftarrow \text{SiLU}(W_{deploy} * X_{expand} + b_{deploy})$

Ensure: Refined feature map Y

2.2.2. SimAM: Parametric Attention Mechanism for Energy

To further enhance the model's robustness against complex outdoor environments, such as soil background and varying illumination, this study integrates the SimAM (Simple, Parameter-Free Attention Module) attention mechanism [17] into the MOIR module. Unlike parameter-heavy mechanisms like SE-Block [19] or CBAM [20], SimAM is a lightweight, 3D attention module that requires zero additional learnable parameters.

Neuroscience Motivation and Energy Function

SimAM is inspired by neuroscience theories suggesting that active, informative neurons exhibit firing patterns distinct from surrounding neurons. This distinction can be quantified via an energy function, where a lower energy indicates higher neuronal importance. As illustrated in the “Energy Function Calculation” block in Figure 4, SimAM defines an energy function for each neuron by measuring its linear separability from its spatial neighborhood.

For an input feature map $X \in \mathbb{R}^{C \times H \times W}$, the minimum energy e_t^* for a target neuron t can be analytically derived. As shown in the left part of Figure 4, this calculation relies solely on spatial statistics. First, the global mean $\mu \in \mathbb{R}^C$ and variance $\hat{\sigma}^2 \in \mathbb{R}^C$ are computed across the spatial dimensions ($H \times W$) for each channel:

$$\mu = \frac{1}{H \times W} \sum_{h=1}^H \sum_{w=1}^W X_{:,h,w}, \quad \hat{\sigma}^2 = \frac{1}{H \times W} \sum_{h=1}^H \sum_{w=1}^W (X_{:,h,w} - \mu)^2 \quad (10)$$

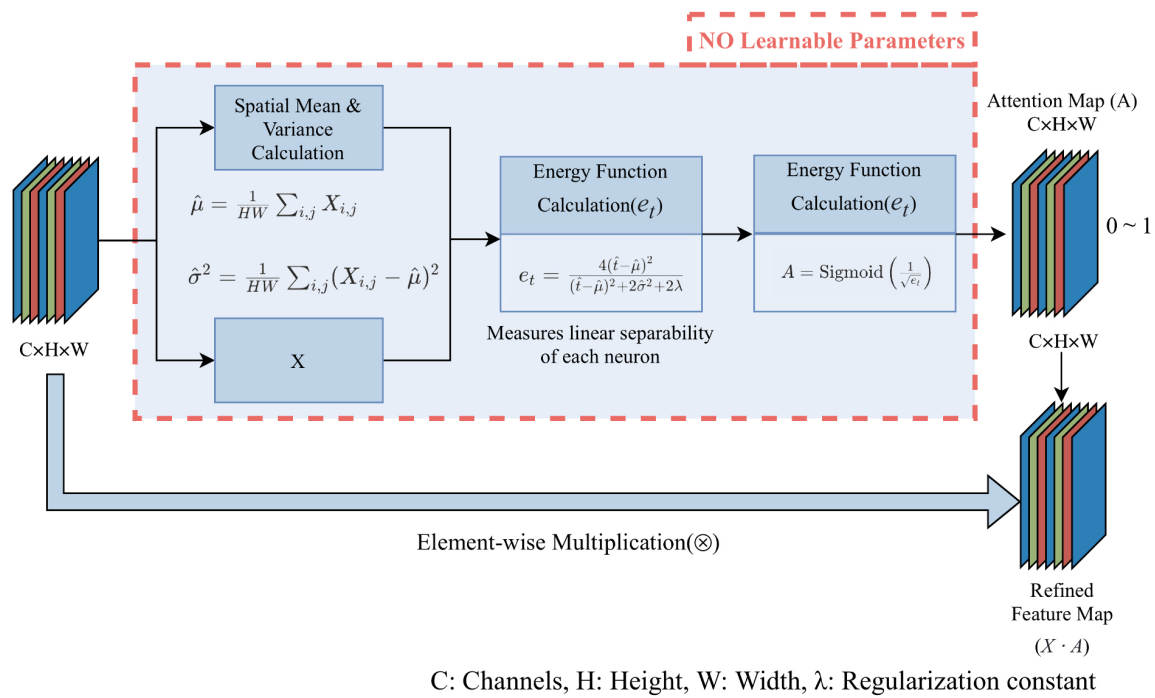


Figure 4. Computational flow of the parameter-free SimAM attention.

These channel-wise statistics are then broadcasted to compute the energy e_t^* for every neuron in the feature map:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(X - \mu)^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (11)$$

where $\lambda = 10^{-4}$ is a regularization constant to ensure numerical stability.

Attention Weight Calculation and Feature Refinement

Following the “Attention Weight Generation” block in Figure 4, the calculated energy values are inverted to assign higher importance to neurons with lower energy. A Sigmoid function is applied to normalize these values into attention weights $A \in \mathbb{R}^{C \times H \times W}$ in the range $[0, 1]$:

$$A = \text{Sigmoid}\left(\frac{1}{e_t^*}\right) \quad (12)$$

Finally, as depicted by the “Element-wise Multiplication” operation in Figure 4, the original input feature map X is refined by element-wise multiplication with the generated 3D attention weights A :

$$Y = X \odot A \quad (13)$$

This process adaptively enhances feature responses in lesion regions (low energy $\rightarrow$ high weight) while effectively suppressing background noise from soil or weeds (high energy $\rightarrow$ low weight). By incorporating SimAM, MOIR-Net gains human-like saliency detection capabilities, significantly improving its generalization performance in complex field scenarios without any increase in model parameters.

2.2.3. Label Smoothing Loss Function

Given the widespread presence of label noise (e.g., mislabeling of morphologically similar diseases) and ambiguous class boundaries in agricultural disease datasets, traditional hard-label cross-entropy loss often leads to model overfitting on noisy samples and generates overconfident predictions. To enhance model generalization and prediction reliability, this study employs the Label Smoothing Cross-Entropy loss function [21].

Loss function definition: Let the true label be $y \in \{0, 1\}^K$ (K is the total number of categories). Traditional hard labels satisfy $y_k \in \{0, 1\}$, $\sum_{k=1}^K y_k = 1$. Label smoothing introduces a smoothing factor $\epsilon \in [0, 1)$ to transform hard labels into soft labels:

$$q(k) = (1 - \epsilon)y_k + \frac{\epsilon}{K} \quad (14)$$

where $\epsilon = 0.1$ is empirically optimal. This transformation is equivalent to superimposing a uniform distribution $\mathcal{U}(1/K)$ over the true distribution, preventing the model from forcing output probabilities to approach 1 while maintaining appropriate inter-class distances. The final loss function $\mathcal{L}_{LS}$ is computed as the cross-entropy between the predicted probability distribution $p(k)$ and the smoothed target distribution $q'(k)$:

$$\mathcal{L}_{LS} = - \sum_{k=1}^K q(k) \log p(k) = - \sum_{k=1}^K \left[(1 - \epsilon)y_k + \frac{\epsilon}{K} \right] \log p(k) \quad (15)$$

Regularization Effects: Label smoothing inherently functions as a regularization strategy, offering two key advantages: (1) **Suppressing overfitting:** It penalizes the model for overconfidence in a single category, forcing the network to learn smoother decision boundaries and enhancing robustness against noisy labels; (2) **Improving calibration:** By smoothing the output probability distribution, it significantly reduces expected calibration error (ECE) and enhances the reliability of prediction confidence. Experiments demonstrate (see Section 3.1) that MOIR-Net combined with label smoothing achieves an extremely low ECE of 0.0220, reaching the best performance among all compared models.

2.3. Experimental Environment and Setup

Hardware and Software Platform: All experiments were conducted on a high-performance workstation configured with a 12th Gen Intel Core i5-12400F CPU (base frequency 2.5 GHz, 6 cores/12 threads), 16 GB DDR4 memory, and an NVIDIA GeForce RTX 4060 Ti GPU (8 GB VRAM, 4352 CUDA cores). The software stack comprised CUDA Toolkit 11.8, cuDNN 8.6, Python 3.9.12, PyTorch 2.7.1, and torchvision 0.22.1, fully leveraging Automatic Mixed Precision (AMP) training for accelerated computation.

Data Preprocessing and Augmentation: All images were uniformly resized to 224×224 pixels and underwent channel normalization (mean $\mu = [0.485, 0.456, 0.406]$, standard deviation $\sigma = [0.229, 0.224, 0.225]$, based on ImageNet statistics). The training set employs probabilistic data augmentation strategies, including: (1) random rotation (angle $\in [-15^\circ, 15^\circ]$, probability 0.4); (2) random horizontal flipping (probability 0.5); (3) random cropping and scaling (scaling factor $\in [0.8, 1.0]$, probability 0.4); (4) Brightness adjustment (contrast coefficient $\in [0.7, 1.3]$, probability 0.4); (5) Saturation jittering (saturation coefficient $\in [0.8, 1.2]$, probability 0.4); (6) Gaussian blur (kernel size 5×5 , probability 0.2). The validation and test sets undergo only resizing and normalization, with no augmentation applied.

