applied sciences

MDPI

Article

LEA: A Lightweight Local Enhanced Attention Module with Local-Competitive Spatial Recalibration for CNN-Based Agricultural Image Classification

Kepeng Luo, Zhanyi Lian, Xin Han and Yunfeng Wang *

School of Electronic Science and Engineering, Xiamen University, Xiamen 361100, China; 36120241150535@stu.xmu.edu.cn (K.L.); 36120241150566@stu.xmu.edu.cn (Z.L.)
* Correspondence: yunfengwang@xmu.edu.cn

Abstract

Local Enhanced Attention (LEA) adds neighborhood aggregation to a learned spatial gate for CNN-based agricultural image classification. The module combines strip filtering, channel-wise normalization and sigmoid gating, followed by aggregation without additional learned parameters. In a seven-method PlantSeg comparison, LEA recorded the highest mean lesion CAM energy (41.612%) and tied with SE in mean pointing accuracy (52.054%). Its PlantVillage mean accuracy was 99.902%, compared with 99.883% for baseline. The measured ResNet-50 configuration added 0.946% parameters, with higher latency and memory use. Classification, binary CAM overlap and corruption results varied across settings. These results support the use of LEA in agricultural classification workflows that inspect disease-related spatial cues.

Keywords: lightweight attention mechanism; convolutional neural networks; spatial attention; ImageNet-1K; agricultural image classification; plant disease classification; pest and weed classification

1. Introduction

Image-based recognition of pests, diseases and weeds supports crop protection decisions [1,2]. Field images often place small diagnostic regions within cluttered vegetation, with variable illumination and partial occlusion. PlantDoc [2], DeepWeeds [3] and natural scene studies such as FAF-Net [4] reflect the need to evaluate recognition under these conditions.

Earlier agricultural vision systems combined color, texture, shape or edge descriptors with classifiers such as support vector machines and random forests [5–7]. These handcrafted representations are sensitive to shadows, occlusion and background clutter, particularly when the target occupies a small image region.

Convolutional neural networks (CNNs) learn hierarchical image features directly and have been widely applied to plant disease recognition [8]. CNN–Transformer hybrids and Vision Transformer models have also been explored for pest and disease classification [9,10]. Their accuracy and deployment costs depend on architecture and training data, making matched comparisons essential when assessing lightweight CNN alternatives.

Agricultural classification often depends on fine local cues, including insect wing veins, lesion textures and weed leaf margins. Background clutter, illumination changes, occlusion and similar-looking classes can obscure these cues or encourage reliance on image

Check for updates

Academic Editor: Atsushi Mase

Received: 13 October 2026
Revised: 31 October 2026
Accepted: 2 November 2026
Published: 9 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

context. A useful feature-enhancement module should retain these details at an acceptable computational cost.

Attention modules recalibrate CNN features using channel, spatial or combined information. Squeeze-and-Excitation (SE) [11] and Efficient Channel Attention (ECA) [12] estimate channel weights. The Convolutional Block Attention Module (CBAM) [13] and Triplet Attention (TA) [14] model channel–spatial or cross-dimensional interactions. Coordinate Attention (CA) [15] and Efficient Multi-Scale Attention (EMA) [16] incorporate positional or multi-scale information. Agricultural models such as RAI-Net [17] and AgriFewNet [18] use attention to refine disease-related features or support lightweight feature extraction.

These mechanisms impose different inductive biases. SE and ECA pool spatial information into channel weights, whereas CBAM and CA retain spatial information in different forms. Local filtering is already central to convolutional attention. For example, Large Separable Kernel Attention (LSKA) [19] uses cascaded one-dimensional depthwise kernels, and SoftPool [20] uses exponential-weighted local averaging. The question examined here is whether neighborhood aggregation of a learned bounded gate benefits agricultural classification when inserted into existing CNN blocks.

The Separable Convolution Module (SCM) applies orthogonal depthwise strip convolutions followed by affine channel-wise GroupNorm. A separate sigmoid bounds its output before Local Softmax Modulation (LSM) aggregates neighboring gates. The aggregation uses position-dependent local-maximum offsets and pooling, and the resulting map multiplies the input features. ResNet uses one spatial gate per channel; the conceptual MobileNetV2 variant adds a pointwise projection to obtain one shared gate. This composition retains spatial resolution and allows local gate aggregation to be studied within existing CNN blocks.

The study makes three contributions:

- We formulate a spatial recalibration module combining strip filtering, bounded gating and same-resolution neighborhood aggregation, with channel-wise and channel-shared variants.
- We define the implemented gate, its bounded weights and boundary rules, and we evaluate exact exponential-weighted pooling as separate formula controls.
- We assess component controls, class-level behavior, image corruptions and GPU cost to characterize accuracy, attribution and computational trade-offs.

LEA aggregates neighboring values of a learned bounded spatial gate before multiplying that gate with CNN features. This neighborhood step distinguishes the tested composition from strip filtering or normalization alone.

2. Related Work

2.1. Attention Mechanisms in CNNs

CNN attention mechanisms assign input-dependent weights to channel or spatial feature responses.

SE [11] uses global pooling to estimate inter-channel weights. ECA [12] instead models local cross-channel interactions without reducing channel dimensionality. Both approaches provide channel recalibration with a small parameter increment.

Subsequent designs incorporated spatial information. CBAM [13] applies channel and spatial attention sequentially, whereas TA [14] uses tensor rotations to model cross-dimensional dependencies. CA [15] encodes position through direction-aware aggregation. EMA [16] combines grouped multi-scale processing with cross-spatial learning. Pyramid Channel and Spatial Attention (PCSA) [21] couples pyramid features with channel and spatial weighting.

Convolution-based attention is especially relevant to LEA. Visual Attention Network (VAN) [22] combines depthwise, dilated depthwise and pointwise convolutions to construct large-kernel attention. LSKA [19] factorizes spatial kernels into horizontal and vertical operations. LEA builds on these established filtering operations. Its ResNet branch combines small strip filters, affine channel-wise GroupNorm, sigmoid gating and neighborhood aggregation; the conceptual MobileNetV2 branch additionally projects to one channel.

The earlier LSM formula is related to SoftPool [20], but the original implementation uses a different local maximum at each input position. These offsets do not generally cancel within an output window. Section 3.3 therefore distinguishes the implemented operator from exact formula controls. Both are applied to sigmoid gates at stride one before feature multiplication.

Spatial and Channel Synergistic Attention (SCSA) [23] combines multi-semantic spatial guidance with channel self-attention. It is evaluated under the PlantSeg fine-tuning protocol in Section 4.5.6.

2.2. Agricultural Pest, Plant Disease, and Weed Classification

Plant disease, pest and weed classification present different imaging conditions. PlantVillage contains largely centered leaves against controlled backgrounds and supports high CNN classification accuracy [8]. PlantDoc instead contains images from non-controlled environments [2]. FAF-Net [4] and an ensemble evaluated on PlantVillage, PlantDoc and FieldPlant [24] address localized symptoms and variation across imaging settings.

The Dangerous Farm Insects Image Dataset supports fine-grained harmful insect classification [25]. ConvViT [9] combines local CNN features with Transformer-based global modeling for this task. DeepWeeds contains weed images from Australian rangelands, where surrounding vegetation complicates recognition [3].

Recent agricultural models use multi-scale features, multiple views and attention in different ways. RAI-Net [17] combines residual learning, attention and Inception modules for tomato disease classification. Multi-ViT [10] aggregates multiple leaf images through Transformer branches and attention scores. AgriFewNet [18] combines spatial and channel attention in a lightweight few-shot framework. RyNet [26] fuses low-, mid- and high-level features and evaluates attention-enhanced CNNs on CIFAR-10 and PlantVillage. EFM-ResNet [27] uses multi-scale fusion and feature enhancement for tobacco strip classification.

These systems address different tasks and training regimes, whereas LEA is evaluated as an insertable module across insect, disease and weed classification. Comparisons with complete agricultural systems such as ConvViT [9] require aligned data partitions and training settings.

3. Materials and Methods

3.1. Overview of LEA

LEA applies spatially varying, channel-wise gates to intermediate CNN features. Figure 1 shows the agricultural evaluation pipeline and gate branch; Figure 2 details LEA. The controlled experiments implement the attention modules within torchvision ResNet blocks.

For features $X \in \mathbb{R}^{B\times C\times H\times W}$, let D_K denote successive depthwise convolutions with $K \times 1$ and $1 \times K$ kernels. SCM includes normalization but excludes the sigmoid:

$$Z = \mathrm{SCM}(X) = \mathrm{GN}_C(D_K(X)), \qquad A = \sigma(Z), \qquad Y = X \odot \mathcal{L}(A). \tag{1}$$

Here, GN_C is GroupNorm with C groups, one per channel. The operator $\mathcal{L}$ denotes either the original implementation or an explicitly named formula control. The original LEA is $X \odot \mathcal{L}_{\mathrm{orig}}(A)$. The gate preserves the spatial dimensions H and W.

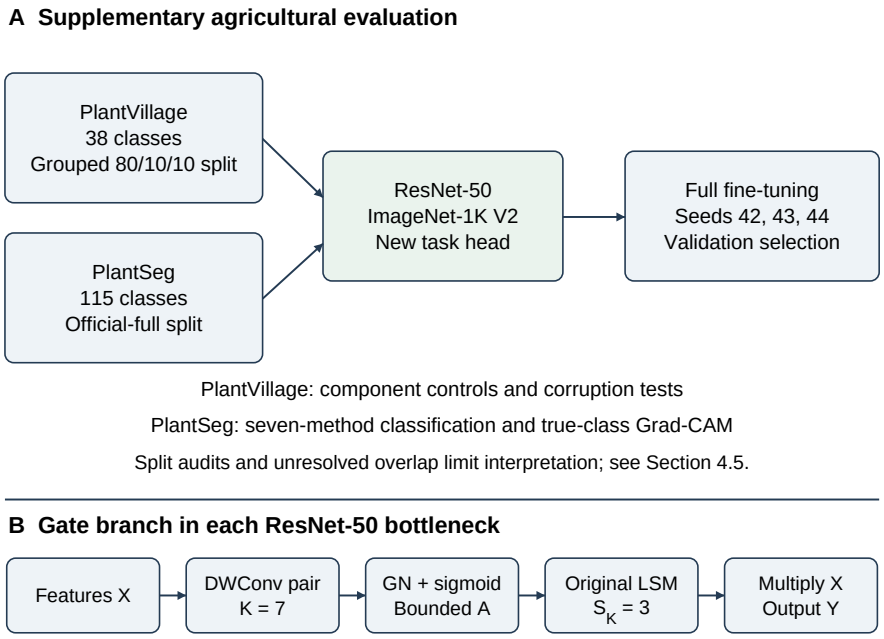


Figure 1. Supplementary agricultural evaluation and the LEA gate branch. PlantVillage uses an exact-duplicate-grouped split; PlantSeg uses the official full split. All models are fine-tuned, and checkpoints are selected on validation accuracy. DWConv denotes depthwise convolution; GN denotes affine GroupNorm with one group per channel. In the lower panel, LSM denotes the implemented original aggregation. Experimental details are given in Section 4.5.

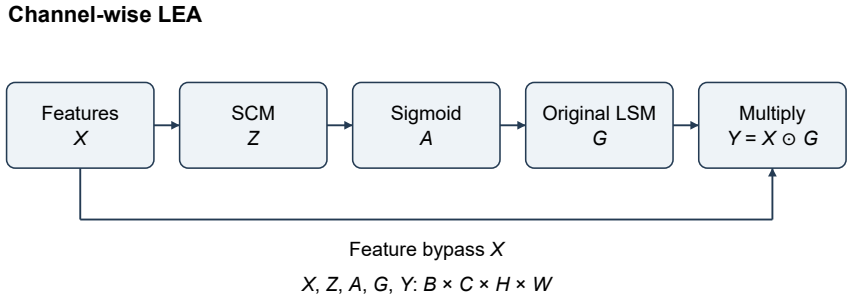


Figure 2. Channel-wise LEA with an explicit feature bypass. SCM produces the pre-sigmoid output Z; sigmoid gives A; the original LSM produces the gate G used in $Y = X \odot G$. All tensors retain spatial size $H \times W$.

3.2. Separable Convolution Module

SCM uses bias-free depthwise convolutions with stride one and symmetric zero padding (Figure 3). For $K = 7$, the paddings are $(3, 0)$ and $(0, 3)$. Normalization uses GroupNorm(C, C) with learnable scale and bias, $\epsilon = 10^{-5}$ and no running statistics. Statistics are computed over spatial positions within each sample and channel, as in affine InstanceNorm without running statistics. The gate therefore depends on both local filtering and spatial normalization.

$$D_K(X) = \mathrm{DWConv}_{1\times K}\big(\mathrm{DWConv}_{K\times 1}(X)\big), \quad Z_{b,c,i,j} = \gamma_c \frac{D_K(X)_{b,c,i,j} - \mu_{b,c}}{\sqrt{v_{b,c} + 10^{-5}}} + \beta_c. \quad (2)$$

The sigmoid maps Z to gate values in $[0, 1]$ in finite precision. The controlled comparison tests whether subsequent aggregation adds value beyond these established filtering and normalization operations. The controlled experiments fix $K = 7$ and $S_K = 3$ before evaluation.

