

Article

EATMamba: Evolutionary Token-Refined Vision Mamba for Tomato Leaf Disease and Pest Classification

Yingbiao Hu ¹, Huinian Li ², Yu He ¹, Zhenfu Pan ¹, Ningxia Chen ³, Chengcheng Yang ³ and Wei Ke ^{1,*}¹ Faculty of Applied Sciences, Macau Polytechnic University, Macao 999078, China² School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao 999078, China³ College of Artificial Intelligence, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China

* Correspondence: wke@mpu.edu.mo

Abstract

Accurate and efficient recognition of tomato leaf diseases and pests is essential for precision agriculture, yet practical deployment remains challenging due to complex backgrounds, domain shifts, and subtle inter-class visual differences. From a broader mathematical perspective, this task can be viewed as token-level representation refinement under noise, ambiguity, and distribution shift. Recent state-space-model-based vision backbones provide favorable efficiency for high-resolution imagery but lack explicit mechanisms for adaptive feature refinement under noisy conditions. To address these issues, we propose EATMamba, an evolution-inspired Vision Mamba framework for tomato leaf disease and pest classification. Rather than being limited to a task-specific classifier, EATMamba is formulated as an evolution-inspired differentiable token-refinement mechanism designed to be compatible with state-space visual recognition backbones. EATMamba introduces two lightweight and fully differentiable modules—Evolutionary Crossover–Interaction and Knowledge-Guided Mutation–Selection—which perform token-level recombination and selective refinement to emphasize discriminative disease cues while suppressing irrelevant background information. These modules are inspired by crossover, mutation, and selection concepts, but are implemented as trainable differentiable operations over visual tokens, forming a generate–recombine–select style mechanism for representation refinement. The scope of this study is low-cost RGB-based visible-symptom disease and pest classification, rather than pre-symptomatic early disease detection. Extensive experiments on two complementary tomato datasets, including a controlled high-resolution dataset and an in-the-wild farm dataset, demonstrate that EATMamba consistently outperforms representative CNN-, Transformer-, and state-space-model-based baselines. Ablation studies and visualization analyses further confirm the complementary contributions of the proposed modules. Overall, EATMamba provides an effective and efficient framework for fine-grained plant disease recognition and illustrates how evolution-inspired principles can be incorporated into modern vision backbones for robust agricultural image analysis.



Academic Editor: Jian-Fang Hu

Received: 3 June 2026

Revised: 19 July 2026

Accepted: 22 July 2026

Published: 5 August 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](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: Vision Mamba; evolution-inspired token refinement; state space models; tomato leaf disease recognition; fine-grained agricultural image classification; pest classification

MSC: 68T07; 68T10; 68T45

1. Introduction

Tomato is one of the most widely cultivated horticultural crops, yet its yield and quality are persistently threatened by foliar diseases and pest infestations that can spread quickly in greenhouses and open fields. Timely and accurate recognition of these diseases and pests is therefore important for reducing yield loss, guiding pesticide use, and supporting precision crop management. In practical production, however, disease diagnosis still relies heavily on manual observation by farmers or agricultural technicians. This process is labor-intensive, subjective, and difficult to scale for continuous field monitoring, especially when non-expert farmers need to distinguish visually similar symptoms under complex environmental conditions. Vision-based diagnosis is attractive because it is non-destructive, fast, and compatible with low-cost imaging devices, enabling scalable monitoring and timely intervention. However, practical deployment still faces a clear gap between controlled-condition recognition and in-the-wild performance: symptom appearance varies with illumination [1], background clutter, viewpoint, growth stage, and mixed stress factors [2], while different diseases may exhibit highly similar lesion patterns, making fine-grained discrimination error-prone under field conditions [3]. Importantly, the presence of visible symptoms does not necessarily mean that accurate disease or pest classification is straightforward. For example, visually similar categories, such as Early Blight and Late Blight, or Bacterial Spot and Septoria Leaf Spot, may share necrotic spots, chlorotic halos, and irregular discoloration patterns, which can lead to misdiagnosis by non-expert observers and unstable model predictions under complex field conditions. Early deep learning systems verified the feasibility of image-based plant disease recognition on curated datasets [4], but also revealed reduced robustness when images deviate from laboratory settings [1]. Recent studies further emphasize that bridging the gap between controlled-condition recognition and practical deployment remains an open challenge for plant disease diagnosis [5]. These factors motivate models that are resilient to domain shifts and background noise while remaining efficient for high-resolution imagery where small lesions can be lost after aggressive downsampling [6]. Data augmentation strategies such as mixup and CutMix, as well as generative approaches based on conditional adversarial networks [7], have been shown to improve model robustness and generalization under diverse acquisition conditions [8], which is particularly relevant for agricultural settings where training data may be captured under varied lighting and background conditions [9].

These characteristics lead to three practical challenges. First, disease-related regions are often sparse and occupy only a small portion of the image, especially when visible symptoms are small or weak. Second, background textures, illumination changes, and viewpoint variations may dominate visual responses and mislead the classifier. Third, several tomato diseases exhibit highly similar local symptoms, such as necrotic spots, chlorotic halos, and irregular leaf discoloration, making purely local texture modeling insufficient for reliable fine-grained discrimination. Therefore, an effective model should preserve high-resolution lesion details, model long-range contextual dependencies, and adaptively refine discriminative evidence under noisy acquisition conditions [10]. From a general visual pattern-recognition viewpoint, these difficulties can be formulated as a token-level refinement problem in which sparse discriminative tokens should be strengthened while nuisance tokens are suppressed. This perspective is not specific to tomato leaves; it also applies to image classification tasks where target evidence is weak, distributed, or partially corrupted.

To study these challenges in a realistic tomato setting, we consider two complementary datasets with different acquisition regimes. The Tomato Leaves Dataset (TLD) was constructed by integrating our collected high-resolution tomato leaf images with publicly available tomato leaf disease images from the Kaggle Tomato Leaf Disease Dataset

(6 Classes) [11]. The Kaggle subset contains RGB tomato leaf images organized for image classification and covers five common tomato leaf disease categories and one healthy category. After dataset integration, class-name harmonization, duplicate screening, and quality filtering, TLD contains over 20,000 RGB images covering nine tomato leaf diseases, one pest class (two-spotted spider mite), and healthy leaves. Detailed class-level statistics of TLD are provided in Section 4.1.

The Tomato Leaf Disease Detection (TLDD) dataset contains approximately 10,200 RGB images (256×256) captured on household farms using a Canon EOS 1300D camera, covering eight diseases, the same pest class, and healthy leaves; compared with TLD, TLDD exhibits stronger background diversity and more pronounced in-the-wild variations. Detailed class-level statistics of TLDD are also provided in Section 4.1. Together, these datasets highlight a key tension for practical systems: TLD encourages fine-grained recognition with rich spatial details and partially curated public-source samples, whereas TLDD stresses robustness under uncontrolled acquisition.

Figure 1 illustrates representative samples and typical visual challenges from the two datasets, including controlled high-resolution symptoms, in-the-wild background clutter, illumination variation, and fine-grained inter-class ambiguity.

From an architectural perspective, convolutional neural networks (CNNs) such as ResNet [12] have been widely adopted for feature extraction in plant disease recognition due to their strong inductive bias for local lesion textures and edge patterns. Efficient architectures such as EfficientNet further demonstrate that compound scaling of depth, width, and resolution can yield favorable accuracy–efficiency trade-offs, although their deployment for fine-grained agricultural diagnosis remains less explored compared with that for generic visual recognition tasks. Nevertheless, lesion evidence is often sparse, distributed, and context-dependent, which benefits from long-range interaction and global context modeling. Vision Transformers (ViTs) provide global token interactions and have become competitive backbones for visual recognition [13]. Data-efficient training strategies (e.g., DeiT) reduce the data and compute barrier, while hierarchical designs (e.g., Swin Transformer [14]) improve scalability by restricting attention to local windows and enabling multi-scale learning. Meanwhile, ConvNeXt shows that modernized convolutional designs can narrow the gap with transformers, underscoring that both architectural efficiency and global-context capability matter in practice [15]. Despite these advances, attention-based models still incur non-trivial cost at higher resolutions, whereas high-resolution inputs are often desirable in agricultural imagery to preserve small lesions and subtle texture cues.

State space models (SSMs) have recently emerged as a promising alternative to attention, offering efficient sequence processing with strong long-range dependency modeling. Mamba introduces selective state space models whose parameters are input-dependent, enabling content-aware propagation with hardware-friendly efficiency [16]. These properties make SSM/Mamba-based backbones particularly suitable for high-resolution leaf image analysis, where long visual token sequences need to be processed efficiently while preserving both subtle lesion details and global contextual cues. Building upon this direction, Vision Mamba (Vim) constructs a generic vision backbone by representing images as token sequences and applying bidirectional Mamba blocks to aggregate data-dependent global context with favorable compute and memory scaling [17]. In parallel, VMamba adapts selective scan to 2D vision structures through multi-route scanning, further improving scaling behavior for high-resolution visual inputs [18]. While SSM-based backbones make high-resolution inference more practical, tomato disease recognition still suffers from domain shift and background distractors, where sparse discriminative symptoms can be overwhelmed by nuisance textures.

TLD Dataset (3264×2448, controlled conditions)



Early Blight



Late Blight



Healthy

TLDD Dataset (256×256, in-the-wild farm)



Bacterial Spot



Septoria Spot



Yellow Leaf Curl

Inter-class Visual Similarity



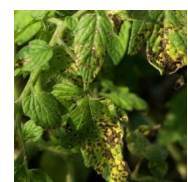
VS



Early Blight vs Late Blight



VS



Bacterial Spot vs Septoria

Figure 1. Representative samples and visual challenges in tomato leaf disease and pest classification. The TLD contains high-resolution images collected under laboratory or greenhouse conditions, where fine lesion details are preserved. The TLDD contains in-the-wild farm images with stronger background clutter, illumination variation, viewpoint changes, and scale differences. Several disease categories exhibit visually similar symptoms, making fine-grained classification challenging.