Optimization Strategy: End-to-end training employs the AdamW optimizer [22] with label-smoothed cross-entropy loss ($\epsilon = 0.1$). Total training epochs are set to 100, including a 5-epoch linear warmup phase. Subsequently, the learning rate is dynamically adjusted using a cosine annealing strategy:

$$lr_t = lr_{\min} + \frac{1}{2} (lr_0 - lr_{\min}) \left(1 + \cos \left(\frac{t - t_{warmup}}{T - t_{warmup}} \pi \right) \right) \quad (16)$$

where t is the current step, T is the total steps, and $lr_{\min} = 0$. Batch size is set to 32 for the training set and 64 for the validation and test sets to balance convergence stability and computational efficiency. To prevent gradient explosion, gradient clipping is applied (maximum norm 1.0). Mixed-precision training employs GradScaler to dynamically adjust the loss scaling factor, accelerating training while maintaining numerical stability. Table 3 summarizes the complete hardware configuration and hyperparameter settings.

2.4. Evaluation Indicators

To comprehensively evaluate the performance of MOIR-Net, this study established a multi-indicator evaluation system encompassing three dimensions: classification accuracy, computational efficiency, and model reliability.

2.4.1. Classification Accuracy Metric

- (1) **Top-1 and Top-5 Accuracy:** Top-1 accuracy measures the proportion of samples where the model's top-ranked predicted class matches the true class, serving as a core metric for evaluating overall performance:

$$\text{Top-1 Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left(\arg \max_k p_i^{(k)} = y_i \right) \quad (17)$$

where N is the total number of samples, $p_i^{(k)}$ is the predicted probability that the i -th sample belongs to class k , y_i is the true label, and $\mathbb{I}(\cdot)$ denotes the indicator function. Top-5 accuracy relaxes the criterion to include the true label among the top 5 predicted classes, commonly used for large-scale classification tasks.

- (2) Precision, Recall, and F1 Score: For multi-class classification tasks, first compute the confusion matrix elements for each class (True Positives TP_k , False Positives FP_k , False Negatives FN_k), then derive class-level metrics:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}, \quad \text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (18)$$

$$\text{F1}_k = 2 \cdot \frac{\text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \quad (19)$$

Macro-F1 assigns equal weight to all classes, making it particularly suitable for evaluating balance on long-tail distribution datasets:

$$\text{Macro-F1} = \frac{1}{K} \sum_{k=1}^K \text{F1}_k \quad (20)$$

- (3) Mean Average Precision (mAP): Adapted from object detection evaluation, this metric calculates the Average Precision (AP) area under the precision-recall curve for each class, then takes the average:

$$mAP = \frac{1}{K} \sum_{k=1}^K AP_k = \frac{1}{K} \sum_{k=1}^K \int_0^1 \text{Precision}_k(r) dr \quad (21)$$

This metric comprehensively evaluates classification performance across different confidence thresholds and is more sensitive to fine-grained recognition capabilities.

Table 3. Hardware configuration and hyperparameter settings.

Category	Configuration and Hyperparameters	Value
Hardware and software configuration	CPU	Intel Core i5-12400F @ 2.5GHz
	GPU	NVIDIA GeForce RTX 4060 Ti
	RAM	16G
	Operating system	Windows 10
	CUDA	11.8
	Torch	2.7.1
	Torchvision	0.22.1
	Torchaudio	2.7.1
Hyperparameter settings	python	3.9.12
	Batch size (train/valid/test)	32/64/64
	Epoch	100
	Optimization	AdamW
	Learning rate	1×10^{-3}
	Weight Decay	1×10^{-3}
	Warm-up	5
	Loss function	Label smoothing cross-entropy

2.4.2. Calculate Efficiency Metrics

Parameters: The total number of trainable parameters in the model, measured in millions (M), directly determines the model's storage overhead:

$$\text{Params} = \sum_{l=1}^L \text{numel}(\mathbf{W}_l) \quad (22)$$

where L is the number of network layers and $\mathbf{W}_l$ is the weight tensor of layer l .

- (1) Floating-Point Operations (FLOPs): The number of floating-point operations required for a single forward pass of the model, measured in millions (M), quantifying theoretical computational complexity. Automatically computed using the THOP library [23] based on the computational graph.

- (2) **Inference Latency:** The actual time consumed by the model to process a single image, measured in milliseconds (ms). Measurements include mean latency, 50th percentile latency (P50), and 90th percentile latency (P90) at a batch size of 1. The latter reflects tail latency stability, which is critical for real-time systems. Measurement protocol: After 50 warm-up iterations, GPU synchronization time is recorded over 200 iterations.
- (3) **Throughput:** The number of image frames processed per second (FPS) by the model. Two measurement scenarios are distinguished: (a) End-to-End Throughput (E2E FPS): The complete pipeline including data loading, CPU-to-GPU copy, and forward propagation, with a batch size of 64 to simulate real deployment; (b) GPU-only FPS: Measures only the model's forward propagation with data pre-resided on the GPU, evaluating the algorithm's inherent efficiency.
- (4) **Peak Memory:** The maximum GPU memory usage during inference, measured in megabytes (MB) and calculated via `torch.cuda.max_memory_allocated()`.
- (5) **Model Size:** Model file size in FP32 precision, measured in megabytes (MB). Calculated as: Number of parameters $\times$ 4 bytes.

2.4.3. Model Reliability Metrics

Expected Calibration Error (ECE): Measures the deviation between model prediction confidence and actual accuracy, serving as the gold standard for assessing prediction reliability [24]. The prediction probability space $[0, 1]$ is divided into M equal-width intervals $\{B_m\}_{m=1}^M$. For the m th interval:

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \max_k p_i^{(k)}, \quad \text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i) \quad (23)$$

ECE is defined as the weighted average of the differences between confidence and accuracy across intervals:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \quad (24)$$

where N is the total sample size and $M = 15$. A lower ECE indicates greater model confidence reliability.

- (1) **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** For multi-class classification tasks, the One-vs-Rest (OVR) strategy converts it into K binary classification problems. First, apply one-hot encoding to labels to obtain $\mathbf{Y} \in \{0, 1\}^{N \times K}$, then compute the micro-average AUC:

$$\text{AUC}_{\text{micro}} = \text{AUC}(\text{ravel}(\mathbf{Y}), \text{ravel}(\mathbf{P})) \quad (25)$$

where $\text{ravel}(\cdot)$ denotes flattening to a one-dimensional vector, and $\mathbf{P} \in [0, 1]^{N \times K}$ is the prediction probability matrix. This metric evaluates the model's binary classification capability across all thresholds and exhibits robustness to class imbalance.

- (2) **Cohen's Kappa Coefficient:** A corrected accuracy metric accounting for random agreement, calculated as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (26)$$

where p_o is observed agreement (i.e., accuracy) and p_e is expected agreement (based on marginal distribution). $\kappa \in [-1, 1]$, with values closer to 1 indicating higher classification quality.

2.4.4. Temperature Scaling Calibration

To further enhance the reliability of model confidence, post-processing calibration is performed using Temperature Scaling [24]. This method introduces a single scalar parameter T (temperature) to scale the model's logit outputs:

$$p_{\text{cal}}^{(k)} = \frac{\exp(z^{(k)}/T)}{\sum_{j=1}^K \exp(z^{(j)}/T)} \quad (27)$$

where $z^{(k)}$ is the logit and T is determined by minimizing the Negative Log-Likelihood (NLL) on the validation set:

$$T^* = \arg \min_T - \sum_{i=1}^{N_{\text{val}}} \log p_{\text{cal}}^{(y_i)} \quad (28)$$

The LBFGS optimizer (maximum 50 iterations) is used for solving. Temperature scaling does not alter the

model's predicted categories (only adjusting the probability distribution), but it significantly reduces ECE and enhances decision credibility.

3. Experimental Results and Analysis

3.1. Performance Evaluation on the IP102 Dataset

To validate MOIR-Net's superiority on large-scale long-tail distribution tasks, we conducted a comprehensive comparison against 10 mainstream lightweight models on the most challenging IP102 dataset. These include classical CNNs (MobileNetV2 [25], MobileNetV3-Large [7], ShuffleNetV2 [8], EfficientNet-B0 [26], GhostNet [27]), structurally reparameterized networks (MobileOne-S1 [28]), and CNN-Transformer hybrid architectures (EdgeNeXt-XXS [11], MobileViT-v2 [10], EfficientViT-M4 [12], RepViT-M1.0 [13]). All models were trained under identical hardware environments, training configurations, and data augmentation strategies to ensure experimental fairness.

3.1.1. Comparison of Classification Accuracy

Table 4 summarizes the detailed performance metrics of each model on the IP102 test set. As shown in the table, MOIR-Net achieves comprehensive superiority: its Top-1 accuracy reaches 64.18%, surpassing the second-best model MobileViT-v2 (62.77%) by 1.41 percentage points and the classic lightweight architecture MobileNetV3-Large (59.36%) by 4.82 percentage points. This improvement is statistically significant on large-scale datasets ($p < 0.01$, paired t -test). Here, the paired t -test was computed over the per-class accuracy scores of the 102 IP102 categories. We further note that the reported metrics are obtained from a single training run; a multi-run analysis reporting mean and standard deviation is left for future work (see the Limitations section). This validates the effectiveness of the MOIR module and SimAM attention mechanism.