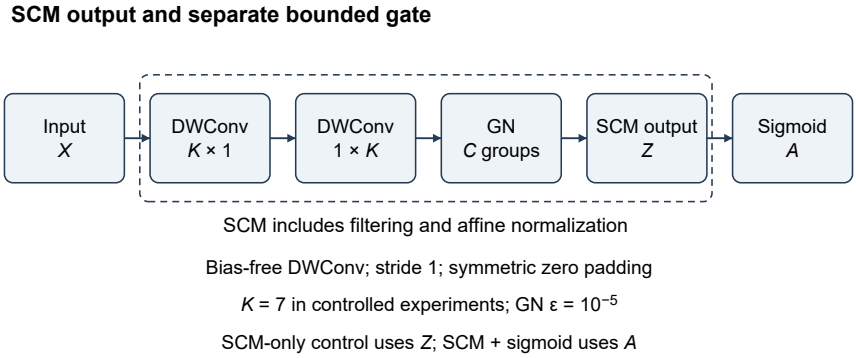


Figure 3. SCM and the separately applied sigmoid. DWConv denotes bias-free depthwise convolution; GN denotes affine GroupNorm with C groups and C channels. SCM ends at Z, before sigmoid, making the component-control definitions explicit.

Design Purpose and Testable Properties

LEA changes gate construction while retaining feature resolution. For finite sigmoid outputs, $0 \le a_q \le 1$ and $a_q \le m_q \le 1$, giving $-1 \le a_q - m_q \le 0$ and $e^{-1} \le b_q \le 1$. Bounded gates thus limit the exponential weight range. Strip filters use fewer weights than square depthwise filters of equal width, and aggregation adds no trainable parameters. Latency and memory are measured separately. Component controls compare the neighborhood rule with sigmoid gating and uniform averaging.

3.3. Original Local Aggregation and Formula Controls

The original LSM implementation differs from the local softmax-weighted formula previously used to describe it (Figure 4). For one sample and channel, let a_q be a sigmoid gate. Let V_p contain valid positions in the $S_K \times S_K$ window centered at p. The original implementation first computes

$$m_q = \max_{r \in V_q} a_r, \qquad b_q = \exp(a_q - m_q). \tag{3}$$

It then applies zero-padded average pooling to $a_q b_q$ and b_q separately and divides the results. In real arithmetic, with the default padding included in the averaging divisor, this is

$$\mathcal{L}_{\text{orig}}(A)_p = \frac{\sum_{q \in V_p} a_q b_q}{\sum_{q \in V_p} b_q + S_K^2 \epsilon_{\text{o}}}, \qquad \epsilon_{\text{o}} = 10^{-8}. \tag{4}$$

The offsets m_q vary within an output window and generally remain in the ratio. The operator therefore differs algebraically from local SoftPool, including away from image boundaries. Padding contributes zero mass after exponentiation, and local maxima use valid image positions.

For comparison, the exponential-weighted local mean underlying SoftPool [20] is implemented in two separately named controls:

$$\mathcal{L}_{\text{valid}}(A)_p = \frac{\sum_{q \in V_p} a_q e^{a_q}}{\sum_{q \in V_p} e^{a_q} + \epsilon_{\text{f}}}, \tag{5}$$

$$\mathcal{L}_{\text{zero}}(A)_p = \frac{\sum_{q \in V_p} a_q e^{a_q}}{\sum_{q \in V_p} e^{a_q} + (S_K^2 - |V_p|) + \epsilon_{\text{f}}}, \qquad \epsilon_{\text{f}} = 10^{-6}. \tag{6}$$

The valid control excludes padding from both sums. The zero control includes padded zero values, each contributing $e^0 = 1$ to the denominator. Both controls use pooling during

training; unfolding serves only as a numerical reference. Their scores are reported separately from original LEA.

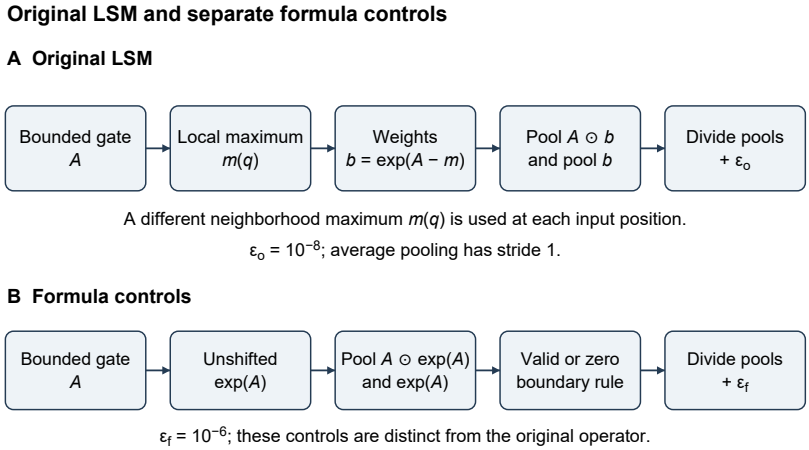


Figure 4. Original LSM and separately evaluated formula controls. Position-dependent local maxima m_q produce the original weights b_q. Both numerator and denominator are pooled before division. The formula controls instead pool unshifted exponentials and apply the stated valid or zero boundary convention. Pooling uses stride one throughout.

3.3.1. Aggregation Properties

Without the stabilizer, the exact formula is a convex weighted average bounded by the window's input range. For sigmoid inputs, the ratio of any two exponential weights is at most e. A 3 × 3 window with one value of 0.9 and eight values of 0.1 yields approximately 0.274, illustrating attenuation of an isolated peak. Adding a positive denominator stabilizer retains the zero-to-maximum bound for non-negative gates but removes the exact minimum-input lower bound. The original operator also aggregates non-negative bounded gates, with weights determined partly by position-specific maxima. Its weights need not follow gate-value order. Both operators rescale rather than amplify individual feature magnitudes because their gates are bounded by one.

3.3.2. Evaluated Operators

In the controlled ResNet-50 experiments, original LEA uses Equations (3) and (4). Formula-valid and formula-zero are evaluated as distinct controls with the boundary rules specified above.

3.4. Integration into CNN Backbones

In the supplementary ResNet experiments, LEA is inserted into every residual block after its final BatchNorm. It precedes shortcut addition and the final ReLU.

For ResNet-style backbones [28], this placement modulates the main branch before it is added to the shortcut. Given the block input X, the main-branch mapping $F(\cdot)$, and the shortcut mapping $\delta(\cdot)$, the output of an LEA-enhanced residual block is written as

$$Y = \mathrm{ReLU}\big(\delta(X) + \mathrm{LEA}(F(X))\big), \tag{7}$$

where $\delta(X)$ denotes the shortcut output. It is an identity mapping when the dimensions are matched and a projection shortcut when dimension matching is required.

The conceptual MobileNetV2 design uses a channel-shared gate after the final point-wise projection in each inverted residual block. LEA precedes element-wise addition when a shortcut is present and otherwise modulates the projected block output. The channel-shared variant, $\mathrm{SCM_L}$, reduces the attention map from C channels to one.

As shown in Figure 5, SCM_L adds a (1×1) pointwise convolution after the two strip depthwise convolutions to reduce the channel dimension from C to 1:

$$U_L = P(D_K(X)), \qquad P : \mathbb{R}^{B\times C\times H\times W} \to \mathbb{R}^{B\times 1\times H\times W}. \tag{8}$$

The projected map is normalized, passed through sigmoid and LSM, and broadcast across feature channels. Here, P is a learned 1×1 convolution and $SCM_L(X) = IN(P(D_K(X)))$. The resulting gate is shared across feature channels.

The insertion positions of LEA in ResNet and MobileNetV2 are summarized in Figure 6.

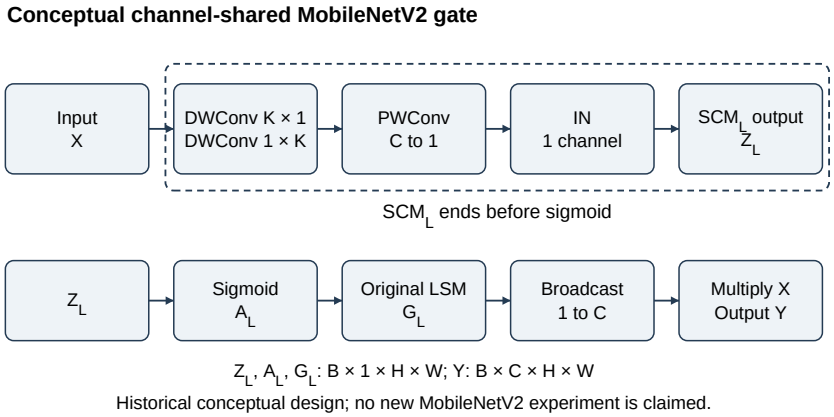


Figure 5. Conceptual channel-shared MobileNetV2 variant. PWConv denotes a pointwise convolution, and IN denotes instance normalization. The SCM-L branch reduces C channels to one and ends at Z_L; sigmoid, original LSM and channel broadcasting follow. This panel illustrates the conceptual channel-shared design.

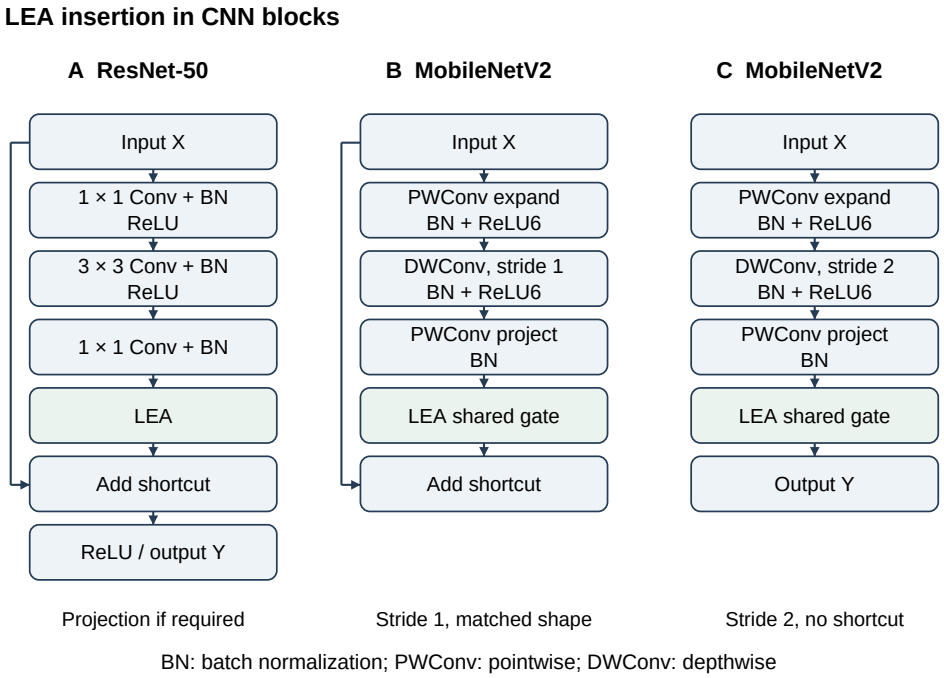


Figure 6. LEA insertion in a ResNet-50 bottleneck and conceptual MobileNetV2 inverted residual blocks. BN denotes batch normalization; PWConv and DWConv denote pointwise and depthwise convolutions. LEA follows the final BN and precedes shortcut addition when present. A MobileNetV2 identity shortcut requires stride one and matching input/output shapes; the ResNet shortcut uses a projection when required.

3.5. Computational Complexity

The additional learnable parameters of the standard SCM mainly come from the two orthogonal strip depthwise convolutions. For an input feature tensor with C channels and a strip kernel size K, the convolutional parameter count of SCM, ignoring bias and normalization parameters, is

$$P_{\mathrm{SCM}} = 2KC. \tag{9}$$

A standard $K \times K$ square depthwise convolution requires K^2C parameters. Therefore, the parameter ratio between SCM and the standard square depthwise convolution is

$$\frac{P_{\mathrm{SCM}}}{P_{K\times K}} = \frac{2}{K}. \tag{10}$$

For example, when $K = 7$, the convolutional weights use approximately 28.6% of those in a 7×7 square depthwise convolution. Affine GroupNorm adds $2C$ parameters, giving $(2K + 2)C$ added parameters per channel-wise block, with no convolution bias.

For the MobileNetV2 variant, $\mathrm{SCM}_{\mathrm{L}}$ additionally uses a (1×1) convolution to map C channels to one shared attention channel. Ignoring bias and normalization parameters, its convolutional parameter count is

$$P_{\mathrm{SCM}_{\mathrm{L}}} = 2KC + C. \tag{11}$$

The projection produces a single spatial channel and adds a small number of parameters.

The computational complexity of the standard SCM is $O(2KBCHW)$. For $\mathrm{SCM}_{\mathrm{L}}$, the additional (1×1) projection introduces $O(BCHW)$ operations, giving a convolutional complexity of $O((2K + 1)BCHW)$. LSM has no learnable parameters. Its channel-wise complexity is $O(S_K^2 BCHW)$, reduced to $O(S_K^2 BHW)$ for a shared gate before broadcasting. An unfolded reference can require $O(BCHWS_K^2)$ or $O(BHWS_K^2)$ intermediate storage, respectively; the evaluated implementations use pooling. We report Conv/Linear MACs alongside measured latency and peak GPU memory to describe both arithmetic and execution cost.

4. Experimental Results and Analysis

The evaluation covers general image classification, agricultural recognition, component ablations, image corruptions and spatial attribution. Sections 4.2–4.4 present the previously reported comparisons; Section 4.5 presents controlled ResNet-50 experiments under the fixed protocols below. These experiment sets use different implementations and data partitions and are presented separately. ImageNet-100 and entries without an identifiable configuration or cost definition are excluded; the supporting audit records are supplied in the Supplementary Materials.

4.1. Experimental Setup

The general classification experiments cover ImageNet-1K [29] and CIFAR-100 [30], representing large-scale and low-resolution inputs, respectively.