However, most existing Mamba-based vision backbones mainly focus on efficient global sequence modeling. They propagate visual tokens through selective state-space dynamics, but do not explicitly generate alternative token refinements, evaluate their usefulness, or selectively retain disease-relevant evidence. This limitation is particularly important for agricultural disease recognition, where discriminative symptoms may be sparse, weak, and easily confused with background textures. A mechanism that can reorganize token evidence through learnable recombination and selection is therefore desirable. Mathematically, this motivates an evolution-inspired differentiable refinement mechanism that maps visual tokens to improved representations through recombination, candidate generation, and soft selection rather than feature propagation alone.

Evolution-inspired computation offers a complementary design perspective for adaptive token refinement through variation and selection. EATFormer demonstrates that evolutionary principles can guide effective feature interaction and aggregation in transformer-style vision models [19]. Evolution Transformer further shows that evolution-strategy-like updates can be represented and learned as parametric operators, supporting the feasibility

of embedding evolution-inspired generate-and-select dynamics into neural models [20]. These perspectives motivate an efficient backbone that not only models global context but also explicitly reorganizes token evidence, retaining beneficial refinements and suppressing distractors under domain shift.

Motivated by the above, we propose EATMamba, an evolution-inspired Vim-based framework for tomato leaf disease and pest classification. The model is positioned as a differentiable token-refinement mechanism, with tomato leaf disease classification serving as a representative fine-grained validation scenario. EATMamba treats image tokens as refinement candidates and introduces two fully differentiable evolution-inspired modules inserted stage-wisely on top of Vim: (1) an *Evolutionary Crossover–Interaction* module that performs soft recombination among token representations to consolidate globally consistent disease cues and (2) a *Knowledge-Guided Mutation–Selection* module that injects structured perturbations guided by a learnable memory and applies soft selection to retain beneficial variations. Both modules are regulated by learnable intensity coefficients (stage-wise α and β), enabling adaptive balancing between Vim backbone propagation and evolution-inspired refinements.

The main contributions of this study are summarized as follows:

- We propose EATMamba, an evolution-inspired token-refined Vision Mamba framework for tomato leaf disease and pest classification. Beyond this specific application, the framework is formulated as a differentiable token-refinement mechanism for visual pattern recognition under noise, ambiguity, and distribution shift. The proposed framework combines the long-range modeling capability of state-space vision backbones with explicit token-level evidence refinement.
- We design an Evolutionary Crossover–Interaction (ECI) module that performs crossover-inspired soft token recombination. In mathematical terms, the recombination distribution defines a soft interaction matrix over visual tokens. This module strengthens globally consistent disease cues while reducing the influence of irrelevant background tokens.
- We develop a Knowledge-Guided Mutation–Selection (KGMS) module, which uses a learnable memory bank to guide mutation-inspired candidate generation and a soft selection gate to retain beneficial refinements. This separates memory retrieval, candidate generation, and soft selection into explicit differentiable steps, providing an end-to-end trainable generate–recombine–select style mechanism for sparse lesion representation.
- We clarify the mathematical relationship between the proposed evolution-inspired refinement mechanism and self-attention, token mixing, state-space modeling, and memory-based refinement through theoretical discussion and comparative analysis.
- We conduct extensive experiments on two complementary tomato datasets covering controlled and in-the-wild acquisition conditions. EATMamba consistently outperforms representative CNN-, Transformer-, and SSM-based baselines, including ConvNeXt, MambaVision, and EfficientVMamba. Ablation, visualization, and efficiency analyses verify the effectiveness of the proposed evolution-inspired token refinement strategy.

2. Related Work

2.1. Efficient CNN Backbones for Fine-Grained Plant Disease Recognition

CNN backbones remain a strong baseline for leaf disease recognition because their localized inductive bias aligns well with lesion textures and edge patterns. Representative architectures such as ResNet and DenseNet demonstrate that deep hierarchical representations and dense connectivity can significantly improve feature reuse and optimization

stability [12]. For practical agricultural deployment, computational efficiency is equally important, motivating lightweight CNN families. MobileNetV2 reduces compute via inverted residuals and depthwise separable convolutions, while ShuffleNetV2 emphasizes hardware-aware efficiency beyond FLOPs, yielding favorable speed–accuracy trade-offs on resource-limited platforms. EfficientNet further proposes compound scaling to jointly balance depth, width, and resolution, which is particularly relevant when preserving high-resolution cues is beneficial for small or sparse lesions. Lightweight architectures such as GhostNet generate additional feature maps through efficient linear transformations, and RepVGG employs structural reparameterization to decouple training-time multi-branch topologies from inference-time efficiency. The ResNeXt family demonstrates that aggregated residual transformations improve representational capacity without substantially increasing parameter count. Efficient channel attention mechanisms such as ECA-Net further enhance feature discriminability with negligible overhead. The ConvNeXt family systematically modernizes convolutional designs to bridge the performance gap with transformers through improved architectural recipes [15].

Despite their effectiveness, purely convolutional architectures may fail to capture global context in cluttered backgrounds or under symptom ambiguity, where discriminative evidence is spatially distributed across the leaf. Domain-specific studies further confirm that CNN-based plant diagnosis systems, while achieving strong performance on curated benchmarks, degrade under field conditions with variable lighting and complex backgrounds [21]. Boundary-aware feature enhancement methods such as SEAFEC have been proposed to improve feature extraction and disease recognition under challenging field conditions [22]. Efficient tick-shape network designs have also been explored for resource-constrained image classification [23], while hybrid ResNet-PCA approaches with deep neural networks have shown promise for sustainable plant disease detection [24]. Flow-based calibration networks further demonstrate the importance of learnable geometric transformations in visual perception systems [25].

2.2. Vision Transformers and Hybrid Backbones for Global Context Modeling

Vision Transformers (ViTs) address the limited receptive-field issue by modeling global token interactions through self-attention [13]. Subsequent efforts improve training efficiency and inductive bias: DeiT introduces transformer distillation to reduce data requirements, and Swin Transformer constructs a hierarchical backbone with shifted-window attention for scalable multi-scale representations [14]. Beyond vanilla ViTs, several designs target efficiency and dense prediction. Pyramid Vision Transformer (PVT) uses a progressive pyramid structure to alleviate the high-cost low-resolution output issue of standard ViTs, making it a versatile backbone for dense tasks. CvT injects convolutional token embeddings and projections into transformers to strengthen locality while retaining global modeling. CoAtNet further combines convolution and attention in a principled stage-wise manner, achieving strong accuracy–efficiency trade-offs across data regimes. The CSWin Transformer introduces cross-shaped window attention, enabling efficient capture of both horizontal and vertical contextual information [26]. MaxViT proposes multi-axis attention combining blocked local and dilated global attention with linear complexity, making it scalable to high-resolution inputs. PoolFormer demonstrates that the MetaFormer architecture itself, rather than the specific token mixer, is a key driver of transformer performance. VOLO further introduces outlook attention that dynamically aggregates local features for fine-grained recognition. MLP-Mixer shows that even MLP-based token mixing can achieve competitive results, further broadening the design space for vision backbones. GAN-boosted vision transformers with explainable AI have also shown promising results for multiclass plant disease classification [27]. Fine-grained perception enhancement

through counterfactual token selection further improves ViT discriminability [28]. Spectral-domain tokenization has been proposed to enable efficient ViT training under limited data regimes [29]. Attention-locating algorithms that explicitly suppress background interference have been developed for fine-grained visual classification [30], and semantic feature integration networks provide complementary multi-level feature fusion for fine-grained recognition [31].

Cross-modal query–prototype matching further suggests that prototype-level alignment can improve the utilization of heterogeneous visual and textual cues without additional task-specific training [32].

Nevertheless, attention mechanisms can still incur non-trivial compute/memory at high resolutions, motivating alternatives that preserve long-range modeling while scaling more favorably [33].

2.3. State Space Models for Vision: From Structured SSMs to Mamba-Based Backbones

State space models (SSMs) provide an alternative route to long-range dependency modeling with favorable computational scaling. Structured state space models such as S4 show that long sequences can be modeled efficiently via a structured parameterization of state transitions, enabling competitive performance on long-context benchmarks [34]. Mamba advances this line with selective SSMs whose parameters are input-dependent, enabling content-aware sequence propagation with linear-time complexity and hardware-friendly implementation [16]. Building on Mamba, Vision Mamba (Vim) constructs a vision backbone by converting images into token sequences and applying bidirectional Mamba blocks to aggregate global context efficiently [17]. VMamba further adapts selective scan to 2D vision structures through multi-route scanning to better match spatial layouts and improve scaling on high-resolution inputs [18]. PlainMamba enhances the non-hierarchical Mamba for visual recognition by leveraging continuous 2D scanning and direction-aware updating [35]. More recently, MambaVision explores a hybrid Mamba–Transformer backbone and reports strong performance across classification and downstream vision tasks [36]. VMamba-FCS further extends the VMamba family with a feature-cross scanning strategy for enhanced visual representation learning [10]. LocalMamba further enhances spatial locality by employing a windowed selective scan [37], while EfficientVMamba optimizes the balance between efficiency and accuracy through a hierarchical SSM design [38]. In contrast, MambaOut contends that the SSM component might be redundant for image classification tasks, and that gated CNN architectures can attain comparable performance [39]. Meanwhile, distillation-free scaling of foundation models provides an alternative perspective on efficient vision model design [40]. MambaML further extends SSM-based modeling to multi-label image classification tasks [41], while 2DMamba adapts selective scan for efficient 2D image representation with applications in gigapixel imagery [42]. The role of state space models in vision remains an active topic of investigation [43]. Continual prior compensation has also been explored for sequential human motion prediction, showing that progressively exploiting prior predictions can improve long-range visual sequence modeling [44]. These SSM-based backbones are appealing for agricultural imagery, where high-resolution inference is often required. However, because they are typically feed-forward, they could benefit from an additional mechanism that explicitly reorganizes token evidence in the presence of background distractors and domain shifts [45].

2.4. Evolutionary Computation and Evolution-Inspired Neural Modeling

Evolutionary computation (EC) offers a complementary inductive bias through variation and selection. NEAT demonstrates that evolving both network weights and topologies can improve neuroevolution by progressively augmenting structures [46]. In large-scale

vision settings, evolutionary search has been used to discover strong image classifiers. Genetic CNN applies genetic operations to evolve convolutional architectures. Evolution strategies have also been studied as scalable black-box optimizers, offering parallel-friendly optimization dynamics that complement gradient-based learning [47]. Neuroevolutionary design further demonstrates that evolutionary algorithms can guide deep neural network construction, highlighting the flexibility of evolutionary search beyond architecture design [48]. Evolutionary multi-objective search has also been applied to spiking neural architecture design, extending evolutionary principles to neuromorphic computing [47]. Graph-based evolutionary NAS provides another direction for efficient architecture exploration [49]. Evolutionary NAS methods have also been applied to agricultural image recognition through Taguchi design and differential evolution [50]. Adversarially robust lightweight architectures discovered through differentiable NAS further broaden the scope of evolution-inspired neural design [51].