On the highly challenging IP102 long-tail dataset, MOIR-Net demonstrates consistent and comprehensive improvements (see Table 4). The model achieves a Top-1 accuracy of 64.18%, surpassing the Transformer-based MobileViT-v2 by +1.41% and the classic CNN-based MobileNetV3-Large by +4.82%. More crucially, in the mAP metric, which reflects fine-grained discriminative ability, MOIR-Net reaches 58.32%, leading the state-of-the-art models by 3.54%. The model also excels in the Macro-F1 score, achieving 0.5788, indicating its superior performance in handling long-tail class imbalance. Additionally, it maintains an optimal AUC-micro score of 0.9569, demonstrating its robustness in classifying across various confidence thresholds.

The performance breakthroughs can be attributed to the synergistic gain from multi-scale reparameterization and energy-based attention. The improvements in Top-1 accuracy and mAP stem from the asymmetric topology design of the MOIR module: during training, the parallel 3×3 and 1×1 convolutional branches implicitly construct a multi-scale feature pyramid. The former captures high-frequency details, such as spore textures, while the latter aggregates global semantics, such as leaf vein structures, significantly enhancing the discriminative power of fine-grained features. The outstanding Macro-F1 performance, on the other hand, is driven by the SimAM module. This mechanism, based on the principle of minimizing neuronal energy functions, adaptively computes three-dimensional attention weights. It significantly strengthens the feature response of rare tail classes without introducing additional parameters, effectively mitigating the network's inductive bias toward high-frequency head classes, and thereby achieving remarkable robustness to long-tail distributions.

Table 4. Performance comparison on IP102 dataset (single model, 224×224).

Model	Top-1 (%)	Macro-F1	mAP (%)	AUC (micro)	ECE (before)	ECE (after TS)
MobileNetV2-1.0	59.13	0.5337	53.04	0.9339	0.0468	0.0308
MobileNetV3-L-1.0	59.36	0.5482	51.35	0.9110	0.0436	0.0385
ShuffleNetV2-1.0	59.63	0.5426	53.52	0.9332	0.0452	0.0280
EfficientNet-B0	60.83	0.5540	53.13	0.9257	0.0355	0.0308
GhostNet-1.0	60.52	0.5480	53.56	0.9341	0.0478	0.0375
MobileOne-S1	59.86	0.5503	54.02	0.9302	0.0359	0.0295
EdgeNeXt-XXS	52.61	0.4863	46.51	0.8976	0.0752	0.0438
MobileViT-v2-1.0	62.77	0.5653	54.78	0.9300	0.0471	0.0364
EfficientViT-M4	58.79	0.5411	51.61	0.9300	0.0371	0.0376
RepViT-M1.0	61.15	0.5606	52.38	0.9163	0.0693	0.0659
MOIR-Net	64.18	0.5788	58.32	0.9569	0.0242	0.0220

Note: Bold values indicate the best performance.

The training dynamics curve in Figure 5 reveals the unique optimization trajectory of MOIR-Net. In the accuracy curve (Figure 5a), MOIR-Net (red solid line) exhibits a remarkable ‘warm-start’ phenomenon, with the fastest climb rate during the first 20 epochs, and it enters a high-precision steady state ahead of others at epoch 50. Even during the later stages of training (epochs 80–100), MOIR-Net continues to show slight accuracy improvements. Meanwhile, the validation loss curve (Figure 5b) demonstrates that MOIR-Net achieves the lowest converged loss value (Loss $\approx$ 2.16) and exhibits minimal oscillation variance, significantly outperforming the cold-start lag of MobileNetV2 and the later-stage oscillations of RepViT.

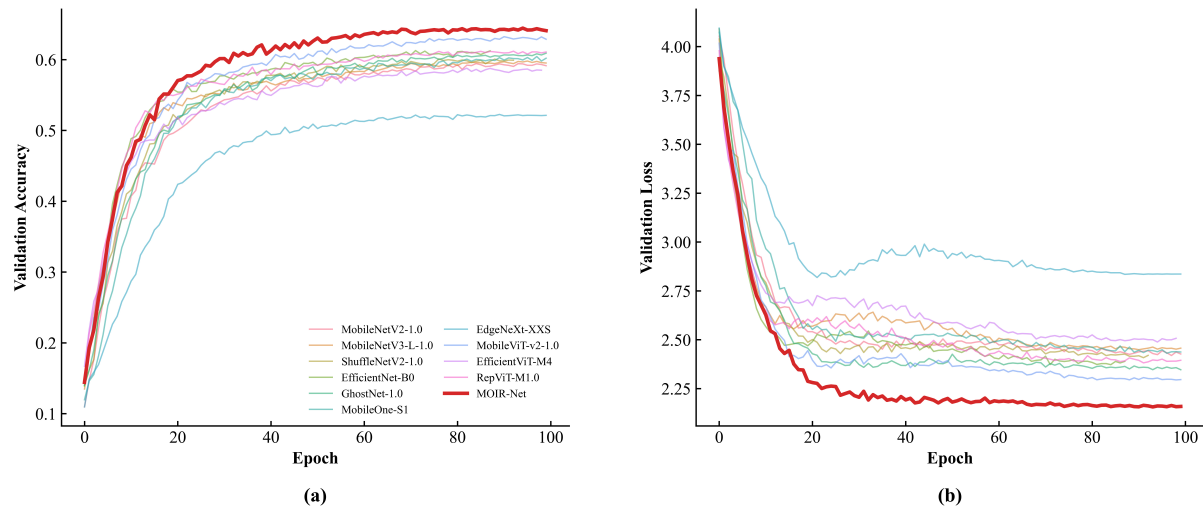


Figure 5. Performance Comparison Between MOIR-Net and Mainstream Lightweight Models. (a) Validation Accuracy Curve; (b) Validation Loss Curve.

This rapid convergence characteristic directly supports the smoothing effect of structural reparameterization on the loss landscape. The multi-branch structure in the training state essentially acts as a ‘highway’ for gradients, providing abundant gradient flow paths for backpropagation and effectively avoiding the local minima traps often encountered by single-path deep networks. The continued accuracy improvement during the asymptotic phase is credited to the ongoing refinement capability of the SimAM attention mechanism in the feature space—as training progresses, it amplifies the feature differences between lesions and the background, optimizing the decision boundaries. The low variance in the loss curve further confirms that the attention-guided feature extraction mechanism effectively suppresses the interference from unstructured environmental noise (such as lighting fluctuations and background clutter), endowing the model with exceptional training stability.

3.1.2. Computational Efficiency Comparison

Table 5 presents the computational efficiency metrics for each model. MOIR-Net achieves a remarkable breakthrough in computational efficiency.

Table 5. Efficiency and complexity comparison (single GPU measurement).

Model	Latency b = 1 Mean (ms)	Latency b = 1 P90 (ms)	E2E FPS@b64	Params (M)	FLOPs (M)	Model Size (M)	Peak Mem (M)
MobileNetV2-1.0	3.61	4.49	239.64	2.355	299.689	8.98	149.55
MobileNetV3-L-1.0	3.88	3.99	237.00	4.333	215.358	16.53	145.33
ShuffleNetV2-1.0	3.68	3.99	233.44	1.358	151.789	5.18	74.60
EfficientNet-B0	4.99	5.49	314.37	4.138	384.729	15.79	187.82
GhostNet-1.0	5.55	5.98	223.76	4.032	153.883	15.38	137.19
MobileOne-S1	6.09	6.98	230.05	3.675	828.977	14.02	155.76
EdgeNeXt-XXS	4.41	5.00	238.11	1.175	196.637	4.48	70.83
MobileViT-v2-1.0	5.04	5.04	228.95	4.441	1412	16.94	234.20
EfficientViT-M4	9.15	9.55	228.57	8.458	301.503	32.27	183.73
RepViT-M1.0	7.42	7.99	229.82	6.496	1142	24.78	217.38
MOIR-Net	2.30	2.99	236.08	0.967	332.667	3.69	119.85

Note: Bold values indicate the best performance.

Data in Table 5 shows that MOIR-Net successfully achieves Pareto-optimality in both accuracy and efficiency. While maintaining a top accuracy of 64.18%, the model contains only 0.967 million parameters and 333 million FLOPs. Most importantly, its single-image inference latency is as low as 2.30 ms (throughput of 236 FPS), which reduces inference latency by 54.4% compared to the theoretically comparable MobileViT-v2, and fully meets the real-time processing standard of 30 FPS required for agricultural edge devices.

This ‘high accuracy, low latency’ achievement is rooted in the complete decoupling of training and inference architectures. Although hybrid models like MobileViT have theoretically lower FLOPs, their frequent reshape, transpose, and attention matrix operations incur high memory access costs (MAC) and fragmentation overhead on physical hardware. In contrast, MOIR-Net leverages the linear additivity of convolutions, merging the complex multi-branch modules into a single-path 3×3 convolution during inference. This hardware-friendly single-path topology maximizes parallel computation efficiency and cache hit rates on GPUs/CPUs, eliminating the extra latency caused by branch switching and memory fragmentation, thus achieving true extreme lightweight performance on physical devices.