Four datasets cover agricultural tasks. The Dangerous Farm Insects [25] dataset concerns fine-grained pest classification. PlantVillage [8] and PlantDoc [2] concern disease classification under controlled and natural scene conditions, respectively. The DeepWeeds [3] dataset concerns weed classification in natural vegetation.

Table 1 lists the dataset counts for these comparisons. ImageNet-1K uses its official validation set and CIFAR-100 its official test set. The PlantVillage and DeepWeeds literature reference rows are identified separately from the in-study agricultural comparisons.

Table 1. Dataset counts for the general and agricultural classification comparisons.

Dataset	Classes	Training Images	Evaluation Images
ImageNet-1K	1000	1,281,167	50,000
CIFAR-100	100	50,000	10,000
Dangerous Farm Insects	15	1263	316
PlantVillage	38	43,444	10,861
PlantDoc	28	2342	236
DeepWeeds	9	10,501	3507

The reported general classification recipe used ResNet-18, ResNet-50 [28] and MobileNetV2 [31], with attention inserted into residual or inverted residual blocks. Training from scratch used seed 42 on one NVIDIA GeForce RTX 4090.

The reported ImageNet-1K recipe used 224 × 224 inputs, random resized cropping and random horizontal flipping. Models were trained for 90 epochs with SGD, batch size 128, momentum 0.9 and initial learning rate 0.1. The learning rate decreased by a factor of 0.1 at epochs 30 and 60. Evaluation used resizing, a 224 × 224 center crop and Top-1 accuracy.

The reported CIFAR-100 recipe used 32 × 32 images with random cropping and horizontal flipping during training. Models were trained for 300 epochs with SGD, batch size 128, initial learning rate 0.1 and a multi-step decay schedule. Evaluation used the original 32 × 32 images without random augmentation.

The agricultural recipe fine-tuned ImageNet-1K-pretrained ResNet-50 for 50 epochs, using 224 × 224 inputs and ImageNet normalization. Batch sizes were 16 for Dangerous Farm Insects, PlantDoc and DeepWeeds, and 64 for PlantVillage. Dangerous Farm Insects and DeepWeeds used Adam with initial learning rate 1×10^{-4}. PlantVillage and PlantDoc used AdamW with initial learning rates 1×10^{-4} and 5×10^{-5}, respectively.

General classification tables report Top-1 accuracy and, where retained, parameter counts and FLOPs. Agricultural tables report accuracy (Acc.), Macro-F1 and Weighted-F1 to describe overall, class-averaged and support-weighted performance.

Dangerous Farm Insects and PlantVillage use the reported 1263/316 and 43,444/10,861 training/evaluation partitions, approximately 80/20. PlantDoc lists 2342/236 images across 28 classes [2], and DeepWeeds lists 10,501/3507 images. The supplementary dataset audit documents partition roles, class mappings and available manifests.

ImageNet initialization provides a starting representation for agricultural fine-tuning [8]. The controlled experiments use full-network fine-tuning and validation-based checkpoint selection, as specified in Section 4.5.1.

In the controlled experiments, K, S_K, augmentation and class-balancing settings are fixed before testing. Checkpoints are selected by validation accuracy, and repeated-run results are reported as means and sample standard deviations.

4.2. General Image Classification

The general classification tables cover three ImageNet-1K backbones, CIFAR-style ResNet depths and MobileNetV2 width multipliers.

4.2.1. Results on ImageNet-1K

Table 2 reports ImageNet-1K accuracy for ResNet-18, ResNet-50 and MobileNetV2. Common-head architecture costs are examined in Section 4.5.3.

Table 2. ImageNet-1K classification accuracy across three backbones.

Backbone	Method	Top-1 (%)
ResNet-18	Baseline [28]	69.76
	SE [11]	70.59 (+0.83)
	BAM [32]	**71.12 (+1.36)**
	CBAM [13]	70.73 (+0.97)
	TA [14]	71.09 (+1.33)
	EMCA [33]	71.00 (+1.24)
	LEA (Ours)	71.10 (+1.34)
ResNet-50	Baseline [28]	76.13
	SE [11]	76.71 (+0.58)
	BAM [32]	75.98 (−0.15)
	CBAM [13]	77.34 (+1.21)
	GSoP-Net [34]	77.68 (+1.55)
	A2-Nets [35]	77.00 (+0.87)
	GCNet [36]	77.70 (+1.57)
	GALA [37]	77.27 (+1.14)
	ABN [38]	76.90 (+0.77)
	SRM [39]	77.13 (+1.00)
	TA [14]	77.48 (+1.35)
	EMCA [33]	77.33 (+1.20)
	ECA [12]	77.48 (+1.35)
	LEA (Ours)	**78.03 (+1.90)**
MobileNetV2	Baseline [31]	71.88
	SE [11]	72.42 (+0.54)
	CBAM [13]	69.33 (−2.55)
	TA [14]	**72.62 (+0.74)**
	LEA (Ours)	72.55 (+0.67)

Bold indicates the highest Top-1 accuracy within each backbone group.

LEA recorded the highest listed ResNet-50 accuracy (78.03%), whereas BAM led on ResNet-18 and TA on MobileNetV2. LEA exceeded the listed baseline on each backbone, indicating that the relative benefit varied with the backbone.

4.2.2. Results on CIFAR-100

The CIFAR-100 comparison used ResNet-20, ResNet-32, ResNet-44 and ResNet-56, together with MobileNetV2 width multipliers of 1.0, 0.75 and 0.5.

Table 3 reports CIFAR-100 accuracy and model costs. The MobileNetV2-1.0 LEA main-table entry is excluded because its accuracy and cost values could not be assigned to one configuration; its labeled sweep is reported in Section 4.3.3.

LEA led the four listed CIFAR-style ResNet comparisons, with baseline differences of 0.68 to 1.56 percentage points. Results differed for narrow MobileNetV2 models. LEA led at width 0.75, whereas CA exceeded LEA at width 0.5 (71.22% versus 70.23%). The ranking therefore depended on the backbone configuration.

Table 3. Comparison results on CIFAR-100.

Backbone	Method	Top-1 (%)	Params (M)	FLOPs (M)
ResNet-20	Baseline	68.60	0.278	41.6
	SE	69.19 (+0.59)	0.281	41.8
	CA	68.57 (−0.03)	0.287	42.1
	TA	68.53 (−0.07)	0.280	42.5
	LEA (Ours)	**69.28 (+0.68)**	0.283	43.2
ResNet-32	Baseline	70.54	0.473	70.4
	SE	71.42 (+0.88)	0.478	70.7
	CA	70.86 (+0.32)	0.488	71.2
	TA	70.50 (−0.04)	0.476	71.9
	LEA (Ours)	**71.57 (+1.03)**	0.481	73.1
ResNet-44	Baseline	71.26	0.667	99.1
	SE	71.98 (+0.72)	0.674	99.6
	CA	72.35 (+1.09)	0.688	100.0
	TA	71.75 (+0.49)	0.671	101.3
	LEA (Ours)	**72.63 (+1.37)**	0.680	103.0
ResNet-56	Baseline	72.05	0.862	127.9
	SE	73.38 (+1.33)	0.871	128.4
	CA	73.13 (+1.08)	0.888	129.5
	TA	72.74 (+0.69)	0.867	130.7
	LEA (Ours)	**73.61 (+1.56)**	0.878	132.8
MobileNetV2-1.0	Baseline	71.99	2.35	94.7
	SE	72.46 (+0.47)	2.38	94.8
	TA	71.63 (−0.36)	2.36	106.1
	CA	73.16 (+1.17)	2.80	99.5
	LEA (Ours)	Withdrawn	–	–
MobileNetV2-0.75	Baseline	70.82	1.48	65.1
	SE	72.16 (+1.34)	1.50	65.2
	TA	71.49 (+0.67)	1.49	74.3
	CA	72.27 (+1.45)	1.74	68.4
	LEA (Ours)	**72.31 (+1.49)**	1.50	66.7
MobileNetV2-0.5	Baseline	69.65	0.82	31.8
	SE	70.49 (+0.84)	0.82	31.8
	TA	70.72 (+1.07)	0.82	37.9
	CA	**71.22 (+1.57)**	0.94	33.6
	LEA (Ours)	70.23 (+0.58)	0.83	32.9

Bold indicates the highest Top-1 accuracy within each backbone group.

4.3. Kernel and Gating Ablations

The ablations examined strip kernel size K, neighborhood size S_K, channel sharing, sigmoid gating and local aggregation. ResNet used channel-wise gates, whereas MobileNetV2 used the conceptual channel-shared design in Section 3.4.

The sweeps examine the effects of strip kernel size, neighborhood size and gate construction. Section 4.5.2 complements them with explicitly defined component controls.

4.3.1. Ablation on ResNet Backbones

Table 4 reports the ImageNet-1K ResNet-18 sweep. The $K = 7, S_K = 3$ score exceeded $K = 5, S_K = 5$ by 0.42 percentage points. We use $K = 7, S_K = 3$ as the fixed setting for the controlled agricultural experiments.

Table 4. Ablation results on ResNet backbones for ImageNet classification.

Backbone	Dataset	Setting	Top-1 (%)	Params (M)	FLOPs (G)
ResNet-18	ImageNet-1K	Baseline	69.76	11.69	1.82
		$K = 5, S_K = 5$	70.68 (+0.92)	11.71	1.84
		$K = 7, S_K = 3$	**71.10 (+1.34)**	11.72	1.84

Bold indicates the highest Top-1 accuracy among the listed settings.

4.3.2. Ablation on MobileNetV2 Backbones

Table 5 examines the channel-shared MobileNetV2 gate, in which a pointwise projection produces one map for all channels. The highest listed accuracy occurred at $K = 3, S_K = 3$.

Table 5. Ablation results on MobileNetV2 for ImageNet classification.

Backbone	Dataset	Setting	Top-1 (%)	Params (M)	FLOPs (G)
MobileNetV2	ImageNet-1K	Baseline	71.88	3.51	0.32
		$K = 3, S_K = 3$	**72.55 (+0.67)**	3.52	0.32
		$K = 3, S_K = 5$	72.01 (+0.13)	3.52	0.32
		$K = 5, S_K = 3$	72.37 (+0.49)	3.52	0.33
		$K = 5, S_K = 5$	72.40 (+0.52)	3.52	0.33
		$K = 7, S_K = 3$	72.54 (+0.66)	3.53	0.33
		$K = 7, S_K = 5$	72.33 (+0.45)	3.53	0.33

Bold indicates the highest Top-1 accuracy among the listed settings.

4.3.3. Ablation on CIFAR-100

Table 6 lists sweeps for ResNet-56 and MobileNetV2-1.0, representing residual and lightweight configurations.

Table 6. Ablation results on CIFAR-100.

Backbone	Setting	Top-1 (%)	Params (M)	FLOPs (M)
ResNet-56	Baseline	72.05	0.862	127.9
	$K = 3, S_K = 3$	73.29 (+1.24)	0.870	130.8
	$K = 3, S_K = 5$	73.31 (+1.26)	0.870	130.8
	$K = 5, S_K = 3$	73.21 (+1.16)	0.874	131.8
	$K = 5, S_K = 5$	73.14 (+1.09)	0.874	131.8
	$K = 7, S_K = 3$	73.12 (+1.07)	0.878	132.8
	$K = 7, S_K = 5$	**73.61 (+1.56)**	0.878	132.8
	w/o Sigmoid	71.74 (−0.31)	0.878	132.8
	w/o LSM	73.14 (+1.09)	0.878	132.0
MobileNetV2-1.0	Baseline	71.99	2.35	94.7
	$K = 3, S_K = 3$	72.77 (+0.78)	2.36	95.5
	$K = 3, S_K = 5$	72.81 (+0.82)	2.36	95.5
	$K = 5, S_K = 3$	72.53 (+0.54)	2.37	96.1
	$K = 5, S_K = 5$	73.21 (+1.22)	2.37	96.1
	$K = 7, S_K = 3$	72.74 (+0.75)	2.37	96.6
	$K = 7, S_K = 5$	**73.29 (+1.30)**	2.37	96.6
	w/o Sigmoid	72.41 (+0.42)	2.37	96.6
	w/o LSM	72.54 (+0.55)	2.37	96.5

Bold indicates the highest Top-1 accuracy within each backbone group.

On ResNet-56, all listed K and S_K combinations exceeded baseline. The highest score was 73.61% for $K = 7, S_K = 5$, compared with 72.05% for baseline. A window of $S_K = 5$ performed better than $S_K = 3$ in most matched settings, consistent with a benefit from broader

local context. Removing sigmoid reduced accuracy below baseline, whereas removing LSM retained a smaller gain than full LEA.

All listed MobileNetV2-1.0 configurations also exceeded baseline. The highest score used $K = 7$, $S_K = 5$, with a 1.30-percentage-point difference. Removing sigmoid or LSM reduced the score relative to full LEA while remaining above baseline.

The labeled MobileNetV2-1.0 sweep gives 73.21% at $K = 5, S_K = 5$ with 96.1 M reported operations, and 73.29% at $K = 7, S_K = 5$ with 96.6 M.

The sweeps show sensitivity to kernel size, window size and channel sharing. MobileNetV2 rows without sigmoid or LSM also use a no-strip configuration, so these rows change more than one operation. The component controls in Section 4.5.2 specify each computation graph explicitly.

4.4. Agricultural Image Classification

The agricultural comparison uses ResNet-50 across Dangerous Farm Insects, PlantVillage, PlantDoc and DeepWeeds.

4.4.1. Classification Results on Agricultural Datasets

Table 7 reports Acc., Macro-F1 and Weighted-F1. Published ResNet-50 results provide context for PlantVillage and DeepWeeds and are labeled as literature references; unreported F1 scores are marked "–".