Recently, EC concepts have been connected more directly to representation learning: EATFormer formulates evolution-inspired blocks for transformer-style information interaction and aggregation [19]. Evolution Transformer shows that evolution-style update rules can be represented and learned as parametric operators, supporting differentiable “generate-and-select” dynamics inside neural models [20]. Neuro-inspired domain generalization methods further demonstrate that biologically motivated design principles can enhance model robustness under distribution shift [52]. These advances motivate embedding explicit, learnable evolution-inspired refinement operators at the token level to refine representations under ambiguity.

2.5. Multispectral and Hyperspectral Imaging for Plant Disease Diagnosis

Beyond RGB-based recognition, multispectral and hyperspectral imaging has been widely explored for plant disease detection because spectral responses contain information related to pigments, moisture variation, and biochemical changes [53]. Compared with RGB images, hyperspectral data can provide richer spectral signatures and has been reported to be useful for early-stage or pre-symptomatic disease monitoring. However, hyperspectral systems usually require specialized sensors, calibration procedures, and higher acquisition costs, which limit their large-scale deployment in practical agricultural environments.

Therefore, this study focuses on RGB-based visible symptom classification, which better matches low-cost field deployment scenarios. Nevertheless, multispectral and hyperspectral extension remains an important future direction.

2.6. Positioning of EATMamba

Existing CNN and transformer families provide strong local and global modeling capabilities, respectively, but may face efficiency bottlenecks when high-resolution cues are critical. SSM-based backbones offer linear-complexity global modeling that is well aligned with high-resolution agricultural imagery, while evolution-inspired modeling provides a principled mechanism to generate candidate refinements and retain beneficial evidence. A recent tomato-focused review further highlights practical constraints such as limited data regimes, multimodal heterogeneity, robustness, and deployment cost, motivating architectures that are both efficient and adaptive [54]. Few-shot learning strategies such as sequence-weighted ensemble meta-learning have also been explored to address data scarcity in crop disease recognition [55]. Noisy-label person re-identification studies further indicate that probabilistic evidence propagation can help distinguish unreliable samples from hard but informative training instances [56]. EATMamba is positioned to bridge these gaps by integrating evolution-inspired differentiable refinement modules into a Vision Mamba backbone. While the Vim-S backbone provides linear-complexity global

modeling, the proposed ECI module introduces a controlled quadratic overhead for token-level recombination. The overall framework aims to keep this overhead moderate through sparse stage-wise insertion while enhancing token-level evidence re-organization for robust tomato disease and pest recognition.

It is worth emphasizing that EATMamba differs from directly applying existing attention or evolution-inspired transformer blocks. First, EATMamba is built upon a state-space vision backbone, and the proposed modules are inserted as token-refinement operations rather than replacing the backbone propagation mechanism. Second, ECI formulates token interaction as differentiable token-level recombination with residual fusion, while KGMS introduces memory-guided mutation-inspired candidate generation and soft selection. Third, the stage-wise intensity coefficients explicitly control the contribution of crossover and mutation-selection, making the evolution-inspired refinement process adaptive across network depths. These designs make EATMamba particularly suitable for fine-grained agricultural recognition, where the model must preserve weak lesion evidence while suppressing background interference. More generally, EATMamba can be interpreted as a token-refinement operator that complements, rather than replaces, existing token modeling mechanisms. Self-attention emphasizes affinity-based contextual aggregation, token mixing emphasizes deterministic token transformation, state-space models emphasize sequential propagation, and memory modules emphasize prototype retrieval; EATMamba separates crossover-based recombination, memory-guided mutation, and soft selection.

Table 1 summarizes the conceptual differences between EATMamba and related token-refinement mechanisms.

Table 1. Conceptual comparison between EATMamba and related token refinement mechanisms.

Mechanism	Main Operation	Candidate Generation	Selection	Relation to EATMamba
Self-attention	Affinity-based weighted aggregation	No explicit candidate generation	Implicit attention weights	ECI also uses soft affinities, but treats them as crossover-inspired recombination distributions.
Token mixing	Deterministic token transformation	No	No explicit selection	EATMamba refines tokens through recombination, candidate generation, and soft selection.
Memory module	Prototype retrieval or memory readout	Partial	Usually implicit	KGMS uses memory retrieval as guidance for mutation-inspired candidate generation.
Gating module	Feature-wise or token-wise modulation	Usually no	Gate-based selection	KGMS applies a soft selection gate after memory-guided candidate generation, not direct gating alone.
EATMamba	Recombination + memory-guided candidate generation + soft selection	Yes	Differentiable soft selection	Provides an evolution-inspired generate–recombine–select refinement mechanism within a Vision Mamba backbone.

This comparison clarifies that EATMamba should not be interpreted as a classical evolutionary algorithm or as a simple re-labeling of attention, memory retrieval, or gating. Instead, it translates the generate–recombine–select design principle into differentiable token refinement operations compatible with state-space visual backbones.

3. Method

3.1. Problem Formulation and Evolutionary Perspective

We first formulate EATMamba at the level of a generic visual token matrix rather than a specific crop image. Let $X \in \mathbb{R}^{M \times D}$ denote a visual token set, where each row represents one token representation and the full matrix represents the token state at a given network depth. The proposed refinement operator maps the current token set to a refined token set through differentiable recombination, candidate generation, and soft selection.

$$X^+ = X + \alpha(\mathcal{C}_\theta(X) - X) + \beta(\mathcal{S}_\theta(\mathcal{C}_\theta(X)) - \mathcal{C}_\theta(X)), \quad (1)$$

This formulation is independent of a specific crop category and can be inserted into token-based visual recognition backbones. Given an input crop image $I \in \mathbb{R}^{H \times W \times 3}$ with label $y \in \{1, \dots, C\}$, our goal is to learn a classifier f_ω that predicts logits $\hat{y} \in \mathbb{R}^C$. We build EATMamba upon the Vision Mamba (Vim-S) backbone and integrate two evolution-inspired differentiable refinement modules to refine token representations in a biologically inspired yet fully differentiable manner.

We use evolutionary terms only as design analogies. In this formulation, each token representation is treated as a refinement unit, and the token set is treated as a collection of candidate representations. Crossover, mutation, and selection are implemented as differentiable token recombination, mutation-inspired candidate generation, and soft selection, rather than as population-level stochastic evolutionary optimization. Following this view, EATMamba introduces two evolution-inspired refinement operators that can be composed into a single token-refinement step while remaining compatible with Vim's sequential state-space modeling. In this interpretation, Vim provides long-range state-space propagation, whereas EATMamba provides an explicit differentiable token-refinement step before selected backbone blocks.

Figure 2 provides a high-level overview of the complete EATMamba workflow, while Figure 3 presents the detailed stage-wise architecture and module composition. The input image is first converted into patch tokens with positional embeddings and a task token. The tokens are then processed by the Vision Mamba backbone, where the proposed token-refinement operator is inserted in a stage-wise manner. Each token-refinement operator consists of ECI and KGMS, and its contribution is controlled by learnable crossover-inspired recombination and mutation-selection-inspired refinement intensities.

3.2. Vim Backbone and Tokenization

We follow Vim-S to convert the image into a visual sequence. Let P be the patch size. The number of patches is

$$J = \frac{HW}{P^2}, \quad M = J + 1, \quad (2)$$

where M includes one task token. Denote by $t_p^j \in \mathbb{R}^{P \times P \times 3}$ the j -th patch, and let $W \in \mathbb{R}^{(P^2 \cdot 3) \times D}$ be the patch projection. We form the initial tokens as

$$T^{(0)} = [t_{\text{cls}}; t_p^1 W; \dots; t_p^J W] + E_{\text{pos}} \in \mathbb{R}^{M \times D}, \quad (3)$$

where $t_{\text{cls}} \in \mathbb{R}^D$ is a learnable task token and $E_{\text{pos}} \in \mathbb{R}^{M \times D}$ is the positional embedding.

Vim-S stacks L bidirectional Mamba blocks with residual connections. Let $\text{VimBlock}_\ell(\cdot)$ denote the ℓ -th block. The backbone update is

$$T^{(\ell)} = \text{VimBlock}_\ell(T^{(\ell-1)}) + T^{(\ell-1)}, \quad \ell = 1, \dots, L. \quad (4)$$

In EATMamba, we keep the Vim block intact (including its bidirectional SSM and Conv1d components) and inject evolution-inspired token refinements into its input tokens.

Unless otherwise specified, we use a patch size of $P = 16$ and an embedding dimension of $D = 384$ (Vim-S configuration). The Vim-S backbone contains $L = 24$ blocks partitioned into $S = 4$ stages with block numbers $\{2, 6, 6, 10\}$. The ETD insertion numbers are set to $K_s = [1, 1, 2, 2]$ from shallow to deep stages. For the evolution-inspired refinement modules, the projection dimension is set to $d = 64$, the memory bank size in KGMS is $K = 32$, and the hidden width of the mutation MLP is $D_h = 256$. These settings are used for all datasets unless otherwise stated.



Figure 2. High-level processing pipeline of EATMamba. An RGB tomato leaf image is tokenized into visual tokens, processed by the Vision Mamba backbone, refined through the ECI and KGMS modules, and finally classified into a disease, pest, or healthy category.