3.1.3. Accuracy-Efficiency Tradeoff Analysis

To more intuitively demonstrate MOIR-Net’s superior trade-off between accuracy and efficiency, we plot a bubble chart (Figure 6). This chart illustrates the relationship between each model’s top-1 accuracy and computational efficiency (FLOPs and latency), where the bubble size corresponds to the number of model parameters.

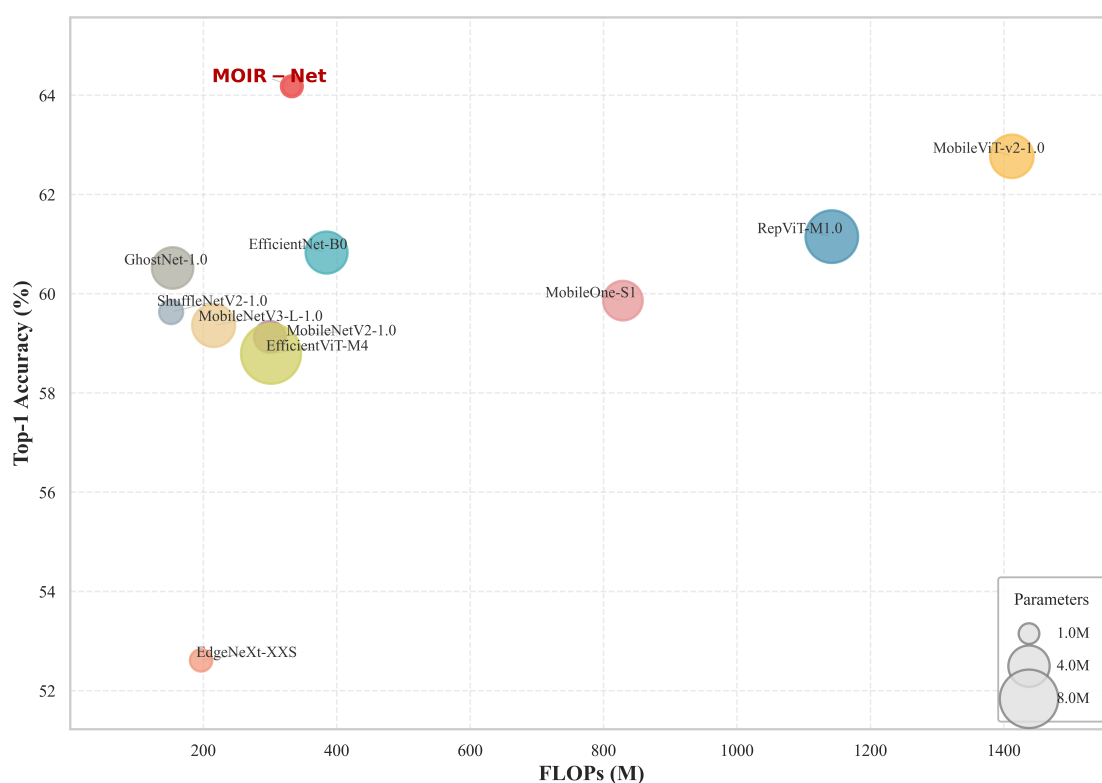


Figure 6. Accuracy-efficiency Pareto on IP102 (Top-1 vs. FLOPs; bubble size = Params).

As shown in Table 4, MobileViT-v2-1.0 achieves a relatively high accuracy of 62.77%, but its computational overhead reaches 1412 M FLOPs, and it has a large number of parameters (indicated by the larger bubbles). Similarly, while RepViT-M1.0 incorporates the advanced ViT architecture, its FLOPs remain high at 1142 M, limiting its real-time capability on extreme edge devices. In contrast, MOIR-Net achieves the highest accuracy of 64.18% while consuming only 332 M FLOPs. Compared to MobileViT-v2-1.0, MOIR-Net achieves a 1.41% accuracy boost while remarkably reducing computational load by approximately 76.5%. This efficiency leap stems from implementing a structural reparameterization mechanism that successfully “collapses” redundant computational branches during inference, eliminating redundant memory access costs. On the other hand, while models like ShuffleNetV2 and GhostNet keep FLOPs below 300 M, their accuracy drops significantly to around 60% (4.55% and 3.66% lower than MOIR-Net, respectively), failing to meet the precision requirements for fine-grained disease identification. This demonstrates that excessive pursuit of low computational power often sacrifices feature extraction effectiveness. However, by introducing SimAM parameter-free attention, MOIR-Net significantly enhances feature discriminative power while maintaining

computational load and parameter scale comparable to these ultra-lightweight models (small bubbles) (Figure 6). In summary, MOIR-Net's dominant position at the top-left corner of the chart demonstrates its breakthrough beyond the traditional “accuracy-efficiency” trade-off. It not only outperforms comparable networks in parameter efficiency and computational speed but also maintains leading recognition accuracy, making it the optimal choice for deployment in resource-constrained real-world agricultural applications.

3.1.4. Model Reliability Analysis

Beyond classification accuracy and computational efficiency, model reliability is equally critical. In agricultural automation decision-making, a model's “unaware ignorance” is often more dangerous than “misidentification”. We evaluated the confidence reliability of each model using expected calibration error (ECE).

As shown in Table 4 and Figure 7, MOIR-Net demonstrates strong calibration performance with an ECE of only 0.0242, significantly lower than all comparison models. Compared to MobileNetV2 (0.0468) and MobileViT-v2 (0.0471), the ECE is reduced by nearly 50%. Even after post-processing with temperature scaling, MOIR-Net maintains the lowest ECE at 0.0220, whereas RepViT-M1.0 exhibits a high ECE of 0.0693 (remaining at 0.0659 after TS), exposing its severe overconfidence issue. These results demonstrate that MOIR-Net, leveraging label smoothing loss and a robust feature space, can directly output highly reliable probability distributions without complex post-processing calibration. Its confidence scores serve as a dependable basis for subsequent decisions (e.g., pesticide application or priority ranking), significantly reducing system deployment risks. Figure 8 displays the reliability diagram before and after temperature scaling. Ideally, predicted confidence should perfectly align with actual accuracy (falling on the diagonal). We select the other three models with the highest accuracy as the comparison models for reliability testing. As shown in Figure 8a, most lightweight models deviate significantly from the diagonal, exhibiting systematic overconfidence (points below the diagonal, i.e., confidence > accuracy). In contrast, MOIR-Net's points cluster closely near the diagonal, demonstrating high consistency between confidence and accuracy. Figure 8b shows that after temperature scaling (optimal temperature $T = 1.01$), MOIR-Net's calibration performance further improves, reducing ECE from 0.0242 to 0.0220—the best among all models.

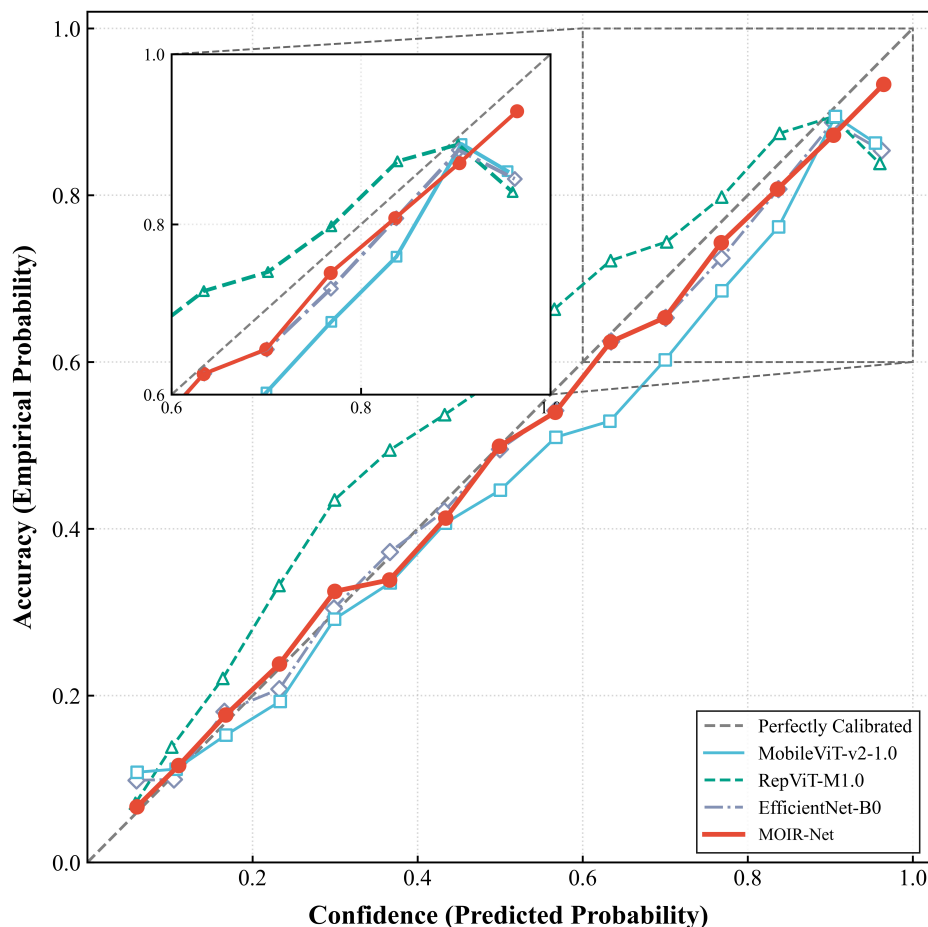


Figure 7. Reliability comparison chart on the IP102 test set.