Table 7. Agricultural classification results with ResNet-50. Literature reference rows are labeled separately.

Dataset	Method	Acc. (%)	Macro-F1 (%)	Weighted-F1 (%)
Dangerous Farm Insects	Baseline	82.28	81.89	82.00
	ABN	81.65	81.53	81.66
	BAM	81.33	81.30	81.38
	CBAM	81.33	81.31	81.36
	ECA	81.96	81.80	81.84
	SE	80.06	79.22	79.28
	LEA (Ours)	**84.18**	**84.19**	**84.07**
PlantVillage	Literature reference [40]	98.98	–	–
	ABN	99.70	**99.58**	99.70
	BAM	99.71	99.56	99.71
	CBAM	**99.75**	99.51	**99.75**
	ECA	99.69	99.44	99.69
	SE	99.73	99.51	99.73
	LEA (Ours)	99.70	99.48	99.70
PlantDoc	Baseline	66.10	67.80	68.37
	ABN	66.10	67.98	68.26
	BAM	65.68	66.68	67.86
	CBAM	65.25	67.97	68.74
	ECA	66.95	67.26	67.91
	SE	**68.22**	**68.75**	69.67
	LEA (Ours)	**68.22**	68.74	**69.75**
DeepWeeds	Literature reference [3]	95.70	–	–
	ABN	96.34	95.21	96.33
	BAM	96.69	95.69	96.69
	CBAM	96.92	96.00	96.91
	ECA	96.92	96.05	96.93
	SE	96.94	95.98	96.93
	LEA (Ours)	**96.97**	**96.17**	**96.97**

Bold indicates the highest value for each metric within each dataset among the in-study methods, including ties.

LEA led the listed insect and DeepWeeds scores, tied with SE in PlantDoc accuracy and trailed CBAM in PlantVillage accuracy. The ranking varied across datasets and metrics.

On Dangerous Farm Insects, LEA recorded 84.18% accuracy against the in-study baseline's 82.28%, a difference of 1.90 percentage points. Macro-F1 and Weighted-F1 differed by 2.30 and 2.07 points, respectively.

LEA and SE tied at 68.22% accuracy on PlantDoc. Their Macro-F1 scores were 68.74% and 68.75%, respectively, while Weighted-F1 scores were 69.75% and 69.67%. Relative to baseline, LEA's 2.12-percentage-point accuracy difference corresponds to about five images on the stated 236-image evaluation set. The two F1 measures give a more differentiated comparison than accuracy alone.

PlantVillage accuracy was 99.75% for CBAM and 99.70% for LEA, placing both near the dataset's accuracy ceiling. LEA therefore offered little additional clean classification benefit in this comparison.

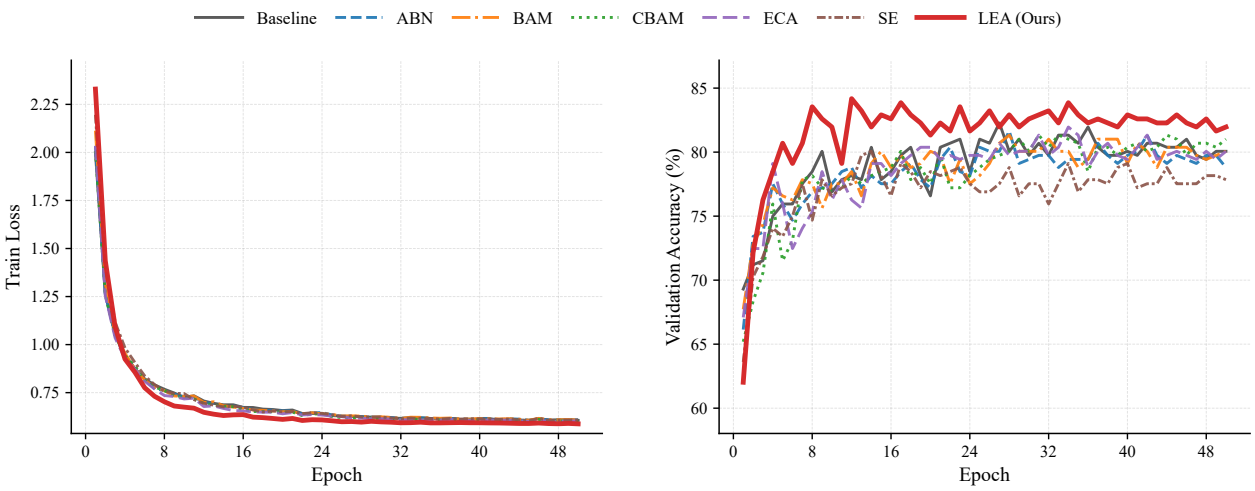
DeepWeeds accuracy was 96.97% for LEA, 96.94% for SE, and 96.92% for ECA and CBAM. The leading methods were closely grouped; LEA's 0.03-point difference from SE corresponds to approximately one image among 3507.

Macro-F1 weights classes equally, whereas Weighted-F1 weights them by support. The PlantVillage class-level analysis in Section 4.5.4 examines how the aggregate results relate to class frequency.

4.4.2. Training Curve Analysis

Training loss and evaluation accuracy curves are shown for Dangerous Farm Insects and PlantDoc. The original plots label the evaluation curves as validation accuracy.

The Dangerous Farm Insects curves show smooth convergence in the illustrated runs (Figure 7). LEA reached lower training loss and maintained higher evaluation accuracy for most epochs than the compared models, consistent with its ranking in Table 7.

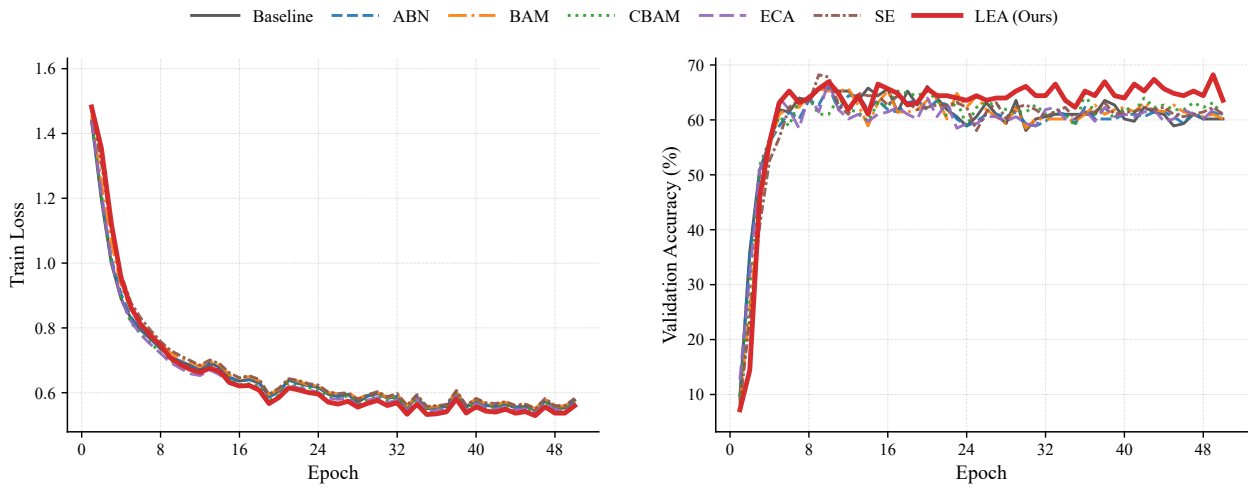


(a) Train loss on Dangerous Farm Insects

(b) Validation accuracy on Dangerous Farm Insects

Figure 7. Training loss and evaluation accuracy curves on Dangerous Farm Insects. (**a**) Training loss. (**b**) Evaluation accuracy.

PlantDoc evaluation accuracy fluctuated more strongly than Dangerous Farm Insects accuracy (Figure 8). Greater variation in backgrounds, illumination and lesion appearance may contribute to this pattern. LEA converged in the displayed run and tied the highest tabled accuracy, with the highest Weighted-F1.



(a) Train loss on PlantDoc (b) Validation accuracy on PlantDoc

Figure 8. Training loss and evaluation accuracy curves on PlantDoc. (**a**) Training loss. (**b**) Evaluation accuracy.

4.4.3. Confusion Matrix Analysis

Figures 9–11 compare baseline and LEA confusion matrices for Dangerous Farm Insects and PlantDoc. Row normalization makes diagonal values into class-wise recall values.The displays use the original class order and values, readable class IDs, and a shared 0–1 color scale.

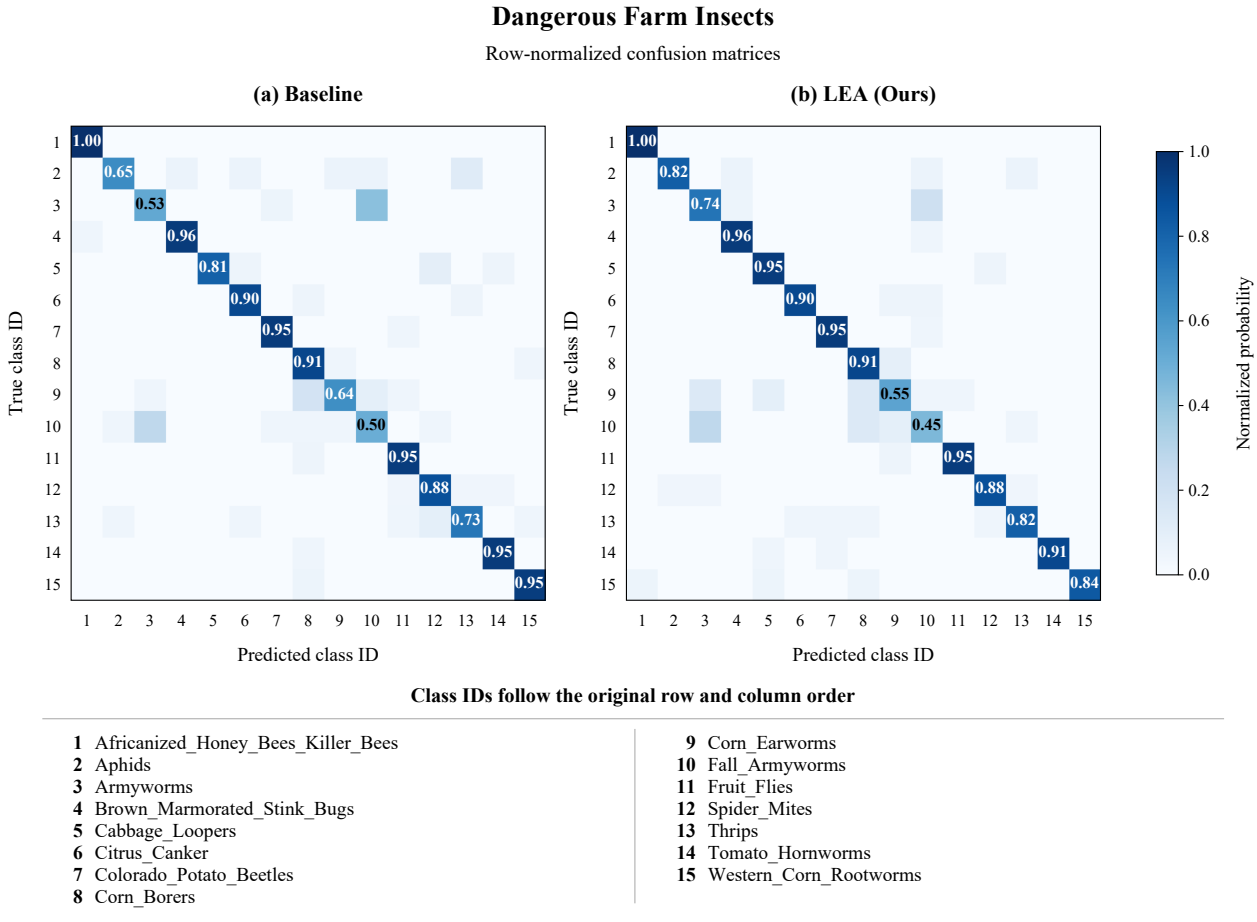


Figure 9. Dangerous Farm Insects confusion matrices. (**a**) Baseline. (**b**) LEA. Class IDs follow the legend, and both panels use a shared 0–1 scale.

LEA shows higher displayed recall for aphids, armyworms, cabbage loopers and thrips in Figure 9. This stronger diagonal pattern agrees with the aggregate classification results.

The PlantDoc matrices show more diffuse class confusion (Figures 10 and 11). LEA shows higher displayed recall for blueberry leaf, cherry leaf, peach leaf and soybean leaf. Improvements also appeared for tomato early blight leaf, tomato septoria leaf spot and tomato leaf late blight.