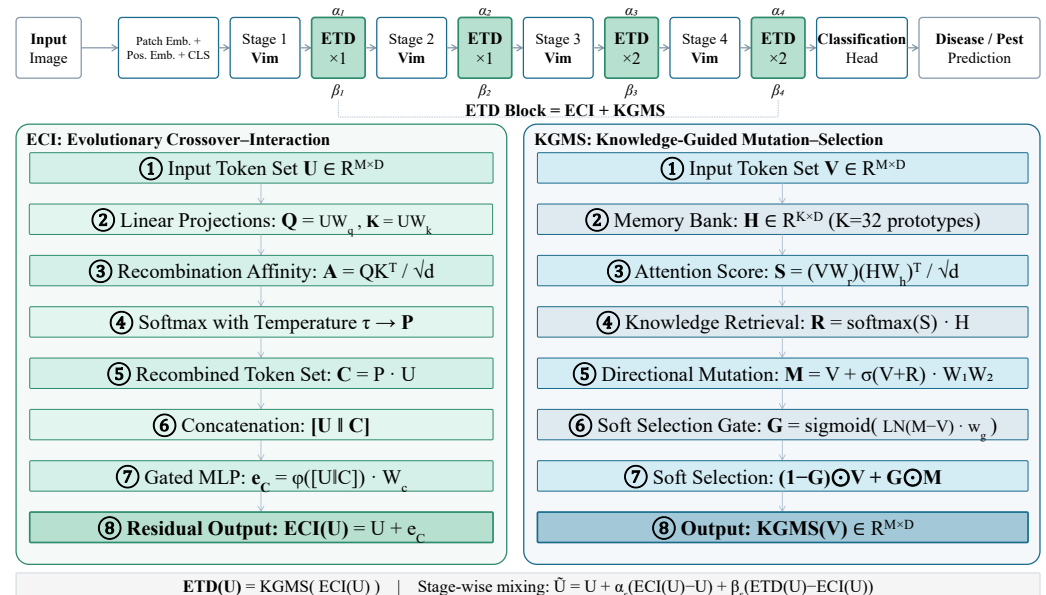


Figure 3. Detailed architecture and module composition of EATMamba. The input tomato leaf image is tokenized into patch tokens and a task token, followed by multiple Vision Mamba stages. The proposed token-refinement operator is inserted stage-wisely and consists of the Evolutionary Crossover-Interaction (ECI) module and the Knowledge-Guided Mutation-Selection (KGMS) module. Learnable coefficients α_s and β_s adaptively control the strengths of crossover-inspired recombination and mutation-selection-inspired refinement at stage s .

3.3. Evolution-Inspired Refinement Module I: Evolutionary Crossover-Interaction (ECI)

The detailed internal construction of the ECI module is illustrated in Figure 4. The motivation of ECI is to enhance disease-relevant token evidence through learnable recombination. In tomato leaf images, discriminative lesion regions are often spatially sparse, while their interpretation depends on surrounding leaf texture, color transition, and global disease context. Purely local modeling may fail to connect these separated cues, whereas direct global aggregation may introduce background noise. Therefore, ECI aims to softly recombine token representations so that weak lesion tokens can interact with contextually related tokens and obtain more discriminative representations.

It should be noted that ECI is not a standard self-attention module. Self-attention usually computes affinity-based weighted aggregation among tokens, whereas ECI is formulated as a crossover-inspired recombination process: a learnable soft recombination distribution weights token sources, and the resulting recombination produces refined token candidates. Thus, the evolutionary terminology is used to describe the design principle of token recombination, while the actual implementation remains a differentiable neural operation trained by backpropagation.

Let $U \in \mathbb{R}^{M \times D}$ be the token set entering the evolution-inspired refinement pathway at some layer. ECI implements crossover-inspired recombination by constructing a soft recombination distribution and recombining token representations accordingly.

We first compute queries/keys using learnable projections $W_q, W_k \in \mathbb{R}^{D \times d}$:

$$Q = UW_q \in \mathbb{R}^{M \times d}, \quad K = UW_k \in \mathbb{R}^{M \times d}. \quad (5)$$

The recombination affinity and the soft recombination distribution are defined as

$$A = \frac{QK^\top}{\sqrt{d}} \in \mathbb{R}^{M \times M}, \quad P = \text{softmax}\left(\frac{A}{\tau}\right) \in \mathbb{R}^{M \times M}, \quad (6)$$

where $\tau > 0$ is a temperature and $\text{softmax}(\cdot)$ is row-wise. The recombined token set is obtained by weighted token recombination

$$C = PU \in \mathbb{R}^{M \times D}. \quad (7)$$

To increase expressiveness while keeping the operator lightweight, we fuse the original and recombined token representations via a gated MLP:

$$\tilde{C} = \phi([U \| C] W_c) \in \mathbb{R}^{M \times D}, \quad (8)$$

where $W_c \in \mathbb{R}^{2D \times D}$, $\|$ is concatenation, and $\phi(\cdot)$ is a pointwise nonlinearity (SiLU). Finally, ECI outputs a refined token set with residual fusion:

$$\text{ECI}(U) = U + \tilde{C}. \quad (9)$$

This operator is permutation-sensitive through positional embeddings, as in Vim, and implements crossover-inspired token recombination through a learnable soft recombination distribution.

Although ECI shares a soft affinity computation with self-attention, its role and interpretation are different. Standard self-attention directly aggregates contextual information for each token, whereas ECI explicitly constructs a recombined token set through soft token recombination and then fuses the original and recombined representations through residual fusion. In this sense, ECI serves as a crossover-inspired token-recombination operator rather than a generic attention layer. The temperature τ controls the sharpness of the recombination distribution, allowing the model to adapt between broad token mixing and selective recombination. Thus, the affinity matrix in ECI is used as a soft recombination distribution over token representations rather than only as a contextual aggregation weight.

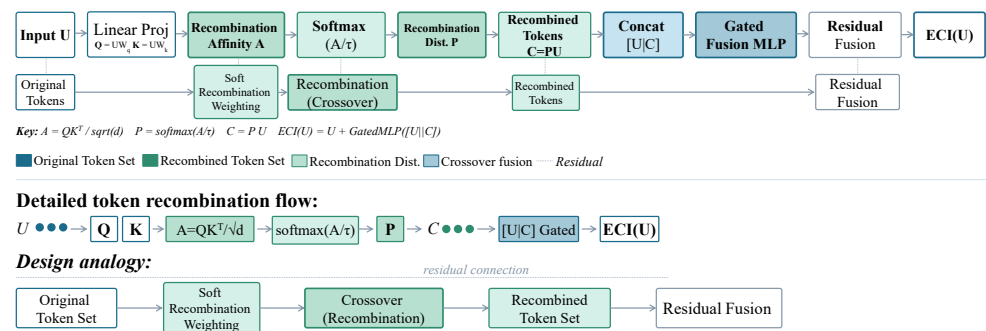


Figure 4. Structure of the Evolutionary Crossover–Interaction (ECI) module. Given an input token set U , ECI computes a soft recombination distribution through projected queries and keys, generates a recombined token set by weighted token recombination, and fuses the original and recombined token representations through a gated MLP and residual connection.

3.4. Evolution-Inspired Refinement Module II: Knowledge-Guided Mutation–Selection (KGMS)

The architecture of the proposed KGMS module is depicted in Figure 5. While ECI recombines existing token evidence, it may still be insufficient when disease-related tokens are weak, sparse, or partially suppressed by background textures. In tomato leaf images, subtle lesion cues can be confused with illumination changes, leaf veins, soil background, or other nuisance patterns. Therefore, an additional refinement mechanism is needed to generate candidate token variations and selectively retain useful changes.

KGMS is designed for this purpose by introducing memory-guided candidate generation and differentiable soft selection. The terms “mutation” and “selection” are used as evolution-inspired design analogies rather than as classical evolutionary optimization. KGMS does not perform population-level stochastic search, non-gradient-based fitness evaluation, or discrete survival decisions. Instead, it implements mutation-inspired candidate generation and soft selection as end-to-end trainable differentiable neural operations over visual tokens.

Let $V \in \mathbb{R}^{M \times D}$ be the input token set to KGMS (typically $V = \text{ECI}(U)$).

We introduce a memory bank $\mathcal{H} \in \mathbb{R}^{K \times D}$ with K prototypes. For each token $v_i \in \mathbb{R}^D$, we retrieve a knowledge vector r_i via attention over memory:

$$S = \frac{(VW_r)(\mathcal{H}W_h)^T}{\sqrt{d}} \in \mathbb{R}^{M \times K}, \quad \Pi = \text{softmax}(S) \in \mathbb{R}^{M \times K}, \quad R = \Pi\mathcal{H} \in \mathbb{R}^{M \times D}, \quad (10)$$

where $W_r, W_h \in \mathbb{R}^{D \times d}$, and R stacks all retrieved r_i . The memory bank $\mathcal{H}$ is initialized with random normal values with a standard deviation of 0.02 and updated via backpropagation during training. We then apply a directional mutation-inspired transformation conditioned on both the token representation and its retrieved knowledge:

$$M = V + \psi((V + R)W_1)W_2 \in \mathbb{R}^{M \times D}, \quad (11)$$

where $W_1 \in \mathbb{R}^{D \times D_h}$, $W_2 \in \mathbb{R}^{D_h \times D}$, $\psi(\cdot)$ is GELU, and D_h is the hidden width.

Selection is implemented as a differentiable soft gate to avoid non-differentiable top- k decisions and preserve small lesion cues. We compute token-wise soft selection gates:

$$G = \text{sigmoid}(\text{LN}(M - V)w_g) \in \mathbb{R}^{M \times 1}, \quad (12)$$

where $w_g \in \mathbb{R}^D$, $\text{LN}(\cdot)$ is LayerNorm, and broadcasting is applied along D . The refined token set is

$$\text{KGMS}(V) = (1 - G) \odot V + G \odot M \in \mathbb{R}^{M \times D}. \quad (13)$$

This module instantiates a knowledge-aware differentiable refinement step: memory retrieval provides prototype guidance, mutation-inspired transformation generates candidate refinements, and soft selection retains useful changes in an end-to-end trainable manner.

KGMS is designed to complement ECI. While ECI emphasizes token-level recombination among existing representations, KGMS generates candidate refinements by injecting memory-guided variations. In a general pattern-recognition setting, this mechanism is useful when discriminative tokens are weak, sparse, or ambiguous; in tomato disease recognition, weak lesion tokens may not be sufficiently enhanced by recombination alone. The learnable memory bank provides reusable disease-related prototypes, and the soft selection gate adaptively determines whether the candidate-refined representation should be fused with the original token representation. Therefore, KGMS provides a controlled candidate-generation and soft-selection mechanism for sparse and ambiguous symptom cues. Compared with direct memory retrieval, KGMS uses the retrieved representation as guidance for mutation-inspired candidate generation and then filters the generated candidate through soft selection.