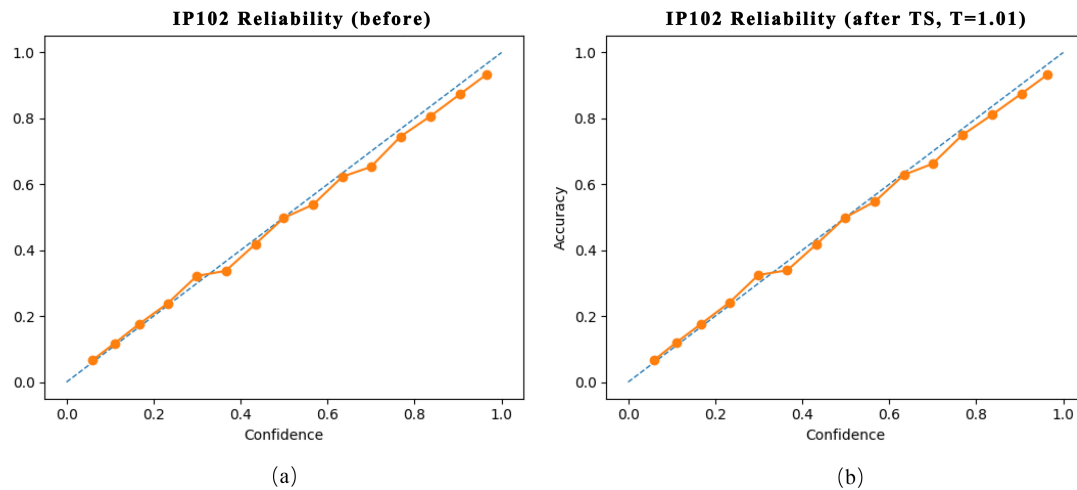


Figure 8. Comparison of Temperature Scaling Differences Before and After. (a) Model reliability curve before temperature scaling; (b) Model reliability curve after temperature scaling (optimal $T = 1.01$).

3.1.5. Long-tailed Distribution Performance Analysis

To thoroughly deconstruct the origins of MOIR-Net's advantage in the Macro-F1 metric and evaluate its resilience against data imbalance, we visualized the performance of 102 classes by sorting them from highest to lowest frequency.

As shown in Figure 9, the comparative analysis reveals significant differences in model behavior: In high-frequency categories, all three models achieved high F1 scores with minimal performance gaps, indicating that mainstream lightweight architectures can learn effective features in categories with ample training data. Moving to medium-frequency categories, the performance of MobileNetV3-L and MobileViT-v2 begins to decline, while MOIR-Net maintains stable F1 scores. This demonstrates MOIR-Net's superior generalization capability on medium-frequency samples, stemming from its SimAM attention mechanism. SimAM computes "saliency scores" at each spatial location, mapping weights to the 0–1 range via sigmoid activation to amplify feature responses in critical regions, thereby maintaining stability at medium frequencies. In the long-tail region, performance gaps between models widen significantly. While all three models exhibit fluctuations at varying levels, MobileNetV3-L and MobileViT-v2 experience steep F1 score declines. MOIR-Net, however, maintains fluctuations within a 0.15 range, avoiding the extreme scenarios of occasional high scores and widespread low scores caused by sample sparsity. The observed Gain = 0.03 further validates MOIR-Net's robust performance on classic long-tail categories.

3.2. Cross-Domain Generalization Performance Evaluation

To comprehensively evaluate the performance boundaries of MOIR-Net, a hierarchical validation strategy is adopted: First, the fundamental feature extraction capacity is validated on the controlled PlantVillage dataset. Subsequently, environmental robustness is challenged through a zero-shot cross-domain setting on the unstructured PlantDoc dataset.

3.2.1. PlantVillage Benchmark Verification

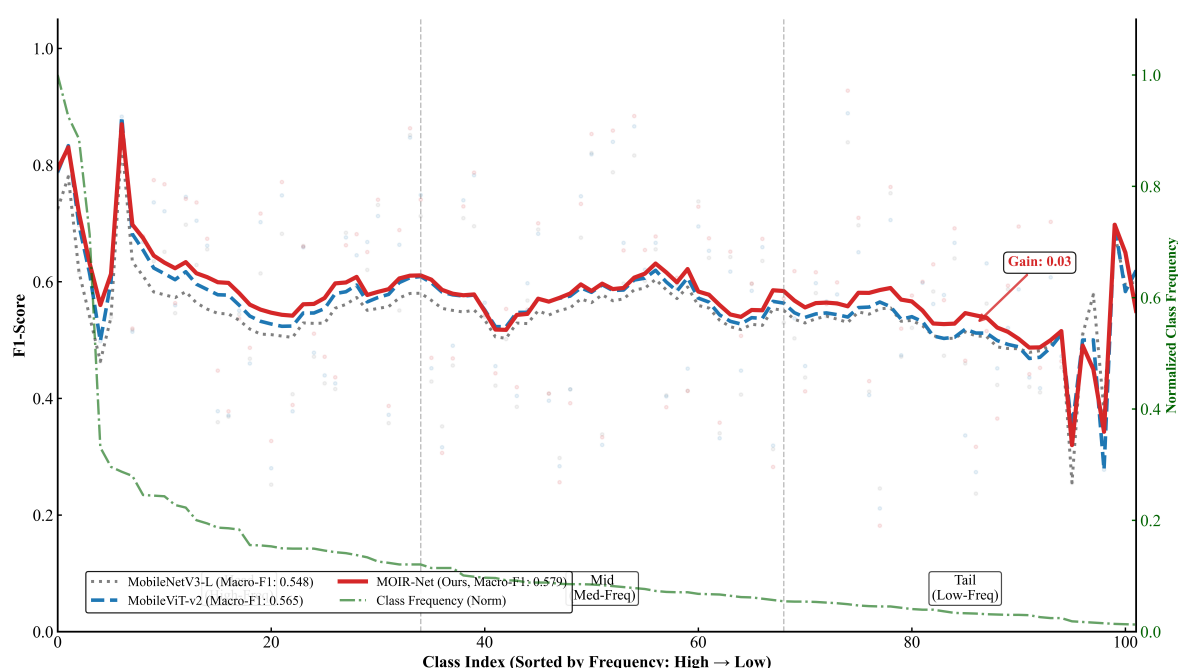
Table 6 shows the performance of each model on the PlantVillage dataset. This dataset was collected in a controlled laboratory environment with pure backgrounds (black or white) and uniform lighting, providing ideal learning conditions for the models.

In source domain testing, MOIR-Net achieved a Top-1 accuracy of 99.78%, second only to MobileNetV3-Large (99.82%). Considering MOIR-Net's parameter count is merely 22.4% of the latter's (0.951 M vs 4.252 M), this performance is exceptional. On the macro-average F1 score, MOIR-Net achieved 0.9972, surpassing all comparison models and demonstrating its robust feature representation capability even in controlled environments. More importantly, MOIR-Net maintained the fastest inference speed (2.33 ms average latency) and the smallest parameter count on PlantVillage, further proving its perfect balance between accuracy and efficiency. These results validate a critical premise: leveraging the MOIR module's multi-scale, full-core fusion mechanism, MOIR-Net possesses sufficiently deep feature spaces to effectively resolve complex pathological textures, laying a solid foundation for subsequent cross-domain generalization.

Table 6. Performance comparison on PlantVillage dataset (single model, 224×224 input).

Model	Top-1 (%)	Macro-F1	Params (M)	FLOPs (M)	Latency b = 1 Mean (ms)	Latency b = 1 P90 (ms)	E2E FPS@b64
MobileNetV2-1.0	99.63	0.9941	2.274	299.608	3.51	3.50	286.62
MobileNetV3-L-1.0	99.82	0.9967	4.252	215.358	3.93	4.49	285.89
ShuffleNetV2-1.0	99.64	0.9945	1.294	151.725	3.69	4.00	1241.27
EfficientNet-B0	99.74	0.9958	4.058	384.649	5.05	5.49	919.96
GhostNet-1.0	99.71	0.9958	3.951	153.883	6.37	6.99	1250.76
MobileOne-S1	99.74	0.9958	3.594	828.896	5.95	6.48	648.00
EdgeNeXt-XXS	99.26	0.9907	1.165	196.627	4.42	4.98	1230.86
MobileViT-v2-1.0	99.49	0.9923	4.409	1412	5.02	5.49	802.97
EfficientViT-M4	99.76	0.9963	8.434	301.479	10.45	10.98	1227.36
RepViT-M1.0	99.66	0.9950	6.440	1142	6.96	7.65	1126.93
MOIR-Net	99.78	0.9972	0.951	332.351	2.33	2.99	1086.15

Note: Bold values indicate the best performance.