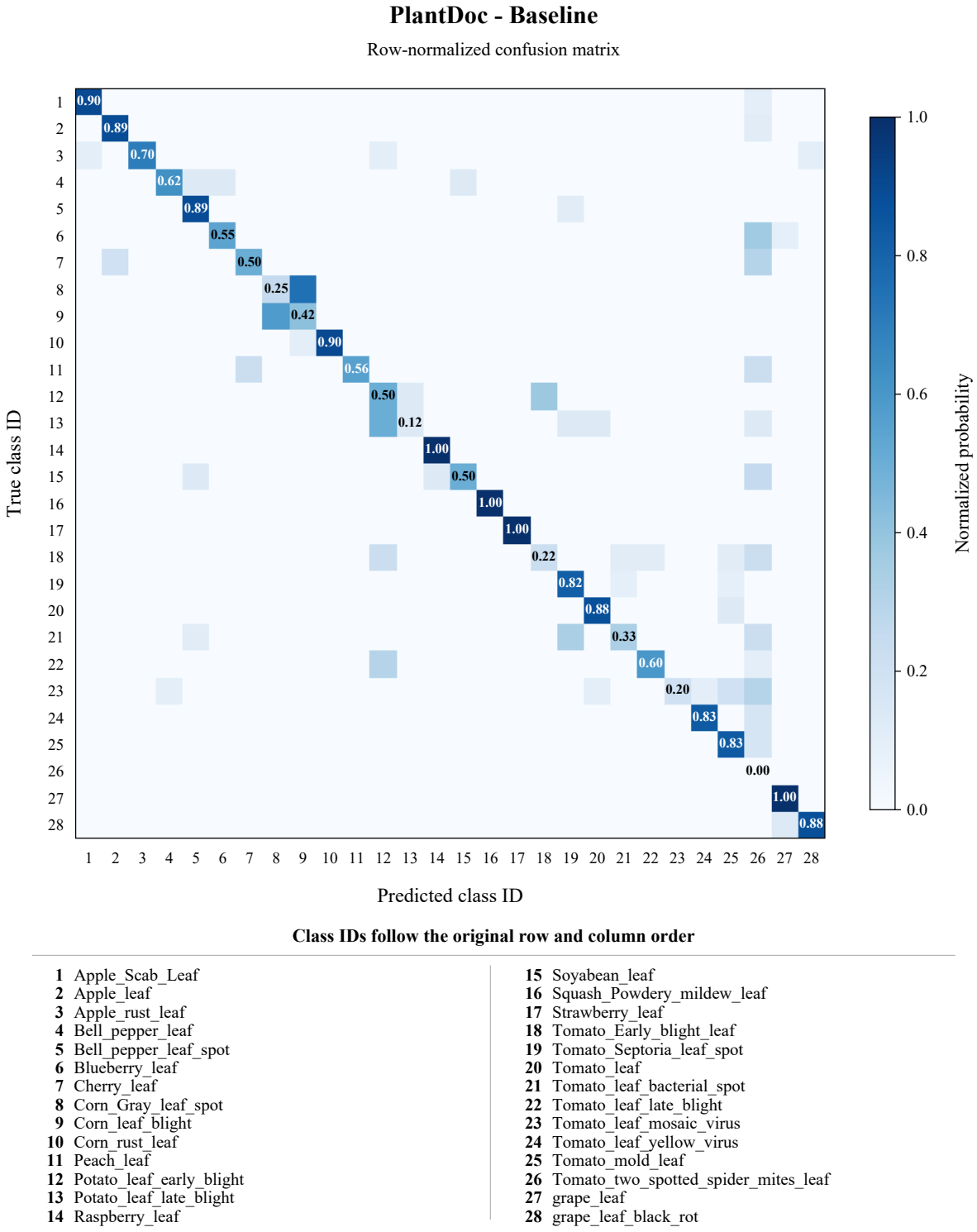
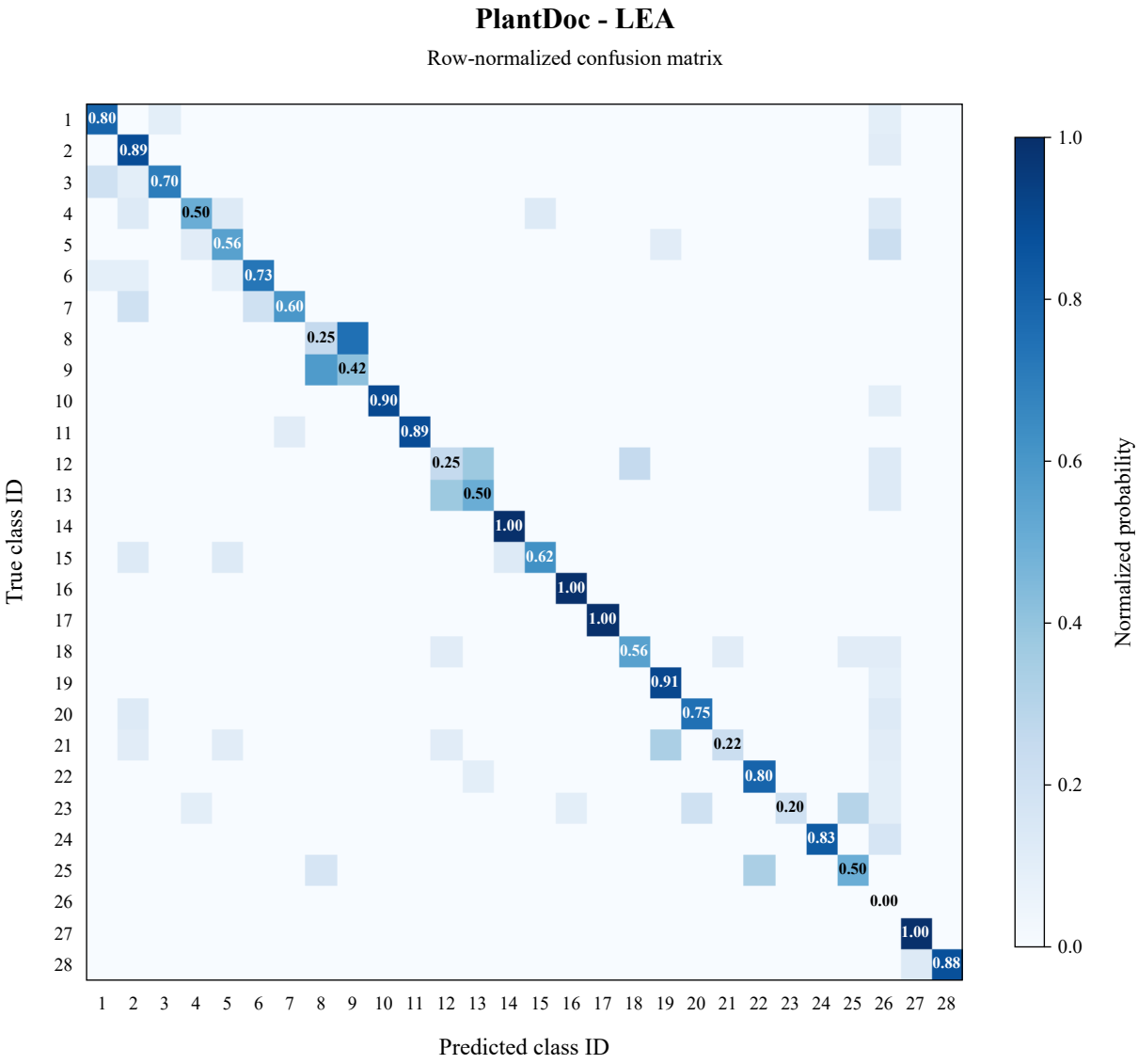


Figure 10. PlantDoc baseline confusion matrix. Numeric class IDs follow the legend; diagonal entries show class-wise recall.



Class IDs follow the original row and column order

1 Apple_Scab_Leaf
2 Apple_leaf
3 Apple_rust_leaf
4 Bell_pepper_leaf
5 Bell_pepper_leaf_spot
6 Blueberry_leaf
7 Cherry_leaf
8 Corn_Gray_leaf_spot
9 Corn_leaf_blight
10 Corn_rust_leaf
11 Peach_leaf
12 Potato_leaf_early_blight
13 Potato_leaf_late_blight
14 Raspberry_leaf
15 Soyabean_leaf
16 Squash_Powdery_mildew_leaf
17 Strawberry_leaf
18 Tomato_Early_blight_leaf
19 Tomato_Septoria_leaf_spot
20 Tomato_leaf
21 Tomato_leaf_bacterial_spot
22 Tomato_leaf_late_blight
23 Tomato_leaf_mosaic_virus
24 Tomato_leaf_yellow_virus
25 Tomato_mold_leaf
26 Tomato_two_spotted_spider_mites_leaf
27 grape_leaf
28 grape_leaf_black_rot

Figure 11. PlantDoc LEA confusion matrix. Class IDs and color scale match the baseline panel.

Both models show 0.00 recall for tomato two-spotted spider mites leaf at the displayed precision. This category warrants targeted error analysis using class support and per-image predictions. Limited examples, similar symptoms and background variation are plausible factors to examine.

The confusion matrices identify the classes that benefit most from LEA and those requiring further investigation, complementing the aggregate F1 scores.

4.4.4. Gradient-Weighted Class Activation Mapping (Grad-CAM) Visualization Analysis

Grad-CAM [41] was applied to three selected images from different categories in each agricultural dataset. Dangerous Farm Insects and PlantDoc compare LEA with in-study baselines, while PlantVillage and DeepWeeds compare available attention-enhanced models selected by the accuracy ranking in Table 7.

In the selected insect images, LEA produced more compact activations than baseline, ECA and ABN (Figure 12). Responses were more concentrated on insect bodies in the thrips and corn earworm examples and on the target in the corn borer example.

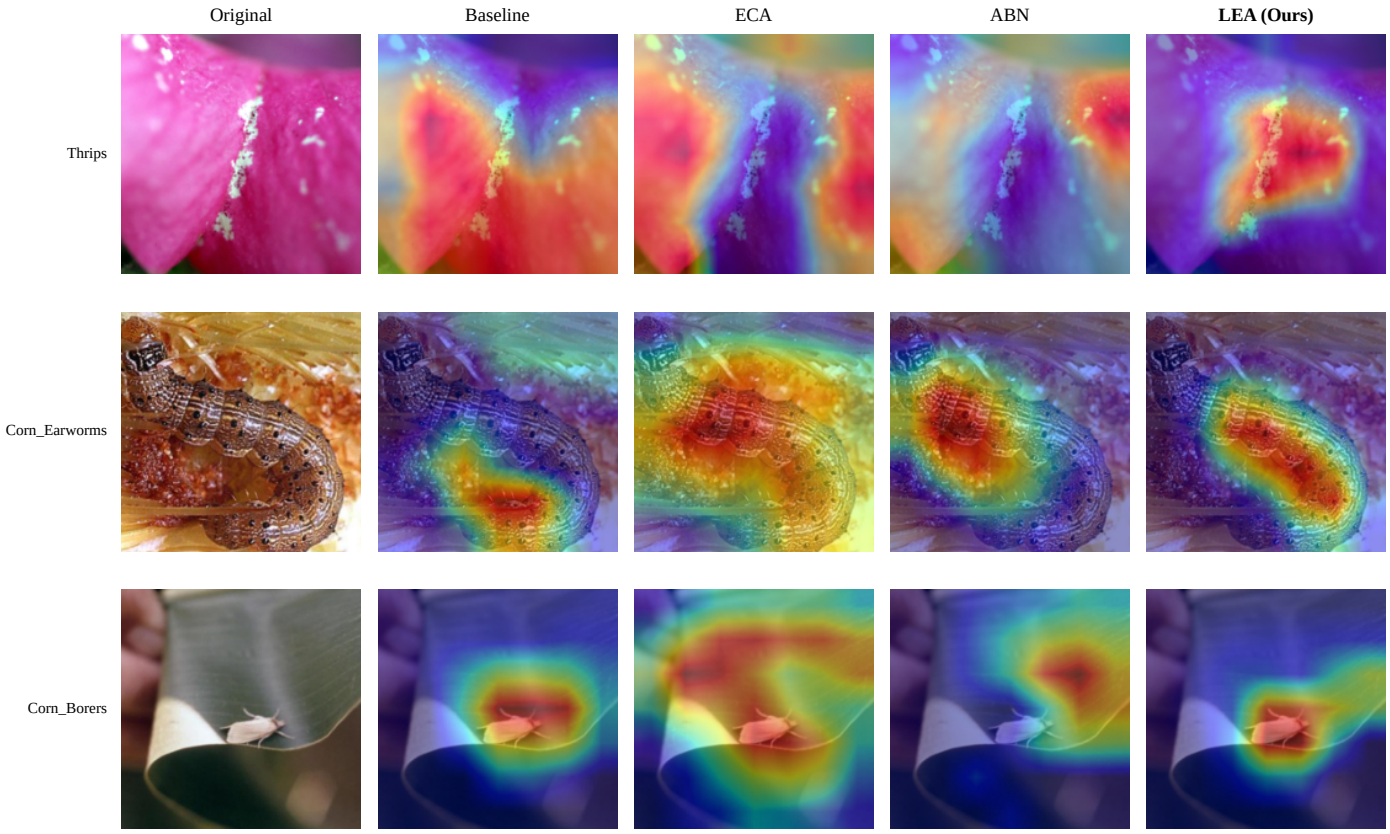


Figure 12. Grad-CAM visualization on Dangerous Farm Insects. The compared methods include the ResNet-50 baseline, ECA, ABN, and LEA. Warmer colors indicate higher Grad-CAM responses, whereas cooler colors indicate lower responses.

LEA also produced more compact leaf region activations in the selected PlantDoc images (Figure 13). Soybean leaf and peach leaf showed less background response than baseline, SE and ECA. Tomato leaf late blight showed a more concentrated response on the diseased leaf.

The DeepWeeds examples show more concentrated LEA responses on snake weed, siam weed and lantana than SE and ECA (Figure 14). Responses to surrounding vegetation were reduced in the first two examples.

LEA produced more compact leaf area activations than CBAM and SE in the selected PlantVillage images (Figure 15).

The visualizations illustrate where the classifiers place their responses. Section 4.5.6 quantifies this behavior using lesion annotations, while Section 4.5.5 examines performance under controlled image perturbations.