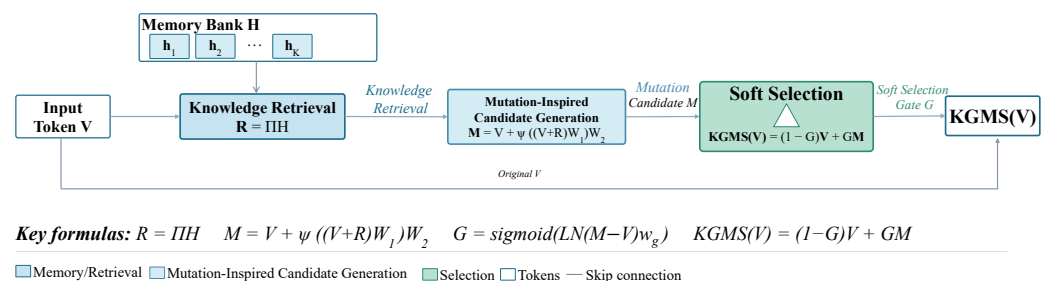


Figure 5. Structure of the Knowledge-Guided Mutation-Selection (KGMS) module. KGMS retrieves disease-related knowledge from a learnable memory bank, generates candidate-refined token representations through a mutation-inspired MLP, and applies a soft selection gate to adaptively fuse the original and candidate-refined representations.

3.5. Overall Architecture of EATMamba: ETD Composition and Stage-Wise Insertion

We define one token-refinement operator (ETD) as the composition of the two evolution-inspired modules:

$$\text{ETD}(U) = \text{KGMS}(\text{ECI}(U)). \quad (14)$$

To control the strength of evolution-inspired refinement and improve interpretability, we introduce learnable nonnegative mixing coefficients that weight the contribution of each operator. For stage s , we learn two scalars:

$$\alpha_s = \text{softplus}(\tilde{\alpha}_s), \quad \beta_s = \text{softplus}(\tilde{\beta}_s), \quad (15)$$

and fuse the intermediate results as

$$\hat{U} = U + \alpha_s (\text{ECI}(U) - U) + \beta_s (\text{ETD}(U) - \text{ECI}(U)). \quad (16)$$

Here, α_s mainly controls crossover/recombination strength, and β_s controls the additional mutation-selection refinement beyond crossover. When both coefficients approach zero, the refinement pathway degenerates to the original backbone, whereas positive coefficients activate crossover-inspired recombination and mutation-selection-inspired refinement.

We partition the L Vim blocks into S stages, where stage s contains L_s blocks and $\sum_{s=1}^S L_s = L$. In each stage, we insert K_s ETD operations (default $K = [1, 1, 2, 2]$ from shallow to deep when $S = 4$; for other S , we allocate larger K_s to deeper stages).

The rationale behind this stage-wise insertion strategy is twofold. First, shallow layers mainly encode low-level edges, colors, and local textures, where excessive token recombination may disturb fine lesion details. Second, deeper layers contain more semantic and compact token representations, making them more suitable for token-level recombination and knowledge-guided refinement. Therefore, we allocate more ETD operations to deeper stages to enhance disease-discriminative semantics while controlling computational overhead. Let $T^{(\ell-1)}$ be the input to a Vim block at some layer within stage s . When ETD is inserted, we update

$$\tilde{T}^{(\ell-1)} = \hat{U} \text{ with } U = T^{(\ell-1)}, \quad T^{(\ell)} = \text{VimBlock}_\ell(\tilde{T}^{(\ell-1)}) + \tilde{T}^{(\ell-1)}. \quad (17)$$

Otherwise, we revert to the vanilla Vim update in Equation (4). This design preserves the backbone computation graph while injecting explicit evolution-inspired refinements at selected depths.

Overall, the evolution-inspired terms used in ECI and KGMS are intended to describe the design intuition of recombination, candidate generation, and soft selection. The proposed modules are not evolutionary algorithms in the classical sense; they are fully differentiable neural operators optimized by standard backpropagation.

3.6. Classification Head and Training Objective

After the final Vim block, we obtain $T^{(L)} \in \mathbb{R}^{M \times D}$. We follow the class-token prediction protocol and compute the feature vector

$$f = \text{Norm}(T_{0,:}^{(L)}) \in \mathbb{R}^D, \quad \hat{y} = W_o f + b_o \in \mathbb{R}^C, \quad (18)$$

where $W_o \in \mathbb{R}^{C \times D}$ and $b_o \in \mathbb{R}^C$. We optimize the standard cross-entropy loss:

$$p = \text{softmax}(\hat{y}), \quad \mathcal{L}_{\text{cls}} = - \sum_{c=1}^C \mathbf{1}[y = c] \log p_c. \quad (19)$$

Optionally, to prevent overly uncertain selection and to encourage confident yet smooth soft-selection behavior, we employ a mild gate regularizer:

$$\mathcal{L}_{\text{gate}} = \frac{1}{M} \sum_{i=1}^M g_i(1 - g_i), \quad \mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{gate}}, \quad (20)$$

where g_i is the i -th scalar in G and $\lambda = 0.01$ balances the regularization strength.

3.7. Complexity Discussion

The proposed operator is differentiable almost everywhere because softmax recombination, memory-guided mutation, and sigmoid selection are all differentiable operations. The row-stochastic recombination distribution gives the crossover-inspired step a controlled recombination interpretation, and the bounded soft selection gate makes the selected output a continuous interpolation between the original and candidate-refined token representations.

Let M denote the token length and D denote the embedding dimension. Each ETD insertion introduces additional costs from ECI and KGMS. For ECI, computing the recombination logits requires $O(M^2 d)$ operations, and generating the recombined token set requires $O(M^2 D)$ operations. For KGMS, memory retrieval requires $O(MKd)$ operations, and the mutation MLP requires $O(MDD_h)$ operations, where K is the number of memory prototypes and D_h is the hidden width.

Although ECI contains a quadratic token interaction term, EATMamba does not apply it densely to all layers. Instead, ETD is inserted sparsely and stage-wisely, with more

insertions in deeper stages where token representations are more semantically compact and better suited for token-level recombination. Because Vim-S employs a non-hierarchical architecture, the token length remains constant across stages; the stage-wise insertion strategy is therefore motivated by representational semantics rather than token-length reduction. Moreover, the projection dimension d and the memory size K are set to moderate values. This design preserves the efficient global modeling capability of the Vim-S backbone while introducing controlled evolution-inspired token refinement.

With $K_s = [1, 1, 2, 2]$ ETD insertions across the four stages, the total parameter count increases from 26.3 M (Vim-S) to 33.5 M (EATMamba), representing a 27.4% overhead. FLOPs increase from 4.8 G to 5.8 G (20.8% overhead) when measured at 224×224 resolution following the fvc core convention. The inference latency increases from 6.5 ms to 7.1 ms per image (batch size = 1), while throughput decreases from 360 to 310 images per second (batch size = 32). These overheads are moderate relative to the accuracy gains (+0.86% Acc on TLD, +1.81% Acc on TLDD), yielding a favorable accuracy–efficiency trade-off for practical agricultural deployment.

4. Experiments

Algorithm 1 summarizes the complete forward inference procedure of EATMamba rather than introducing a standalone optimization or post-processing algorithm. Its overall effectiveness is evaluated through the main-comparison experiments presented later. The ECI and KGMS operations, together with the stage-wise insertion policy, are further assessed in the subsequent ablation and insertion-strategy analyses. Therefore, the complete procedure and its principal operations are quantitatively assessed in the existing experiments.

Algorithm 1 Forward inference procedure of EATMamba

Require: Image I

```

1:  $T^{(0)} \leftarrow$  tokenization by Equation (3)
2: for  $\ell = 1$  to  $L$  do
3:   if ETD inserted at layer  $\ell$  (stage-wise policy) then
4:      $U \leftarrow T^{(\ell-1)}$ 
5:      $U_{cx} \leftarrow$  ECI( $U$ ) by Equation (9)
6:      $U_{etd} \leftarrow$  KGMS( $U_{cx}$ ) by Equation (13)
7:      $\tilde{T}^{(\ell-1)} \leftarrow U + \alpha_s(U_{cx} - U) + \beta_s(U_{etd} - U_{cx})$  by Equation (16)
8:   else
9:      $\tilde{T}^{(\ell-1)} \leftarrow T^{(\ell-1)}$ 
10:  end if
11:   $T^{(\ell)} \leftarrow$  VimBlock $_{\ell}(\tilde{T}^{(\ell-1)}) + \tilde{T}^{(\ell-1)}$  by Equation (17)
12: end for
13:  $f \leftarrow$  Norm( $T_{0,:}^{(L)}), \hat{y} \leftarrow W_o f + b_o$  by Equation (18)
14: return  $\hat{y}$ 
```

4.1. Datasets

Experiments were conducted on two tomato leaf disease and pest datasets with complementary characteristics, covering both controlled and in-the-wild acquisition conditions. Detailed class-level statistics are reported to reflect the fine-grained nature of the recognition task.

The detailed class-level statistics of TLD are summarized in Table 2.

The Tomato Leaves Dataset (TLD) consists of 20,680 RGB images from both self-collected and public-source data. Specifically, part of the dataset was collected from laboratory environments and commercial greenhouses, while an additional public subset was obtained from the Kaggle Tomato Leaf Disease Dataset (6 Classes) [11]. The Kaggle subset is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0),

and it contains RGB images of tomato leaves categorized into six classes, including five common disease classes and one healthy class. In this study, all public-source images were used in accordance with the license requirements, with attribution provided in the references and the dataset source explicitly reported. Before integration, class names were manually harmonized with the taxonomy used in this study, and images with ambiguous labels, severe corruption, or near-duplicate content were removed during quality control. After integration and filtering, TLD covers nine tomato leaf disease classes, one pest class (two-spotted spider mite), and one healthy class, totaling 11 categories. Detailed sample quantity for each category, total sample number, image resolution, and acquisition/source information are summarized in Table 2. The dataset exhibits moderate class imbalance, reflecting both realistic disease occurrence frequencies and the uneven distribution of publicly available tomato disease images. To address this imbalance during evaluation, we report macro-averaged Precision, Recall, and F1-score in addition to accuracy so that minority classes are not overwhelmed by majority classes. During training, we did not apply aggressive over-sampling because repeated minority-class images may increase overfitting for visually homogeneous disease categories. Instead, we used stratified data splitting and standard data augmentation to preserve realistic class distributions while improving generalization.