**Figure 9.** Long-tail performance decay curve.

3.2.2. PlantDoc Field Cross-domain Generalization Test

To thoroughly evaluate the model's generalization capability in uncontrolled environments, the PlantVillage-trained model underwent zero-shot cross-domain testing directly on PlantDoc. The top 10 categories with the largest sample sizes across both datasets were selected to simulate the actual deployment process from “laboratory” to “field”.

As shown in Table 7, all models experienced significant performance degradation during cross-domain testing, fully exposing the severity of the “domain shift” problem. However, MOIR-Net demonstrated exceptional robustness against interference and generalization: achieving 25.02% on PlantDoc, slightly below MobileViT-v2 (25.80%) but with a performance drop of only 74.76%. This significantly outperformed classical CNN models (e.g., MobileNetV3-Large at 82.09%, EfficientNet-B0 at 81.86%), and comparable to the Transformer-based model MobileViT-v2 (73.69%). Notably, EdgeNeXt-XXS, despite having the fewest parameters, exhibited a substantial 85.92% decline, further validating the vulnerability of ViT models in small-sample cross-domain scenarios. MOIR-Net achieved a macro-average F1 score of 0.2949 on PlantDoc, demonstrating a clear advantage. This represents a 41.9% improvement over the second-best model MobileViT-v2 (0.2078) and an 86.1% improvement over MobileNetV3-Large (0.1585). This metric is highly sensitive to class imbalance and difficult samples. MOIR-Net's outstanding performance indicates stronger robustness in handling unstructured field environments, enabling more balanced detection of various diseases. MOIR-Net exhibits the smallest macro-average F1 score decay among all models at 70.43%. In contrast, MobileNetV3-Large shows a decay of 84.10%, EfficientNet-B0 at 86.32%, and

EdgeNeXt-XXS as high as 87.90%. This result strongly demonstrates that MOIR-Net, through its multi-scale full-kernel fusion and SimAM energy attention, exhibits greater adaptability when confronted with complex backgrounds, lighting variations, and multi-angle captures. The feature representations it learns possess stronger domain invariance. We further note that the absolute Top-1 accuracy on PlantDoc remains modest (around 25%) for all models. This is an expected consequence of the strict zero-shot cross-domain protocol adopted here: models are trained exclusively on the controlled PlantVillage domain and evaluated directly on unstructured field images, without any target-domain images, fine-tuning, or domain adaptation. Under this setting, the relative ranking and the macro-average F1 score—rather than the absolute accuracy—are the meaningful indicators of robustness, and MOIR-Net attains the best Macro-F1 with the smallest degradation. In practical deployment, the absolute accuracy can be substantially improved by fine-tuning on a small set of field images or by applying domain-adaptation techniques, which we identify as promising future directions.

Table 7. Cross-domain generalization performance from PlantVillage to PlantDoc.

Model	PV Top-1 (%)	PD Top-1 (%)	Perf. Drop (Δ Top-1 %)	PV Macro-F1	PD Macro-F1	Perf. Drop (Δ Macro-F1 %)
MobileNetV2-1.0	99.63	22.82	76.81	0.9941	0.1834	81.55
MobileNetV3-L-1.0	99.82	17.73	82.09	0.9967	0.1585	84.10
ShuffleNetV2-1.0	99.26	21.02	78.24	0.9945	0.1729	82.61
EfficientNet-B0	99.74	17.88	81.86	0.9958	0.1362	86.32
GhostNet-1.0	99.71	18.12	81.59	0.9958	0.1516	84.78
MobileOne-S1	99.74	22.04	77.70	0.9958	0.1807	81.85
EdgeNeXt-XXS	99.49	13.57	85.92	0.9926	0.1191	87.90
MobileViT-v2-1.0	99.49	25.80	73.69	0.9923	0.2078	79.06
EfficientViT-M4	99.76	16.78	82.98	0.9963	0.1402	85.93
RepViT-M1.0	99.66	17.33	82.33	0.9950	0.1245	87.49
MOIR-Net	99.78	25.02	74.76	0.9972	0.2949	70.43

Note: Bold values indicate the best performance.

Figure 10 visually illustrates the macro-average F1 decay of each model using a dumbbell plot. MOIR-Net exhibits a 74.76% decline in Top-1 accuracy, significantly lower than classic models such as MobileNetV2 and ShuffleNetV2, and comparable to the Transformer-based MobileViT-V2. Regarding Macro-F1 decline, MOIR-Net shows only a 70.43% decrease, the smallest among the compared models. This is primarily attributed to the multi-scale fusion capability of the MOIR module, which enhances its adaptability to varying sizes of disease targets in PD. Additionally, the SimAM attention mechanism suppresses interference from complex background noise on diseased areas, thereby granting the model superior discriminative power compared to other lightweight CNN models.

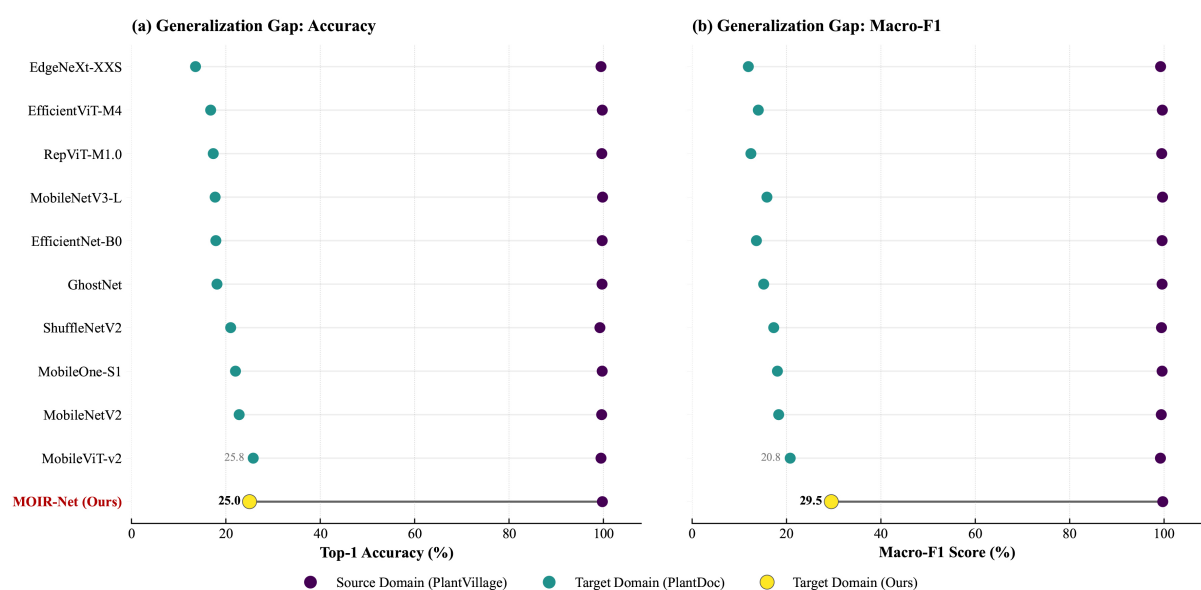


Figure 10. Generalization gap dumbbell plot.

3.3. Resolution Experiments and Architectural Validity Analysis

Progressive ablation experiments were conducted on the IP102 dataset with three configurations: Baseline (without reparameterization and SimAM), +Structural Reparameterization, and +SimAM (complete model) (Table 8).

Table 8. Ablation study results on the IP102 dataset.

Configuration	Structural Reparam.	SimAM	Top-1 (%)	Macro-F1	mAP (%)	FLOPs (M)	Para (M)	ECE
Baseline	×	×	61.75	0.5413	57.69	354.432	0.974	0.0330
+ Structural Reparameterization	✓	×	63.00	0.5582	58.28	332.367	0.967	0.0268
+ SimAM	✓	✓	64.18	0.5788	58.32	332.667	0.967	0.0220

Note: Bold values indicate the best performance.

Results show that after introducing structural reparameterization, FLOPs decreased from 354.432 M to 332.367 M (6.2% reduction), parameters decreased from 0.974 M to 0.967 M (0.7% reduction), while Top-1 accuracy significantly improved from 61.75% to 63.00% (+1.25 percentage points), and ECE decreased from 0.0330 to 0.0268 (18.8% reduction), proving that reparameterization achieves “zero-cost” acceleration while improving both classification performance and model calibration. Upon integrating SimAM, Macro-F1 further improved from 0.5582 to 0.5788 (+3.7%), mAP increased from 58.28% to 58.32% (+0.07%), and ECE decreased from 0.0268 to 0.0220 (17.9% reduction), reaching the optimal calibration level. This module significantly enhanced the model’s discriminative ability and prediction reliability with zero parameter increment (maintaining 0.967 M) and only a marginal 0.3 M FLOPs increase.