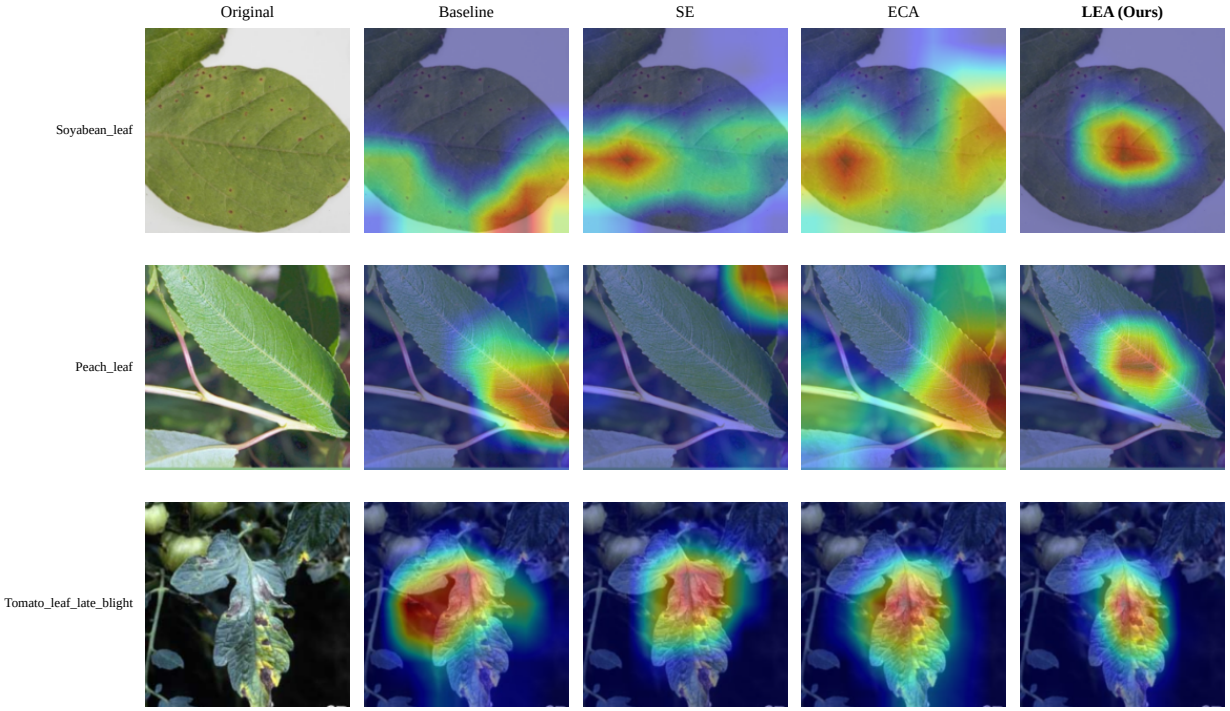


Figure 13. Grad-CAM visualization on PlantDoc. The compared methods include the ResNet-50 baseline, SE, ECA, and LEA. Warmer colors indicate higher Grad-CAM responses, whereas cooler colors indicate lower responses.

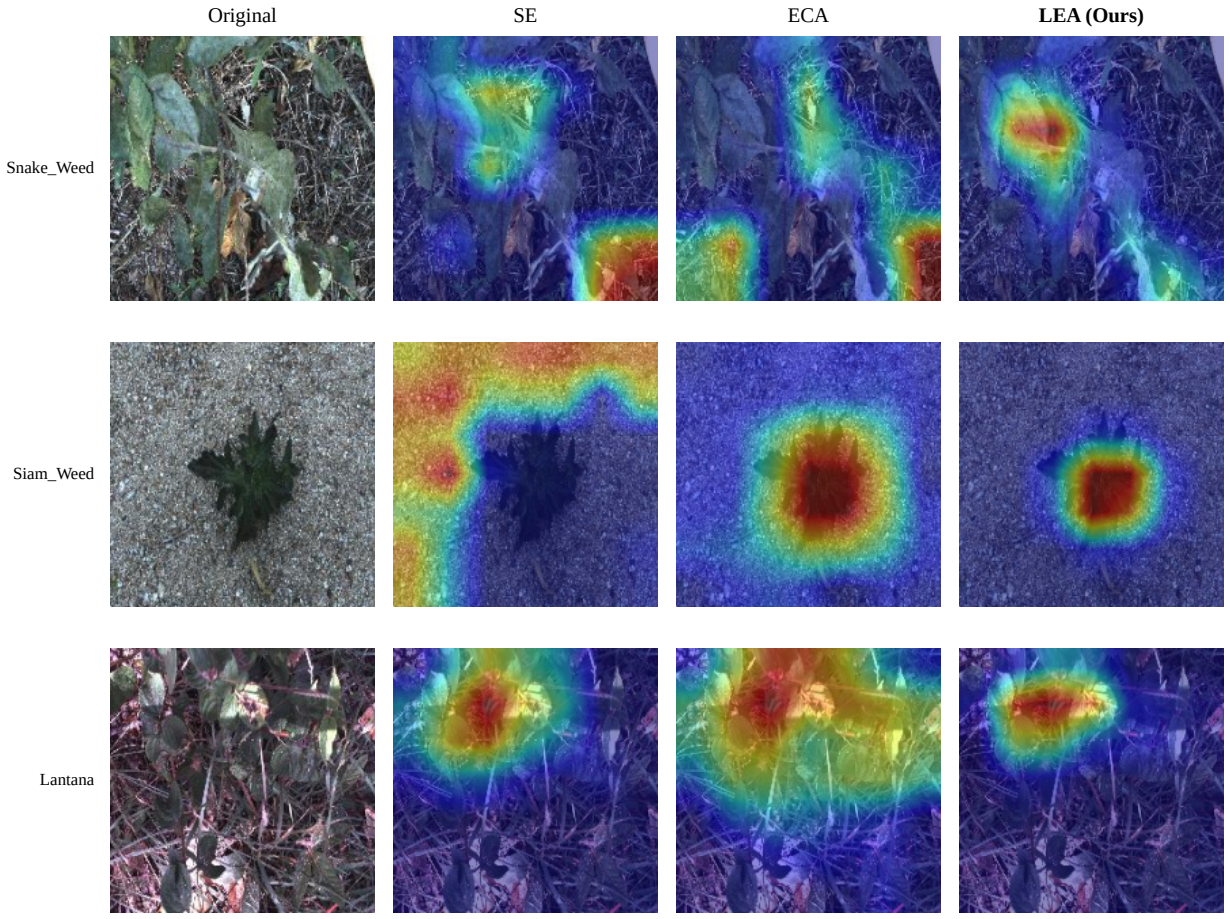


Figure 14. Grad-CAM visualization on DeepWeeds. The compared methods include SE, ECA, and LEA. Warmer colors indicate higher Grad-CAM responses, whereas cooler colors indicate lower responses.

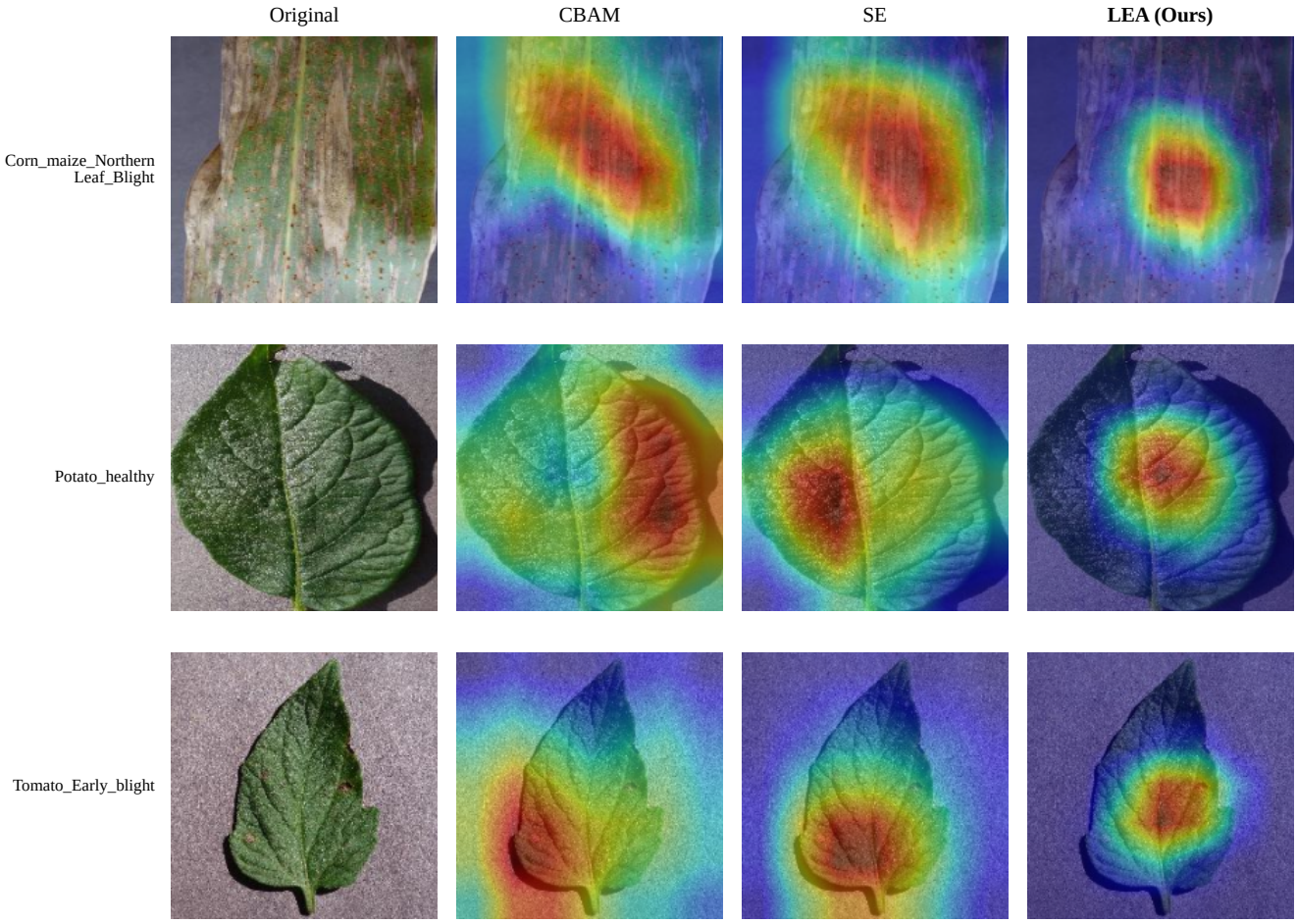


Figure 15. Grad-CAM visualization on PlantVillage. The compared methods include CBAM, SE, and LEA. Warmer colors indicate higher Grad-CAM responses, whereas cooler colors indicate lower responses.

4.5. Controlled Agricultural Experiments

4.5.1. Datasets and Training Protocol

We evaluate ResNet-50 on PlantVillage and PlantSeg using the fixed protocols below. Table 8 summarizes the PlantVillage split used in these experiments.

Table 8. New PlantVillage manifest used for all supplementary training seeds. Counts are images; exact-pixel groups are kept within one split.

Dataset/Backbone	Classes	Train	Validation	Test
PlantVillage/ResNet-50	38	43471	5416	5418

PlantVillage contains 54,305 supported images and 21 exact-pixel duplicate groups, ten spanning the original train/validation folders. Split seed 42 produces an approximately 80/10/10 stratified partition, with each exact-pixel group assigned wholly to one split. The resulting train/validation/test counts are 43,471/5416/5418, with no exact-pixel group crossing splits. Manifests, SHA256 digests, class mappings and class support are retained. A dHash screening at distance at most four flagged 92 cross-split near-duplicate candidate pairs for further inspection.

ResNet-50 uses ImageNet-1K V2 initialization (`resnet50-11ad3fa6.pth`). Its 1000-class head is replaced by a newly initialized 38-class PlantVillage or 115-class PlantSeg head. In-

serted attention follows its declared seeded initialization. Backbone, attention and classifier parameters are trainable from the first epoch. Backbone BatchNorm remains in training mode during optimization; validation and testing use evaluation mode. No frozen-stage schedule, oversampling or class weighting is used.

All variants use seeds 42, 43 and 44; 50 epochs; a batch size of 32; and AdamW with nominal learning rate 3×10^{-4}. Weight decay is 0.05, and the learning rate uses a five-epoch warm-up followed by cosine decay to a final factor of 0.01. Training uses CUDA mixed precision; validation and testing use float32 forward passes. The RTX 4090 environment uses Python 3.8.10, PyTorch 1.11.0+cu113, torchvision 0.12.0+cu113 and cuDNN 8.2. Matmul TF32 is disabled, while cuDNN TF32 is enabled. Four data-loader workers and four CPU threads are used. Shared epoch-derived seeds align data order and augmentation randomness across methods.

Training preprocessing comprises a random resized crop to 224×224 (scale 0.65–1.00, aspect ratio 0.75–1.3333), horizontal/vertical flips (probabilities 0.5/0.2), rotation within 15 degrees, color jitter (brightness/contrast/saturation 0.2, hue 0.03), Gaussian blur (probability 0.10, sigma 0.1–1.0), and tensor Gaussian noise (standard deviation 0.01, probability 0.15). ImageNet normalization is followed by random erasing (probability 0.1, area 0.02–0.12, aspect ratio 0.3–3.3). Mixup uses $\alpha = 0.1$ and unweighted soft-target cross-entropy. Evaluation resizes the shorter side to 256, center-crops 224, and applies the same normalization. Corruptions are applied to decoded RGB images before evaluation resizing/cropping.