Table 2. Detailed statistics of the Tomato Leaves Dataset (TLD).

Class	#Images
Bacterial Spot	2261
Early Blight	1964
Late Blight	2490
Leaf Mold	2203
Septoria Leaf Spot	2305
Spider Mites (Two-spotted)	1398
Target Spot	1462
Tomato Yellow Leaf Curl Virus	1631
Tomato Mosaic Virus	1722
Healthy	2441
Powdery Mildew	803
Total Images	20,680
Number of Classes	11
Image Resolution	3264×2448
Data Source	Self-collected + Kaggle public subset
Acquisition Scene	Laboratory/Greenhouse/Public-source images
License of Public Subset	CC BY 4.0

The detailed class-level statistics of TLDD are summarized in Table 3.

The TLDD was constructed by our group using a Canon EOS 1300D camera on household farms and will be made publicly available upon paper acceptance, together with the exact train/validation/test split indices and the random seed used for reproducibility. The per-class sample counts and core dataset specifications of TLDD are listed in Table 3. For the public-source Kaggle subset, the original folder-level class labels were retained, followed by class-name harmonization, visual consistency checking, duplicate screening, and quality filtering before integration. The integrated dataset was then divided into training, validation, and test subsets under the same stratified protocol to ensure a unified evaluation setting.

Table 3. Detailed statistics of the Tomato Leaf Disease Detection (TLDD) dataset.

Class	#Images
Late Blight	1120
Healthy	1050
Early Blight	1010
Septoria Leaf Spot	980
Tomato Yellow Leaf Curl Virus	1030
Bacterial Spot	1040
Target Spot	1000
Tomato Mosaic Virus	990
Leaf Mold	1000
Spider Mites (Two-spotted)	980
Total Images	10,200
Number of Classes	10
Image Resolution	256 × 256
Acquisition Scene	In-the-wild (Farm)

Dataset Annotation and Expert Validation Protocol

For the self-collected images in TLD and TLDD, annotation was conducted under the guidance of domain experts with long-term experience in tomato cultivation, pest management, and field disease diagnosis. Before annotation, the experts provided labeling guidelines based on visible symptom characteristics, including lesion morphology, chlorosis patterns, necrotic regions, leaf discoloration, and pest-related damage. Trained annotators first assigned preliminary labels according to these guidelines and the corresponding field records.

To improve annotation reliability, the preliminary labels were subsequently reviewed by the experts. Images with severe blur, strong occlusion, uncertain symptoms, mixed infections, or inconsistent judgments were excluded from the final dataset or returned for expert discussion. For visually similar categories, such as Early Blight versus Late Blight and Bacterial Spot versus Septoria Leaf Spot, additional expert verification was performed by comparing disease-specific symptom cues, including lesion shape, expansion pattern, chlorotic halos, water-soaked or necrotic regions, and spot distribution. The final label was determined only after expert confirmation. For the public-source Kaggle subset, the original folder-level labels were retained, while visual consistency checking, duplicate screening, and quality filtering were used to reduce label noise before integration.

Because the final labels were obtained through an expert-supervised consensus procedure rather than independent blind annotation by multiple annotators, a numerical inter-annotator agreement coefficient such as Cohen's kappa or Fleiss' kappa was not computed in this study. Instead, we adopted an expert-guided quality-control protocol, including annotation guideline preparation, preliminary labeling, expert review, ambiguity resolution, and exclusion of unreliable samples.

For both datasets, images were split into training, validation, and test sets with a ratio of 70%/10%/20% using a stratified random protocol to preserve the original class distribution across splits. Representative samples and typical visual challenges from both datasets are shown in Figure 1.

A fixed random seed (seed = 42) was used for reproducibility. When visually near-duplicate images or images captured from the same leaf sequence were available, they were assigned to the same subset to reduce potential data leakage. Specifically, near-duplicate detection was performed using perceptual hashing (pHash) with a Hamming-distance threshold of 12; images with hash distances below this threshold were treated as a single group and assigned to the same split. All reported results are computed on the held-out

test set that was not used during model selection or hyperparameter tuning. To facilitate independent verification, we will release the pre-computed split files (CSV with image paths and assigned subsets) alongside the dataset.

4.2. Implementation Details

4.2.1. Experimental Setup

All classification accuracy experiments were conducted on the AutoDL platform equipped with an NVIDIA RTX 3060 GPU (12 GB), Ubuntu 20.04, Python 3.8, and PyTorch 2.0.0 with CUDA 11.8. The physical batch size was set according to GPU memory constraints, and gradient accumulation was used to obtain an effective batch size of 64 when necessary.

Efficiency metrics (Params, FLOPs, latency, throughput, and inference memory) were measured on a separate workstation with an NVIDIA RTX 4090 GPU and CUDA 12.1. All models were evaluated under identical efficiency measurement protocols to ensure fair comparison: the same batch sizes, warm-up procedure, input resolution (224×224), and PyTorch configuration (FP32, `model.eval()`, `torch.no_grad()`, and no `torch.compile` or AMP). The efficiency hardware differs from the training hardware because the RTX 3060 used for training could not accommodate the larger models (e.g., ViT-B/16) at the batch size required for throughput measurement.

All classification results reported in Tables 4 and 5 were obtained from our own training and evaluation runs rather than copied from the original publications. We used the official or publicly available implementations and ImageNet-pretrained weights whenever available, and adapted only the final classification heads to the target label spaces. All baseline models were trained and evaluated using the same dataset splits, data augmentation protocol, pretraining policy, optimization schedule, number of independent runs, and evaluation metrics. Therefore, the reported results represent a controlled comparison under the unified experimental protocol of this study.

Similarly, all computational-efficiency results in Table 6 were re-measured by us under the same hardware and software environment, rather than adopted from the original papers.

All comparison methods were trained and evaluated under identical experimental protocols to ensure fair comparison. The input resolutions used during training and accuracy evaluation were set according to the characteristics of each dataset: TLD images were resized to 512×512 to preserve fine-grained lesion details from the original high-resolution captures, while TLDD images were processed at their original resolution of 256×256 . These dataset-specific resolutions are used only for accuracy experiments (Tables 4 and 5); they are deliberately decoupled from the efficiency benchmarking described in Section 4.4, where all models are evaluated at a uniform 224×224 resolution for fair cross-model cost comparison.

During training, we applied standard image augmentation strategies, including random resized cropping, horizontal flipping, random rotation, color jittering, and normalization to the pre-resized inputs described above. During validation and testing, only deterministic resizing and normalization were applied. Considering the moderate class imbalance in TLD, we did not apply aggressive over-sampling or inverse-frequency class-balanced sampling during training, because repeatedly sampling minority-class images may increase overfitting for visually homogeneous disease categories. Instead, we used stratified data splitting, standard data augmentation, and macro-averaged Precision, Recall, and F1-score for evaluation to reduce the dominance of majority classes.

All backbone models were initialized with ImageNet-pretrained weights when available. The classification heads were randomly initialized and trained from scratch for the

tomato disease and pest classification task. For fair comparison, the same pretraining protocol was used for all applicable baselines.

4.2.2. Hyperparameter Settings

The AdamW optimizer was employed with an initial learning rate of 1×10^{-3} . A step-based learning rate scheduler halved the learning rate every 50 epochs. The batch size was set to 64, and all models were trained for up to 300 epochs. Early stopping with a patience of 30 epochs was applied to prevent overfitting. Dropout (rate = 0.2) and L2 regularization with a weight decay of 1×10^{-5} were used to improve generalization.

The effective batch size of 64 was used to balance gradient stability and GPU-memory constraints, with gradient accumulation applied when necessary. The initial learning rate and step-decay schedule were used as common and stable settings for AdamW-based fine-tuning and were kept identical for all compared models. Dropout, weight decay, and early stopping provide moderate regularization and convergence control. The gate-regularization coefficient was set to a small value of $\lambda = 0.01$ so that the auxiliary gate loss would not dominate the primary classification objective. These settings were fixed across all experiments, and no model-specific tuning was performed on the test set.

To reduce the influence of random initialization and data shuffling, each experiment was repeated with three independent random seeds. Unless otherwise specified, we report the mean performance over three runs. Standard deviations are provided for the main comparison, ablation, and sensitivity experiments.

The hyperparameters can be divided into backbone-related settings, general training settings, and EATMamba-specific settings. The patch size, embedding dimension, and network depth follow the standard Vim-S configuration so that the proposed method remains directly comparable with its backbone. AdamW was used because it provides stable optimization for Transformer- and state-space-model-based architectures. The learning rate, weight decay, dropout rate, maximum number of epochs, and early stopping patience were fixed for all compared models rather than individually optimized for EATMamba, thereby avoiding a comparison biased toward the proposed method.

For the EATMamba-specific parameters, the reduced projection dimension $d = 64$ and mutation-MLP hidden width $D_h = 256$ were selected to provide sufficient refinement capacity while controlling the additional computational cost. The stage-wise insertion policy $K_s = [1, 1, 2, 2]$ is evaluated in the insertion-strategy analysis presented later.

The memory-bank size $K = 32$ and ECI temperature $\tau = 2.0$ were selected using the validation-set sensitivity analyses presented later. The test set was not used for hyperparameter selection. The relatively small performance variation around the selected values indicates that EATMamba is not highly sensitive to these parameters within reasonable ranges.

4.2.3. Evaluation Metrics

As a multi-class image classification task, model performance was evaluated using Accuracy (Acc), macro-averaged Precision, macro-averaged Recall, and macro-averaged F1-score. Macro-averaging was adopted to ensure that each disease category contributes equally to the final metric, which is important when class imbalance exists.

$$Acc = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\hat{y}_i = y_i), \quad (21)$$

$$Precision_{macro} = \frac{1}{C} \sum_{c=1}^C \frac{TP_c}{TP_c + FP_c}, \quad (22)$$

$$Recall_{macro} = \frac{1}{C} \sum_{c=1}^C \frac{TP_c}{TP_c + FN_c}, \quad (23)$$

$$F1_{macro} = \frac{1}{C} \sum_{c=1}^C \frac{2 \cdot Precision_c \cdot Recall_c}{Precision_c + Recall_c}, \quad (24)$$

where TP_c , FP_c , and FN_c denote the true positives, false positives, and false negatives of class c , respectively. $\hat{y}_i$ and y_i denote the predicted and ground-truth labels of the i -th test sample, and n is the total number of test samples.