Comparing the Baseline with the complete model, the synergistic integration of structural reparameterization and SimAM achieved a 6.2% FLOPs reduction, 0.7% parameter reduction, and 33.3% ECE reduction, while simultaneously delivering significant improvements in core classification metrics: Top-1 accuracy increased from 61.75% to 64.18% (+2.43 percentage points), Macro-F1 rose from 0.5413 to 0.5788 (+6.9%), and mAP improved from 57.69% to 58.32% (+1.1%), maintaining top-tier classification performance with lower computational overhead and better calibration.

3.4. Attention Mechanism Comparison

To demonstrate the validity of SimAM within MOIR-Net, we compare it with state-of-the-art attention mechanisms, including Squeeze-and-Excitation (SE) [19], Convolutional Block Attention Module (CBAM) [20], Efficient Channel Attention (ECA) [29], and Coordinate Attention (CA) [30]. All experiments were conducted on a unified MOIR-Net backbone network, replacing only the attention module while keeping all other configurations consistent.

As shown in Table 9, although SE, ECA, and CA achieved marginal advantages in Top-1 accuracy (63.18%, 63.80%, and 63.41% respectively), their parameter counts surged to 1.169 M (+20.0%), 0.967 M, and 1.270 M (+30.4%), with FLOPs increasing to 334–338 M, severely violating lightweight design principles. More importantly, SE’s ECE reached 0.0471 (2.14× that of SimAM), ECA showed 0.0413 (1.88×), and CA exhibited 0.0362 (1.65×), all significantly higher than SimAM’s 0.0220, exposing serious overconfidence issues. CBAM demonstrated anomalous Top-1 accuracy (59.79%, far below the Baseline’s 61.75%), possibly due to excessive feature suppression introduced by its serial channel-spatial attention mechanism in shallow networks. Leveraging its parameter-free energy function mechanism, SimAM maintained the minimal 0.967 M parameters and 332.367 M FLOPs while achieving the highest Top-1 accuracy (64.18%) and Macro-F1 score (0.5788) among all evaluated methods, along with optimal calibration performance (ECE = 0.0220).

Table 9. Performance comparison of different attention mechanisms on the IP102 dataset.

Attention	Type	Params (M)	FLOPs (M)	Top-1 acc (%)	ECE	Macro-F1
Baseline (None)	N/A	0.974	354.432	61.75	0.0330	0.5413
SE	Channel	1.169	334.203	63.18	0.0471	0.5509
CBAM	Ch+Sp	1.170	336.192	59.79	0.0263	0.5169
ECA	Channel	0.967	334.017	63.80	0.0413	0.5564
CA	Coordinate	1.270	338.428	63.41	0.0362	0.5613
SimAM	Energy	0.967	332.367	64.18	0.0220	0.5788

Note: Bold values indicate the best performance.

3.5. Feature Visualization and Interpretive Analysis

To further validate the reliability and decision-making mechanism of MOIR-Net, Gradient-Weighted Class Activation Map (Grad-CAM) [31] was employed to visualize the regions of interest (ROIs) highlighted by the model. Four representative high-difficulty images were selected from the IP102 test set, covering minute targets (first row), mimetic camouflage (second row), complex backgrounds (third row), and occlusion scenarios (fourth row). MOIR-Net was compared against three state-of-the-art lightweight models (MobileViT-v2, EfficientNet-B0, RepViT-M1.0). Visualization results are shown in Figure 11.

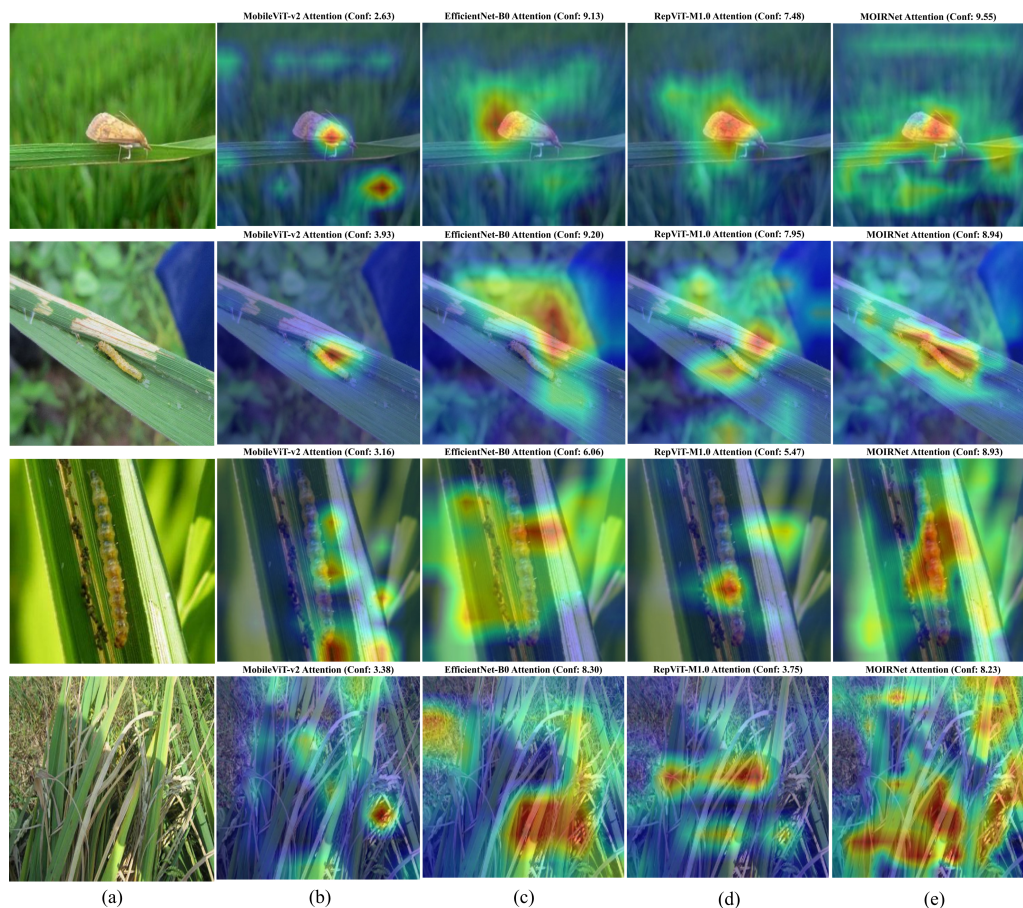


Figure 11. Grad-CAM feature visualization comparison. (a) Original image; (b) MobileViT-v2 heatmap; (c) EfficientNet-B0 heatmap; (d) RepViT-M1.0 heatmap; (e) MOIR-Net heatmap.

Results demonstrate that MOIR-Net's heatmaps exhibit clear, compact red high-activation regions, precisely focusing on disease entities with sharp boundaries. For small targets (aphids with diameter <10 pixels), while MobileViT-v2's attention disperses, EfficientNet-B0's boundaries blur, and RepViT-M1.0's activation intensity weakens, MOIR-Net achieves precise localization. In camouflage scenarios, MOIR-Net's high-activation regions align closely with pest contours while effectively suppressing background region activation. Under complex background interference, MOIR-Net demonstrates "spotlight-like" focusing effects with complete suppression of weeds, soil. From the perspective of the prediction confidence level of the attention heatmap, the confidence level of MOIR-Net in identifying pest and disease areas (such as 8.93, 8.23, etc.) is generally higher than that of MobileViT-v2 (such as 3.16, 3.18, etc.), EfficientNet-B0 (such as 6.06, 8.30, etc.) and RepViT-M1.0 (such as 5.47, 3.75, etc.), indicating that it is more confident in the feature discrimination of the target area and has a more stable attention focusing effect. Visualization clearly reveals MOIR-Net's decision mechanism: multi-scale feature fusion captures multi-level information while SimAM energy attention adaptively enhances disease regions and suppresses background noise, enabling robust discriminative capability in extremely complex scenarios.

4. Discussion

MOIR-Net was designed around a single principle: decoupling the training-time and inference-time topologies so that representational richness and deployment efficiency are no longer in direct conflict. The results across the three datasets support this design. On IP102, the multi-scale omni-kernel branches and the parameter-free SimAM

attention together improve both fine-grained discrimination (mAP) and long-tail balance (Macro-F1), while the inference graph collapses to a single 3×3 convolution, yielding the lowest measured single-image latency among the compared models. On PlantVillage the model is competitive with much larger networks at a fraction of the parameters, and on PlantDoc it exhibits the smallest macro-average F1 degradation under a strict zero-shot protocol.

Compared with similar studies on the same datasets, the IP102 and PlantVillage accuracies obtained here are comparable to or exceed those typically reported by recent lightweight and attention-based agricultural recognition pipelines—including YOLOv11-based crop-disease detectors and transformer-/attention-enhanced classifiers and segmenters (e.g., for tomato, pomegranate, and general plant-disease recognition, and SegFormer-based lesion segmentation; see the additional references listed at the end of this paper)—while requiring substantially fewer parameters and lower latency. Related image-processing and machine-learning studies on super-resolution, entropy-based edge detection, and feature reduction further motivate the lightweight, reliability-oriented design adopted here.