The checkpoint with highest validation accuracy is selected, with ties resolved in favor of the first epoch. This checkpoint is used for clean and corruption tests. LEA uses fixed $K = 7$ and $S_K = 3$. The PlantVillage comparison includes seven configurations with three completed runs each. Recalculation from exported predictions reproduced 21 clean and 189 corrupted metric records.

4.5.2. Component Controls and Repeated-Run Results

The component tables report seven configurations with completed evaluations. Standalone LSM runs did not produce valid test results and are omitted.

Table 9 defines the computation graph of each configuration.

Table 9. Exact computation graphs of the seven reported supplementary configurations. $Z = \mathrm{SCM}(X)$ and $A = \sigma(Z)$. All settings use $K = 7$ and $S_K = 3$.

Variant	Output
Baseline	X
SCM only	$X \odot Z$
SCM + sigmoid	$X \odot A$
SCM + uniform	$X \odot \mathrm{AvgPool}_{3\times3}(A)$
Original LEA	$X \odot \mathcal{L}_{\mathrm{orig}}(A)$
Formula valid	$X \odot \mathcal{L}_{\mathrm{valid}}(A)$
Formula zero	$X \odot \mathcal{L}_{\mathrm{zero}}(A)$

Uniform averaging includes padded zero gates in its divisor. The positive stabilizers and boundary rules are specified in Section 3.3.

Original LEA reached 99.902 ± 0.021% accuracy on PlantVillage, against 99.883 ± 0.021% for baseline (Table 10). The 0.01846-percentage-point difference equals one additional correct prediction among 5418 images in each paired seed. The gain is small in absolute terms because baseline accuracy is already near ceiling.

Table 10. PlantVillage ResNet-50 supplementary results for the seven configurations with completed evaluations. Percentages are mean ± sample SD over seeds 42, 43 and 44; Δ is the mean accuracy difference from the matched baseline in percentage points.

Method	Accuracy	Macro-F1	Weighted-F1	Δ	Params (M)	GMACs
Baseline	99.883 ± 0.021	99.811 ± 0.044	99.883 ± 0.021	+0.000	23.5859	4.0872
Original LEA	99.902 ± 0.021	99.844 ± 0.052	99.901 ± 0.021	+0.018	23.8276	4.1645
SCM only	99.791 ± 0.043	99.711 ± 0.072	99.791 ± 0.042	−0.092	23.8276	4.1645
SCM + sigmoid	99.865 ± 0.046	99.773 ± 0.062	99.865 ± 0.046	−0.018	23.8276	4.1645
SCM + uniform	99.895 ± 0.011	99.846 ± 0.002	99.896 ± 0.011	+0.012	23.8276	4.1645
Formula valid	99.865 ± 0.056	99.787 ± 0.117	99.865 ± 0.056	−0.018	23.8276	4.1645
Formula zero	99.902 ± 0.056	99.858 ± 0.077	99.901 ± 0.057	+0.018	23.8276	4.1645

Parameters and Conv/Linear MACs use the 38-class head and 224-pixel input. MACs exclude normalization, pooling and element-wise operations; they are not full FLOPs. The grouped data partition is specified in Table 8.

Formula-zero matched LEA's displayed mean accuracy but had a larger SD of 0.056; its 99.963% seed-44 result was the highest individual score. SCM with uniform aggregation and formula-zero had a slightly higher mean Macro-F1 than original LEA, showing that the aggregation rule affects accuracy and class-balanced performance differently.

Table 11 reports paired seed differences and 95% Student-t intervals, with $t_{0.975,2} = 4.30265$. The accuracy interval is omitted because the paired accuracy differences have zero sample SD.

Table 11. Original LEA minus baseline under paired training seeds. Intervals summarize paired training-seed differences on the fixed test set.

Dataset	Metric	Δ (pp)	Sample SD	Descriptive 95% Interval
PlantVillage	Accuracy	+0.01846	0.000	Not reported *
PlantVillage	Macro-F1	+0.03283	0.03985	[−0.06617, 0.13183]

* Each seed gives one additional correct test image. The accuracy interval is omitted because paired differences have zero sample SD.

Table 10 also reports parameters and Conv/Linear MACs for each configuration.

4.5.3. Measured GPU Cost and Counting Conventions

The architecture benchmark used an RTX 4090, 224 × 224 inputs and a common 1000-class head. Randomly initialized dense models ran in eager float32 evaluation and inference mode. A total of 5 warm-up passes preceded 20 timed forward passes with CUDA synchronization. We report batch-one latency, batch-four throughput and peak allocated memory. Throughput is batch size divided by mean batch time. Peak memory includes the model, input, and live intermediates. Timing covers GPU forward inference, with SD calculated across passes in one process; disk access, decoding, preprocessing and host-to-device transfer are excluded.

A CPU audit counted 25,557,032/25,557,118 baseline/ECA parameters for 1000-class ResNet-50. The corresponding ResNet-18 counts were 11,689,512/11,689,548, giving increments of 86/36 parameters for ResNet-50/ResNet-18. The audit script and source hash accompany the revision.

Table 12 reports the architecture measurements under this common-head protocol.

Original LEA added 241,664 parameters to ResNet-50 with the same classifier. Batch-one latency rose from 3.805 to 6.345 ms, and batch-four peak allocated memory from 140.0 to 307.8 MiB. Pooling avoids an unfolded neighborhood tensor, but aggregation, normalization and element-wise operations still incur computation and memory access.

Conv/Linear MACs omit normalization, pooling, exponentials, sigmoid, division, element-wise operations and data movement. One MAC equals two arithmetic FLOPs only within this counted subset; the reported MACs are not complete FLOPs.

Table 12. RTX 4090 architecture measurements with a common 1000-class head. Latency uses batch one; throughput and peak allocated memory use batch four. MACs count Conv/Linear operations only.

Net	Method	Params	GMACs	Latency (ms)	Images/s	MiB
R50	Baseline	25.5570	4.0892	3.805 ± 0.172	939.6	140.0
R50	ECA	25.5571	4.0893	4.933 ± 0.031	722.4	140.1
R50	Original LEA	25.7987	4.1665	6.345 ± 0.024	573.9	307.8
R50	Formula zero	25.7987	4.1665	7.648 ± 0.049	463.7	226.7

Params are millions. Eager float32 is forward only;there are 5 warm-ups and 20 timed passes with CUDA synchronization; SD is across passes. Inputs are already on GPU. This table is separate from task-head training counts and excludes full-operator FLOPs.

LEA combines low parameter overhead with a measurable inference cost. Its use is most attractive when spatial attribution is valuable and the hardware budget accommodates the additional latency and memory.

4.5.4. Class Support and Failure Analysis

The supplementary table reports split support and baseline/LEA mean recall and F1 for all 38 PlantVillage classes. Training support ranged from 122 to 4407, a ratio of approximately 36.1, without class balancing. The three smallest training classes were selected independently of test performance. They were healthy potato (122 training, 15 test), apple cedar rust (221, 27) and healthy peach (288, 36). Their baseline-to-LEA mean F1 values were 100.000 → 100.000, 100.000 → 100.000 and 99.531 → 99.061%, respectively. Across all classes, F1 rose in 8, fell in 5 and remained unchanged in 25. The class-level changes were therefore mixed rather than concentrated in the smallest classes.

Class support and per-image predictions would enable the same analysis for the PlantDoc category highlighted in Section 4.4.3.

4.5.5. Controlled Corruption Tests

All successful checkpoints were tested with identical deterministic perturbations. Gaussian-noise SDs were 0.02, 0.05 and 0.10 on RGB values in $[0, 1]$, followed by clipping. Brightness multipliers were 0.8, 0.6 and 0.4. A gray occluder spanned 10%, 20% or 30% of each image dimension, covering approximately 1%, 4% or 9% before cropping. Relative-path hashes fixed noise and occluder placement across methods. Perturbations are applied before the evaluation crop, so the visible occluded area depends on cropping.

Table 13 reports the highest severity, with all levels retained in the supplementary CSV. At noise SD 0.10, baseline and LEA accuracy were 95.755% and 92.746%, with clean-relative drops of 4.128 and 7.155 percentage points. At strongest occlusion, both reached 99.754% mean accuracy, while LEA had slightly lower Macro-F1. Milder occlusion favored LEA slightly. The contrast between noise sensitivity and occlusion performance points to condition-specific robustness.

Table 13. Highest-severity corruption results. Mean ± sample SD over three training seeds; final column is clean minus corrupted accuracy in percentage points.

Dataset	Condition	Method	Accuracy	Macro-F1	Drop
PlantVillage	Noise 0.10	Baseline	95.755 ± 1.856	94.008 ± 2.184	4.128
PlantVillage	Noise 0.10	Original LEA	92.746 ± 1.071	90.748 ± 1.151	7.155
PlantVillage	Dimming 0.4	Baseline	99.883 ± 0.043	99.813 ± 0.061	0.000
PlantVillage	Dimming 0.4	Original LEA	99.871 ± 0.049	99.809 ± 0.066	0.031
PlantVillage	Occlusion 0.3	Baseline	99.754 ± 0.053	99.655 ± 0.068	0.129
PlantVillage	Occlusion 0.3	Original LEA	99.754 ± 0.059	99.607 ± 0.079	0.148

Occlusion 0.3 denotes 30% of each image dimension, i.e., approximately 9% area before cropping. Full results at all severities and Macro-F1 drops are provided in the supplementary CSV. No additional training is used for these evaluations.

4.5.6. PlantSeg Seven-Method Comparison and Mask-Based Attribution

PlantSeg supplies in-the-wild disease images and lesion masks [42]. The downloaded v7 archive contains 7774 images in 115 categories. Its official split comprises 5367 training, 846 validation and 1561 test images. Classification and true-class Grad-CAM were evaluated. Binary CAM IoU measures thresholded attribution overlap rather than supervised segmentation performance.

All seven methods use the official PlantSeg v7 partition. An audit found exact train–test matches involving 80 of 1561 test images (5.125%); 91 test images matched training or validation. We therefore use this partition for a within-protocol comparison of classification and lesion attribution. The supplementary audit lists matching images, hashing rules and mask checks.

Baseline, SE, ECA, CBAM, original LEA, ELA-B and SCSA shared ResNet-50 V2 initialization, full fine-tuning and a 115-class head. Each used seeds 42/43/44, 50 epochs and validation-accuracy checkpoint selection, with the optimizer, augmentation and input recipe above. Attention followed bn3 and preceded residual addition in all sixteen bottlenecks. SE used reduction 16, and ECA the supplied adaptive kernel. CBAM used reduction 16 and spatial kernel 7 without spatial BatchNorm. ELA used ELA-B [43]. SCSA followed its pinned author implementation and defaults [23], including one attention head and a fixed 7 × 7 pooling window. The comparison applies the same nominal training recipe to all modules.

Tables 14 and 15 report all six metric means and sample SDs from the same 21 per-seed records. All methods share a data-manifest hash; source specifications and runner identities are provided in the supplementary records.

The saved evaluator uses the true-class logit at layer4, ReLU, bilinear CAM upsampling and spatial-maximum normalization. Images are resized to 256 and center-cropped to 224. Masks undergo the same crop with nearest-neighbor resizing, and non-zero values define foreground under the official full policy. The binary threshold is fixed at 0.5 without test tuning; masks are not used to train the classifier. Each run declares 1558 eligible images and 3 empty visible masks among 1561 tests. Zero CAMs receive zero hit, energy and IoU on eligible masks, with counts ranging from 3 to 14 across runs. The true-class setup measures attribution with the target class supplied.

LEA exceeded ECA, ELA-B and SCSA in mean classification accuracy but ranked below baseline, SE and CBAM. Its classification benefit was therefore comparator dependent.

LEA recorded the highest mean lesion energy, 41.612 ± 0.281%, among the seven methods. It exceeded baseline by 0.819 percentage points and SE, ECA, CBAM, ELA-B and SCSA by 0.842, 0.933, 1.023, 1.042 and 1.040 points, respectively. LEA led every comparator

in seeds 42, 43 and 44, covering all 18 seed-matched comparisons. A larger fraction of CAM energy therefore fell within annotated lesions.

Table 14. PlantSeg official-full ResNet-50 classification showing mean ± sample SD over seeds 42/43/44, recalculated from the retained per-seed records. All values are percentages; bold indicates the highest mean in each column.

Method	Accuracy	Macro-F1	Weighted-F1
Baseline	**71.535** ± 0.764	**63.974** ± 1.051	**70.887** ± 0.890
SE	71.514 ± 0.303	63.072 ± 0.350	70.669 ± 0.428
CBAM	71.215 ± 1.206	62.897 ± 2.685	70.311 ± 1.478
Original LEA	70.767 ± 0.243	62.550 ± 0.338	69.949 ± 0.301
ECA	70.211 ± 1.342	61.658 ± 1.721	69.255 ± 1.565
ELA-B	68.161 ± 0.618	59.705 ± 1.270	67.376 ± 0.666
SCSA	66.197 ± 0.098	57.010 ± 0.907	65.146 ± 0.120

All methods use the same 115-class partition and nominal training recipe.

Table 15. PlantSeg true-class Grad-CAM diagnostic. Mean ± sample SD in percent; binary CAM IoU is not segmentation mIoU. Bold indicates the highest mean in each column, including ties.

Method	Pointing Accuracy	Lesion Energy	Binary CAM IoU
Baseline	51.776 ± 1.670	40.792 ± 0.470	18.949 ± 0.263
SE	**52.054** ± 1.702	40.769 ± 0.412	18.659 ± 0.519
CBAM	50.513 ± 1.136	40.589 ± 0.375	18.064 ± 0.478
Original LEA	**52.054** ± 1.161	**41.612** ± 0.281	17.772 ± 0.111
ECA	51.498 ± 0.408	40.679 ± 0.192	**18.960** ± 0.492
ELA-B	51.562 ± 0.499	40.569 ± 0.085	18.575 ± 0.186
SCSA	49.807 ± 0.421	40.572 ± 0.017	18.277 ± 0.271

Each run declares 1558 eligible images and three empty visible masks among 1561 tests. Zero CAMs remain in the eligible denominator.