4.3. Experimental Results and Analysis

Because tomato disease datasets may contain class imbalance and fine-grained inter-class ambiguity, relying only on overall accuracy may obscure minority-class behavior. Therefore, we report Accuracy together with macro-averaged Precision, Recall, and F1-score to provide a more balanced evaluation across disease and pest categories.

We compare EATMamba with representative CNN-based, transformer-based, and state-space based models, including recent strong competitors such as ConvNeXt-T [15], MambaVision-T [36], and EfficientVMamba [38]. Quantitative results on the TLD and TLDD are reported in Tables 4 and 5, respectively.

Table 4. Classification performance on the TLD. Mean $\pm$ std over three independent runs. The best result for each metric is shown in **bold**.

Method	Acc (%)	Precision (%)	Recall (%)	F1 (%)
ResNet-50	94.21 $\pm$ 0.52	93.76 $\pm$ 0.61	93.41 $\pm$ 0.68	93.52 $\pm$ 0.64
EfficientNet-B3	94.88 $\pm$ 0.41	94.57 $\pm$ 0.52	94.08 $\pm$ 0.58	94.24 $\pm$ 0.49
ViT-B/16	95.62 $\pm$ 0.44	95.18 $\pm$ 0.49	94.94 $\pm$ 0.52	94.97 $\pm$ 0.44
Swin-T	96.17 $\pm$ 0.33	95.84 $\pm$ 0.41	95.52 $\pm$ 0.45	95.62 $\pm$ 0.38
ConvNeXt-T	96.25 $\pm$ 0.31	95.92 $\pm$ 0.38	95.61 $\pm$ 0.42	95.71 $\pm$ 0.35
Vim-S	96.03 $\pm$ 0.47	95.71 $\pm$ 0.44	95.78 $\pm$ 0.51	95.68 $\pm$ 0.42
VMamba-T	96.61 $\pm$ 0.28	96.42 $\pm$ 0.33	96.04 $\pm$ 0.41	96.15 $\pm$ 0.36
MambaVision-T	96.72 $\pm$ 0.35	96.35 $\pm$ 0.40	96.18 $\pm$ 0.38	96.22 $\pm$ 0.34
EfficientVMamba	96.58 $\pm$ 0.30	96.28 $\pm$ 0.36	95.95 $\pm$ 0.39	96.08 $\pm$ 0.33
EATMamba	96.89 $\pm$ 0.24	96.56 $\pm$ 0.31	96.52 $\pm$ 0.30	96.26 $\pm$ 0.35

Table 5. Classification performance on the TLDD. Mean $\pm$ std over three independent runs. The best result for each metric is shown in **bold**.

Method	Acc (%)	Precision (%)	Recall (%)	F1 (%)
ResNet-50	88.23 $\pm$ 0.71	87.92 $\pm$ 0.79	86.71 $\pm$ 0.88	87.05 $\pm$ 0.84
EfficientNet-B3	89.42 $\pm$ 0.62	89.20 $\pm$ 0.67	88.51 $\pm$ 0.75	88.72 $\pm$ 0.58
ViT-B/16	90.08 $\pm$ 0.58	89.64 $\pm$ 0.72	89.20 $\pm$ 0.80	89.28 $\pm$ 0.63
Swin-T	91.14 $\pm$ 0.54	90.86 $\pm$ 0.59	90.31 $\pm$ 0.67	90.43 $\pm$ 0.52
ConvNeXt-T	91.28 $\pm$ 0.48	90.95 $\pm$ 0.55	90.45 $\pm$ 0.60	90.58 $\pm$ 0.49
Vim-S	91.06 $\pm$ 0.49	90.71 $\pm$ 0.55	90.55 $\pm$ 0.62	90.52 $\pm$ 0.50
VMamba-T	91.91 $\pm$ 0.45	91.78 $\pm$ 0.51	91.10 $\pm$ 0.60	91.28 $\pm$ 0.48
MambaVision-T	92.05 $\pm$ 0.42	91.82 $\pm$ 0.48	91.35 $\pm$ 0.55	91.52 $\pm$ 0.45
EfficientVMamba	91.76 $\pm$ 0.44	91.55 $\pm$ 0.50	90.98 $\pm$ 0.58	91.18 $\pm$ 0.46
EATMamba	92.87 $\pm$ 0.38	92.15 $\pm$ 0.52	92.48 $\pm$ 0.48	91.92 $\pm$ 0.51

As shown in Tables 4 and 5, the experimental results illustrate a typical performance pattern in fine-grained tomato disease and pest classification. On the relatively controlled TLD, all models achieve high accuracy, but EATMamba still obtains the best overall performance with 96.89% accuracy and 96.26% macro-F1. Compared with Vim-S, EATMamba

improves accuracy by 0.86 percentage points and macro-F1 by 0.58 percentage points. On the more challenging TLDD, the performance gap becomes more pronounced because field images contain stronger background clutter, illumination variation, and symptom ambiguity. EATMamba achieves 92.87% accuracy and 91.92% macro-F1, outperforming Vim-S by 1.81 and 1.40 percentage points, respectively. These consistent gains suggest that evolution-inspired token refinement helps the model better identify disease-relevant cues under challenging acquisition conditions.

The comparison with different baseline families further explains where the improvement comes from. CNN-based models such as ResNet-50 and EfficientNet-B3 are effective at capturing local lesion textures, but their limited global-context modeling makes them less robust when symptoms are scattered or mixed with background clutter. Transformer-based models such as ViT-B/16 and Swin-T provide stronger global or window-based token interaction, but their attention computation becomes less efficient at high resolution, and local window attention may still miss long-range symptom relationships. ConvNeXt-T achieves strong performance through modernized convolutional design, yet its feature refinement remains largely implicit and may not selectively suppress nuisance background tokens.

Compared with Vim-S, EATMamba keeps the efficient long-range modeling advantage of state-space propagation while adding explicit token-level evidence refinement. Compared with VMamba-T and EfficientVMamba, which mainly improve selective scanning and computational efficiency, EATMamba further introduces crossover-inspired recombination and memory-guided soft selection to strengthen sparse disease cues. Compared with MambaVision-T, EATMamba achieves larger gains on TLDD, suggesting that the proposed refinement mechanism is particularly useful under in-the-wild conditions where background noise and symptom ambiguity are more pronounced.

Although the main comparison is reported using dataset-level metrics, the macro-averaged Precision, Recall, and F1-score provide a class-balanced view of performance and reduce the dominance of majority classes. This is particularly relevant for visually similar tomato disease categories, such as Early Blight versus Late Blight and Bacterial Spot versus Septoria Leaf Spot, which may share necrotic regions, chlorotic halos, small spot-like lesions, and irregular discoloration. The consistent improvements in macro-Recall and macro-F1, especially on TLDD, suggest that EATMamba improves recognition behavior beyond simply increasing overall accuracy. Together with the visualization analysis, these results indicate that evolution-inspired token refinement helps the model focus more on disease-relevant regions under fine-grained and noisy conditions.

To reduce the influence of random initialization and data shuffling, all experiments were repeated with three independent random seeds, and mean $\pm$ standard deviation is reported. The gains of EATMamba are consistent across the two datasets, with a more pronounced improvement on TLDD than on TLD. Because TLD is relatively controlled and all strong baselines already achieve high accuracy, the margin on TLD is moderate. In contrast, the larger gains on TLDD indicate that the proposed refinement mechanism is more beneficial under in-the-wild acquisition conditions with background clutter, illumination variation, and symptom ambiguity. Compared with MambaVision-T, EATMamba achieves higher accuracy (+0.17% on TLD, +0.82% on TLDD) and macro-F1 (+0.04% on TLD, +0.40% on TLDD), demonstrating competitive performance even against recent hybrid Mamba–Transformer backbones.

As visualized in Figure 6, EATMamba achieves the highest accuracy and macro-F1 on both datasets, with more pronounced gains on the in-the-wild TLDD.

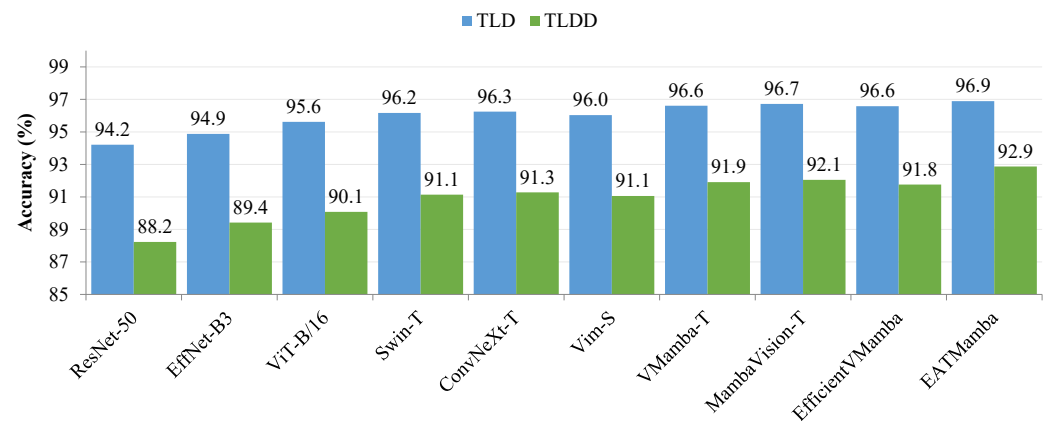


Figure 6. Performance comparison of different models on the TLD and TLDD. EATMamba achieves the highest accuracy and macro-F1 on both datasets, with more pronounced gains on the in-the-wild TLDD.

4.4. Efficiency Comparison

In this study, processing time is quantified using per-image inference latency and batch throughput, computational complexity is quantified using parameter count and FLOPs together with the asymptotic analysis in Section 3.7, and inference efficiency is further evaluated using peak GPU memory consumption.