From a deployment-cost perspective, all experiments were conducted on a consumer-grade NVIDIA GeForce RTX 4060 Ti GPU using an entirely open-source software stack (PyTorch, torchvision, CUDA, and the THOP profiler). The proposed pipeline therefore does not depend on specialized or proprietary hardware and can be reproduced at low cost, which is consistent with the goal of accessible edge deployment in precision agriculture.

5. Limitations

Several limitations of this study should be acknowledged. First, all reported metrics correspond to a single training run; although the IP102 improvements were assessed with a paired *t*-test over per-class scores, a full multi-run analysis reporting mean and standard deviation would more rigorously characterize variance and is left for future work. Second, the cross-domain evaluation on PlantDoc follows a strict zero-shot protocol, so the absolute Top-1 accuracy is low for all models; this reflects the difficulty of the laboratory-to-field domain shift rather than a failure mode specific to MOIR-Net, and fine-tuning or domain adaptation would be required for practical field accuracy. Third, the efficiency metrics (latency, throughput, and memory) were measured on a desktop GPU; on-device measurements on embedded accelerators (e.g., Jetson- or Raspberry-Pi-class hardware) remain to be validated. Finally, the present study addresses image-level classification only; extending the MOIR design to detection and segmentation, and to a broader range of crops and field conditions, is an important direction for future work.

6. Conclusions

This study proposes MOIR-Net, a lightweight crop disease and pest recognition network based on multi-scale reparameterization and energy attention mechanisms. The model achieves 64.18% Top-1 accuracy on the large-scale IP102 dataset, outperforming the second-best model MobileViT-v2 by 1.41 percentage points, while requiring only 0.967 M parameters and 2.30 ms inference latency, significantly superior to existing lightweight networks. In cross-domain generalization tests, MOIR-Net attains a Macro-F1 of 0.2949 on the unstructured PlantDoc field dataset, representing a 41.9% improvement over comparison models with only 74.76% performance degradation, demonstrating strong robustness in unstructured environments. The Expected Calibration Error (ECE) of 0.0220 is the lowest among all compared models, ensuring highly reliable prediction confidence. In summary, MOIR-Net achieves “zero-cost” acceleration through structural reparameterization and enhances feature discriminability via energy attention, providing an efficient, trustworthy, and scalable real-time disease monitoring solution for resource-constrained agricultural edge devices, facilitating the practical deployment of precision agriculture and intelligent plant protection technologies. Future work will focus on (i) multi-run experiments with variance reporting to strengthen statistical rigor; (ii) lightweight domain-adaptation and few-shot fine-tuning to raise the absolute field accuracy on datasets such as PlantDoc; (iii) on-device benchmarking and deployment on embedded edge accelerators; and (iv) extending the MOIR block to object detection and semantic segmentation for in-field pest localization.

Author Contributions

Y.P.: Roles/Writing—original draft, Conceptualization, Formal analysis, Methodology, Experimental analysis and interpretation of data, Software, Visualization; L.L.: Writing—review & editing, Project administration; S.X.: Formal analysis, Writing—review & editing; J.C.: Formal analysis, Validation, Funding acquisition, Resources, Supervision; H.W.: Formal analysis, Methodology, Supervision; B.T.: Formal analysis, Software; Y.L.: Investigation, Visualization; T.L.: Software, Formal analysis; Y.Y.: Formal analysis, Validation; L.Y.: Validation, Formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the Natural Science Foundation of Chongqing, China (Grant No. CSTB2025NSCQ-LZX0123); the Chongqing Natural Science Foundation Innovation and Development Joint Fund Project (Wanzhou District) (Grant No. CSTB2025NSCQ-WZQLHJJMSX0012); the Science and Technology Research Program of Chongqing Education Commission (Grant No. KJZD-K20230120); and the Science and Technology Research Project of Chongqing Municipal Education Commission (Grant No. KJZD-K202501201).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data that support the findings of this study are not publicly available but will be made available by the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used ChatGPT to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Savary, S.; Willocquet, L.; Pethybridge, S.J.; et al. The Global Burden of Pathogens and Pests on Major Food Crops. *Nat. Ecol. Evol.* **2019**, *3*, 430–439. <https://doi.org/10.1038/s41559-018-0793-y>.
2. Barbedo, J.G.A. Factors Influencing the Use of Deep Learning for Plant Disease Recognition. *Biosyst. Eng.* **2018**, *172*, 84–91.
3. Tembhurne, J.V.; Gajbhiye, S.M.; Gannarpwar, V.R.; et al. Plant Disease Detection Using Deep Learning Based Mobile Application. *Multimed. Tools Appl.* **2023**, *82*, 27365–27390. <https://doi.org/10.1007/s11042-023-14541-8>.
4. Chithambarathanu, M.; Jeyakumar, M.K. Survey on Crop Pest Detection Using Deep Learning and Machine Learning Approaches. *Multimed. Tools Appl.* **2023**, *82*, 42277–42310. <https://doi.org/10.1007/s11042-023-15221-3>.
5. He, K.; Zhang, X.; Ren, S.; et al. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. <https://doi.org/10.1109/cvpr.2016.90>.
6. Huang, G.; Liu, Z.; Van Der Maaten, L.; et al. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. <https://doi.org/10.1109/cvpr.2017.243>.
7. Howard, A.; Sandler, M.; Chu, G.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, 27 October–2 November 2019; pp. 1314–1324. <https://doi.org/10.1109/iccv.2019.00140>.
8. Ma, N.; Zhang, X.; Zheng, H.T.; et al. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
9. Wu, X.; Zhan, C.; Lai, Y.K.; et al. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8787–8796. <https://doi.org/10.1109/cvpr.2019.00899>.
10. Mehta, S.; Rastegari, M. Separable Self-Attention for Mobile Vision Transformers. *arXiv* **2022**, arXiv:2206.02680. <https://doi.org/10.48550/arxiv.2206.02680>.
11. Maaz, M.; Shaker, A.; Cholakkal, H.; et al. EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications. In Proceedings of the Computer Vision—ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022; pp. 3–20. https://doi.org/10.1007/978-3-031-25082-8_1.

12. Liu, X.; Peng, H.; Zheng, N.; et al. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 14420–14430. <https://doi.org/10.1109/cvpr52729.2023.01386>.
13. Wang, A.; Chen, H.; Lin, Z.; et al. RepViT: Revisiting Mobile CNN from Vit Perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 15909–15920.
14. Hughes, D.P.; Salathe, M. An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics. *arXiv* **2015**, arXiv:1511.08060. <https://doi.org/10.48550/arxiv.1511.08060>.
15. Singh, D.; Jain, N.; Jain, P.; et al. PlantDoc: A Dataset for Visual Plant Disease Detection. *arXiv* **2019**, arXiv:1911.10317. <https://doi.org/10.48550/arxiv.1911.10317>.
16. Ding, X.; Zhang, X.; Ma, N.; et al. RepvGG: Making VGG-Style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742. <https://doi.org/10.1109/cvpr46437.2021.01352>.
17. Yang, L.; Zhang, R.Y.; Li, L.; et al. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021; pp. 11863–11874.
18. Dejanović, M.; Panić, S.; Kontrec, N.A.; et al. Neural Network-Based Optimization of Repair Rate Estimation in Performance-Based Logistics Systems. *Information* **2025**, *16*, 1031. <https://doi.org/10.3390/info16121031>.
19. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, 18–22 June 2018; pp. 7132–7141. <https://doi.org/10.1109/cvpr.2018.00745>.
20. Woo, S.; Park, J.; Lee, J.Y.; et al. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
21. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; et al. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. <https://doi.org/10.1109/cvpr.2016.308>.
22. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101. <https://doi.org/10.48550/arxiv.1711.05101>.
23. Erickson, B.J.; Korfiatis, P.; Akkus, Z.; et al. Toolkits and Libraries for Deep Learning. *J. Digit. Imaging* **2017**, *30*, 400–405. <https://doi.org/10.1007/s10278-017-9965-6>.
24. Guo, C.; Pleiss, G.; Sun, Y.; et al. On Calibration of Modern Neural Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
25. Sandler, M.; Howard, A.; Zhu, M.; et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. <https://doi.org/10.1109/cvpr.2018.00474>.
26. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
27. Han, K.; Wang, Y.; Tian, Q.; et al. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
28. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; et al. MobileOne: An Improved One Millisecond Mobile Backbone. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7907–7917. <https://doi.org/10.1109/cvpr52729.2023.00764>.
29. Wang, Q.; Wu, B.; Zhu, P.; et al. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542. <https://doi.org/10.1109/cvpr42600.2020.01155>.
30. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722. <https://doi.org/10.1109/cvpr46437.2021.01350>.
31. Selvaraju, R.R.; Cogswell, M.; Das, A.; et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626. <https://doi.org/10.1109/iccv.2017.74>.