LEA tied with SE in mean pointing accuracy (52.054%), above baseline (51.776%), although it did not lead SE in every seed. This metric records whether the activation peak falls within a lesion. LEA exceeded ELA-B and SCSA in accuracy, Macro-F1 and Weighted-F1 in each seed. Its mean accuracy differences were 2.605 and 4.570 percentage points, and Macro-F1 differences were 2.845 and 5.540 points, respectively. Baseline, SE and CBAM nevertheless retained higher classification means. LEA also had lower binary CAM IoU than baseline (17.772% versus 18.949%). Greater energy concentration can coexist with lower thresholded overlap if responses cover only part of a lesion. The metric pattern is consistent with responses concentrating on discriminative lesion regions rather than covering their full extent.

5. Discussion

5.1. Practical Relevance

LEA is relevant to agricultural classification tasks driven by localized cues, such as leaf lesions, insect markings or weed leaf margins. Its PlantSeg results show stronger concentration of attribution energy within annotated lesions. A practical workflow could combine class predictions with Grad-CAM maps to inspect the tissue or morphology driving a decision; application to insect and weed structures is a natural direction for evaluation.

Offline screening and decision support are potential uses when a small parameter increment is desirable and additional inference time is acceptable. The value of LEA depends on the task: clean accuracy changed little on PlantVillage, whereas the PlantSeg attribution

metrics revealed clearer spatial differences. The noise and occlusion tests further suggest that input quality should be considered when selecting a deployment setting.

5.2. Scope and Future Work

The experiments characterize LEA as a local gate-aggregation module, with the clearest gains in lesion-energy concentration and a small clean classification gain on PlantVillage. Broader evaluation on independent field images would clarify how these effects transfer across crops, capture conditions and devices. Useful next steps include resolving the flagged near-duplicate candidates, stabilizing standalone LSM, and comparing complete agricultural systems under matched conditions. Target-device measurements and initialization controls would further guide practical configuration choices.

6. Conclusions

LEA combines strip filtering, normalized bounded gating and neighborhood aggregation in an insertable spatial recalibration module. It preserves spatial resolution with few added parameters. The experiments show a small PlantVillage accuracy gain and stronger lesion-energy concentration on PlantSeg, alongside metric-dependent classification and corruption behavior. Its low parameter overhead is accompanied by additional latency and memory use.

Future work should focus on independent agricultural field images, matched comparisons and target device optimization. More complete component and lesion localization studies can help identify where neighborhood gate aggregation provides the greatest practical benefit.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/app162111085/s1. The revision package includes per-seed metrics, three-seed summaries, all-severity corruption summaries, class support and baseline/LEA comparisons, run-status records, dataset manifests, source/hash records and experiment scripts. PlantVillage summaries are checked against exported predictions; PlantSeg tables use the original 21 per-seed records and are accompanied by the data quality audit. These records accompany the revision.

Author Contributions: Conceptualization, K.L. and Y.W.; methodology, K.L.; software, K.L.; validation, K.L., Z.L., X.H. and Y.W.; formal analysis, K.L.; investigation, K.L.; resources, Y.W.; data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, Z.L., X.H. and Y.W.; visualization, K.L.; supervision, Y.W.; project administration, Y.W.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript

Funding: This research was funded by the Key Technologies R&D Program of Jiangsu (Prospective and Key Technologies for Industry), grant number BE2023005-2.

Data Availability Statement: The ImageNet-1K dataset can be downloaded from the website https://www.kaggle.com/competitions/imagenet-object-localization-challenge/data (accessed on 15 January 2025). The CIFAR-100 dataset can be downloaded from the website https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 15 January 2025). The Dangerous Farm Insects dataset is available at https://www.kaggle.com/datasets/tarundalal/dangerous-insects-dataset (accessed on 15 January 2025). The PlantVillage dataset is available at https://github.com/spMohanty/PlantVillage-Dataset (accessed on 15 January 2025). The PlantDoc dataset is available at https://github.com/pratikkayal/PlantDoc-Dataset (accessed on 15 January 2025). The DeepWeeds dataset is available at https://github.com/AlexOlsen/DeepWeeds (accessed on 15 January 2025). PlantSeg v7 is available at https://doi.org/10.5281/zenodo.17719108 (accessed on 1 November 2026). The revision package contains available split manifests, source/hash records, experiment scripts and result exports, including seven PlantSeg summaries and 21 per-seed rows. Some earlier split manifests, model checkpoints and PlantSeg per-image outputs were not retained.

Acknowledgments: The authors acknowledge the providers of the public datasets used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LEA	Local Enhanced Attention
CNNs	Convolutional neural networks
SCM	Separable Convolution Module
LSM	Local Softmax Modulation
Grad-CAM	Gradient-weighted Class Activation Mapping
FAO	Food and Agriculture Organization
SE	Squeeze-and-Excitation
ECA	Efficient Channel Attention
CBAM	Convolutional Block Attention Module
TA	Triplet Attention
CA	Coordinate Attention
EMA	Efficient Multi-Scale Attention
PCSA	Pyramid Channel and Spatial Attention
DWConv	Depthwise convolution
PWConv	Pointwise convolution
GN	Group normalization
BN	Batch normalization
IN	Instance normalization
FLOPs	Floating-point operations
Acc.	Accuracy
BAM	Bottleneck Attention Module
EMCA	Efficient Multiscale Channel Attention Module
GCNet	Global Context Network
SRM	Style-Based Recalibration Module
GSoP-Net	Global Second-Order Pooling Network
A2-Nets	Double Attention Networks
GALA	Global-and-Local Attention
CAM	Class activation map
SD	Standard deviation
MACs	Multiply-accumulate operations
ELA	Efficient Local Attention
SCSA	Spatial and Channel Synergistic Attention
ABN	Attention Branch Network

References

1. Food and Agriculture Organization of the United Nations. *Scientific Review of the Impact of Climate Change on Plant Pests: A Global Challenge to Prevent and Mitigate Plant Pest Risks in Agriculture, Forestry and Ecosystems*; FAO: Rome, Italy, 2021. [CrossRef]
2. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A Dataset for Visual Plant Disease Detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 5–7 January 2020*; ACM: New York, NY, USA, 2020; pp. 249–253. [CrossRef]
3. Olsen, A.; Konovalov, D.A.; Philippa, B.; Ridd, P.; Wood, J.C.; Johns, J.; Banks, W.; Girgenti, B.; Kenny, O.; Whinney, J.; et al. DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning. *Sci. Rep.* **2019**, *9*, 2058. [CrossRef] [PubMed]
4. Zhao, C.Q.; Sun, F.L. FAF-Net: A Lightweight Frequency-Auxiliary Fusion Network for Plant Disease Classification in Natural-Scene Images. *Appl. Sci.* **2026**, *16*, 6380. [CrossRef]
5. Camargo, A.; Smith, J.S. Image Pattern Classification for the Identification of Disease Causing Agents in Plants. *Comput. Electron. Agric.* **2009**, *66*, 121–125. [CrossRef]
6. Mokhtar, U.; El Bendary, N.; Hassenian, A.E.; Emary, E.; Mahmoud, M.A.; Hefny, H.; Tolba, M.F. SVM-Based Detection of Tomato Leaves Diseases. In *Intelligent Systems'2014*; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2015; Volume 323, pp. 641–652. [CrossRef]
7. Pantazi, X.E.; Moshou, D.; Tamouridou, A.A. Automated Leaf Disease Detection in Different Crop Species through Image Features Analysis and One Class Classifiers. *Comput. Electron. Agric.* **2019**, *156*, 96–104. [CrossRef]
8. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef] [PubMed]

9. Utku, A.; Kaya, M.; Canbay, Y. A New Hybrid ConvViT Model for Dangerous Farm Insect Detection. *Appl. Sci.* **2025**, *15*, 2518. [CrossRef]
10. Baek, E.-T. Attention Score-Based Multi-Vision Transformer Technique for Plant Disease Classification. *Sensors* **2025**, *25*, 270. [CrossRef] [PubMed]
11. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. [CrossRef]
12. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. 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. [CrossRef]
13. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19. [CrossRef]
14. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to Attend: Convolutional Triplet Attention Module. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 3139–3148. [CrossRef]
15. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722. [CrossRef]
16. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In *Proceedings of ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5. [CrossRef]
17. Maurya, R.; Rajput, L.; Mahapatra, S. RAI-Net: Tomato Plant Disease Classification Using Residual-Attention-Inception Network. *IEEE Access* **2025**, *13*, 64832–64840. [CrossRef]
18. Babu, T.; Nair, R.R.; Balusamy, B.; Khoh, W.H.; Nair, J. AgriFewNet—A Lightweight RGB Few-Shot Learning Model for Efficient Plant Disease Classification. *Appl. Sci.* **2025**, *15*, 12787. [CrossRef]
19. Lau, K.W.; Po, L.-M.; Rehman, Y.A.U. Large Separable Kernel Attention: Rethinking the Large Kernel Attention Design in CNN. *arXiv* **2023**, arXiv:2309.01439. [CrossRef]
20. Stergiou, A.; Poppe, R.; Kalliatakis, G. Refining Activation Downsampling With SoftPool. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 10357–10366. Available online: https://openaccess.thecvf.com/content/ICCV2021/html/Stergiou_Refining_Activation_Downsampling_With_SoftPool_ICCV_2021_paper.html (accessed on 1 November 2026) .
21. Zhang, Y. PCSA: Enhancing CNN Performance With Pyramid Channel and Spatial Attention. *IEEE Access* **2024**, *12*, 29062–29071. [CrossRef]
22. Guo, M.-H.; Lu, C.-Z.; Liu, Z.-N.; Cheng, M.-M.; Hu, S.-M. Visual Attention Network. *arXiv* **2022**, arXiv:2202.09741. [CrossRef]
23. Si, Y.; Xu, H.; Zhu, X.; Zhang, W.; Dong, Y.; Chen, Y.; Li, H. SCSA: Exploring the Synergistic Effects Between Spatial and Channel Attention. *arXiv* **2024**, arXiv:2407.05128. [CrossRef]
24. Zubair, F.; Saleh, M.; Akbari, Y.; Al Maadeed, S. A Robust Ensemble Model for Plant Disease Detection Using Deep Learning Architectures. *AgriEngineering* **2025**, *7*, 159. [CrossRef]
25. Satpute, B.S.; Yadav, R.; Yadav, P.K. Dangerous Farm Insects Detection Using Transfer Learning and CNN. In *Proceedings of the 2023 2nd International Conference on Futuristic Technologies (INCOFT), Karnataka, India, 24–26 November 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8. [CrossRef]
26. Chahbouni, A.; El Manaa, I.; El Manaa, K.; Abouch, Y.; Bossoufi, B.; El Ghzaoui, M.; El Alami, R. RyNet: Multi-Level Attention for Rich and Complementary Feature Representation in CNNs. *IEEE Access* **2026**, *14*, 5358–5373. [CrossRef]
27. Zhang, B.; Zheng, S.; Han, L.; Qiu, Y. EFM-ResNet: A Feature Enhanced Network for Tobacco Strips Classification. *IEEE Access* **2025**, *13*, 67017–67028. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
29. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
30. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. [CrossRef]
32. Park, J.; Woo, S.; Lee, J.-Y.; Kweon, I.S. BAM: Bottleneck Attention Module. *arXiv* **2018**, arXiv:1807.06514. [CrossRef]
33. Bakr, E.M.; El-Sallab, A.; Rashwan, M. EMCA: Efficient Multiscale Channel Attention Module. *IEEE Access* **2022**, *10*, 103447–103461. [CrossRef]

34. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global Second-Order Pooling Convolutional Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3024–3033. [CrossRef]
35. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A^2-Nets: Double Attention Networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 350–359. Available online: https://proceedings.neurips.cc/paper_files/paper/2018/file/e165421110ba03099a1c0393373c5b43-Paper.pdf (accessed on 1 November 2026).
36. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019; pp. 1971–1980. [CrossRef]
37. Linsley, D.; Shiebler, D.; Eberhardt, S.; Serre, T. Learning What and Where to Attend. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019. [CrossRef]
38. Fukui, H.; Hirakawa, T.; Yamashita, T.; Fujiyoshi, H. Attention Branch Network: Learning of Attention Mechanism for Visual Explanation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10705–10714. [CrossRef]
39. Lee, H.; Kim, H.-E.; Nam, H. SRM: A Style-Based Recalibration Module for Convolutional Neural Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1854–1862. [CrossRef]
40. Islam, M.M.; Adil, M.A.A.; Talukder, M.A.; Ahamed, M.K.U.; Uddin, M.A.; Hasan, M.K.; Sharmin, S.; Rahman, M.M.; Debnath, S.K. DeepCrop: Deep Learning-Based Crop Disease Prediction with Web Application. *J. Agric. Food Res.* **2023**, *14*, 100764. [CrossRef]
41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. 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. [CrossRef]
42. Wei, T.; Chen, Z.; Yu, X.; Chapman, S.; Melloy, P.; Huang, Z. *A Large-Scale In-the-Wild Dataset for Plant Disease Segmentation [Data Set]*, Version v7; Zenodo: Geneva, Switzerland, 2025. [CrossRef]
43. Xu, W.; Wan, Y. ELA: Efficient Local Attention for Deep Convolutional Neural Networks. *arXiv* **2024**, arXiv:2403.01123. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.