To evaluate the practical efficiency of different models, we conduct a separate efficiency benchmarking experiment that is independent of the accuracy experiments reported above. All efficiency metrics (Params, FLOPs, latency, throughput, and inference memory) were measured at a uniform 224×224 input resolution on an NVIDIA RTX 4090 GPU with CUDA 12.1. This standardized resolution was deliberately chosen to ensure fair cross-model comparison of computational cost, regardless of the resolutions used during training. The accuracy values reported in the rightmost two columns of Table 6 are the results of the dataset-specific accuracy experiments reported in Tables 4 and 5; they were not re-measured at 224×224 . Latency is measured with batch size = 1; throughput is measured with batch size = 32 to saturate the GPU. FLOPs are computed at 224×224 following the fvc core convention.

Table 6. Efficiency comparison of different models. Efficiency metrics (Params, FLOPs, latency, throughput, and inference memory) were measured at a uniform 224×224 resolution on an NVIDIA RTX 4090 GPU with CUDA 12.1. Accuracy values in the rightmost two columns are obtained from dataset-specific evaluation resolutions (512×512 for TLD, 256×256 for TLDD) as reported in Tables 4 and 5. FLOPs are reported following the fvc core convention (1 fused multiply-add = 1 FLOP). Throughput is measured under standard PyTorch (FP32, no torch.compile or AMP). The highest accuracy in each dataset is shown in **bold**.

Method	Params (M)	FLOPs (G)	Latency (ms)	Throughput (img/s)	Inf. Mem. (GB)	TLD Acc (%)	TLDD Acc (%)
ResNet-50	25.6	4.1	4.2	550	1.8	94.21	88.23
EfficientNet-B3	12.0	1.2	3.5	680	1.3	94.88	89.42
ViT-B/16	86.6	17.6	12.5	175	3.2	95.62	90.08
Swin-T	28.3	4.5	7.8	390	2.4	96.17	91.14
ConvNeXt-T	28.6	4.5	5.8	420	2.2	96.25	91.28
Vim-S	26.3	4.8	6.5	360	2.2	96.03	91.06
VMamba-T	30.1	5.2	6.2	385	2.3	96.61	91.91
MambaVision-T	35.0	10.7	8.3	340	2.6	96.72	92.05
EfficientVMamba	26.8	4.9	6.0	400	2.2	96.58	91.76
EATMamba (Ours)	33.5	5.8	7.1	310	2.5	96.89	92.87

Latency is the wall-clock time per image measured with batch size 1. Throughput is measured in images per second with batch size 32. Inference memory denotes peak GPU memory during FP32 inference with batch size 32. FLOPs are measured at 224×224 using fvc core. Efficiency metrics are measured at 224×224 on an RTX 4090, whereas accuracy is reported at the dataset-specific evaluation resolutions.

As shown in Table 6, EATMamba introduces a moderate computational overhead compared with Vim-S. The parameter count increases from 26.3 M to 33.5 M (+7.2 M, +27.4%), and FLOPs increase from 4.8 G to 5.8 G (+1.0 G, +20.8%), while the latency increases from 6.5 ms to 7.1 ms per image. Compared with ViT-B/16 and MambaVision-T, EATMamba achieves higher accuracy with substantially lower computational cost. These results highlight a favorable accuracy–efficiency trade-off: the proposed evolution-inspired modules improve classification performance while preserving the efficient sequence modeling advantage of Vision Mamba. The comprehensive accuracy–cost trade-off of all compared models is intuitively illustrated in Figure 7.

Regarding resolution-dependent cost, it is important to note that, while FLOPs and latency in Table 6 are measured at 224×224 for fair cross-model comparison, ECI contains an $O(M^2D)$ token interaction where M scales with image resolution. At 512×512 (TLD training resolution), the token count increases from 197 to 1025, substantially increasing the ECI overhead. For applications requiring ultra-high-resolution inference, the quadratic ECI cost may necessitate low-rank or sparse token-recombination approximations. We report FLOPs at 224×224 following the standard fvcore convention for cross-model comparison; actual FLOPs at higher resolutions will differ proportionally to the token-length increase.

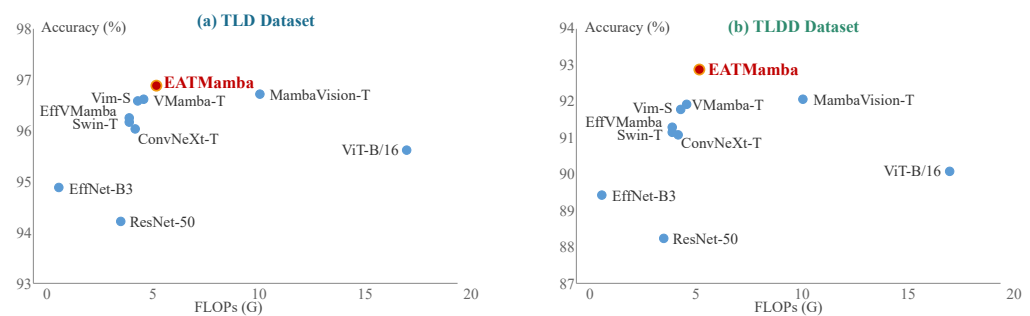


Figure 7. Accuracy–efficiency trade-off comparison. Each point denotes one model, where the x-axis represents computational cost (FLOPs measured at 224×224) and the y-axis represents classification accuracy (obtained under dataset-specific evaluation resolutions). EATMamba achieves a favorable trade-off between recognition performance and computational efficiency.

4.5. Ablation Studies and Analysis

Ablation experiments were conducted on the TLD and TLDD to evaluate the principal operations summarized in Algorithm 1. The Vim-S baseline represents the forward procedure without ETD refinement; Vim-S + ECI evaluates the crossover-inspired recombination operation; Vim-S + KGMS evaluates the memory-guided candidate-generation and soft-selection operation; and the full EATMamba evaluates the complete forward procedure. Standard deviations are reported over three independent runs.

Tables 7 and 8 report the component-level ablation results on TLD and TLDD, respectively.

Table 7. Ablation study on the TLD. Mean $\pm$ std over three runs. The best result for each metric is shown in **bold**.

Configuration	Acc (%)	Precision (%)	Recall (%)	F1 (%)
Vim-S (baseline)	96.03 $\pm$ 0.47	95.71 $\pm$ 0.44	95.78 $\pm$ 0.51	95.68 $\pm$ 0.42
Vim-S + ECI	96.42 $\pm$ 0.35	96.08 $\pm$ 0.41	95.92 $\pm$ 0.46	95.98 $\pm$ 0.38
Vim-S + KGMS	96.38 $\pm$ 0.38	95.98 $\pm$ 0.44	96.08 $\pm$ 0.49	96.02 $\pm$ 0.40
EATMamba	96.89 $\pm$ 0.24	96.56 $\pm$ 0.31	96.52 $\pm$ 0.30	96.26 $\pm$ 0.35

Table 8. Ablation study on the TLDD. Mean $\pm$ std over three runs. The best result for each metric is shown in **bold**.

Configuration	Acc (%)	Precision (%)	Recall (%)	F1 (%)
Vim-S baseline	91.06 $\pm$ 0.49	90.71 $\pm$ 0.55	90.55 $\pm$ 0.62	90.52 $\pm$ 0.50
Vim-S + ECI	91.62 $\pm$ 0.41	91.28 $\pm$ 0.48	91.05 $\pm$ 0.53	91.12 $\pm$ 0.44
Vim-S + KGMS	91.55 $\pm$ 0.44	91.18 $\pm$ 0.50	91.18 $\pm$ 0.56	91.15 $\pm$ 0.47
EATMamba	92.87 $\pm$ 0.38	92.15 $\pm$ 0.52	92.48 $\pm$ 0.48	91.92 $\pm$ 0.51

Tables 7 and 8 report the ablation results on the TLD and TLDD, respectively. Adding ECI to the Vim-S baseline improves performance by enhancing token-level recombination, suggesting that globally consistent disease cues can be better consolidated through crossover-style interaction. Adding KGMS also improves performance, indicating that memory-guided mutation and soft selection help refine sparse lesion representations. The full EATMamba achieves the best results on both datasets, demonstrating that ECI and KGMS provide complementary benefits. The gains are larger on TLDD, which further confirms that evolution-inspired token refinement is particularly useful under in-the-wild background variation and domain shift.

Table 9 compares different ETD insertion strategies on TLD.

Table 9. Effect of different ETD insertion strategies on TLD. Mean $\pm$ std over three runs. Except for the No-ETD baseline, all ETD insertion strategies use the same total number of ETD insertions (6), distributed differently across stages. The best result for each metric is shown in **bold**.

Insertion Strategy	Acc (%)	F1 (%)
No ETD (Vim-S baseline)	96.03 $\pm$ 0.47	95.68 $\pm$ 0.42
Shallow stages only	96.16 $\pm$ 0.39	95.99 $\pm$ 0.35
Middle stages only	96.38 $\pm$ 0.33	96.24 $\pm$ 0.31
Deep stages only	96.56 $\pm$ 0.28	96.42 $\pm$ 0.29
All stages uniformly	96.69 $\pm$ 0.30	96.48 $\pm$ 0.27
Proposed stage-wise	96.89 $\pm$ 0.24	96.26 $\pm$ 0.35

Table 9 analyzes the effect of different ETD insertion strategies. Inserting ETD only in shallow stages provides limited improvement, likely because early layers mainly encode low-level textures and excessive recombination may disturb local lesion patterns. In contrast, deeper insertion yields stronger gains because semantic token representations are more suitable for token-level recombination and knowledge-guided refinement. The proposed stage-wise strategy achieves the highest accuracy (96.89%) among all insertion configurations. A notable accuracy–F1 trade-off emerges: inserting ETD uniformly across all stages achieves a slightly lower accuracy (96.69%) but a higher macro-F1 (96.48% vs. 96.26%), suggesting that excessive recombination in shallow layers may impair fine-grained lesion detail preservation yet improve class balance. Although the proposed stage-wise insertion yields a slightly lower macro-F1 than uniform insertion, it achieves the highest accuracy. Since accuracy is selected as the primary metric in this classification setting, we retain the stage-wise allocation while acknowledging this accuracy–F1 trade-off. We additionally note that concentrating evolutionary refinement in deeper stages aligns better with the representational-semantics motivation and controls computational overhead more effectively.

Table 10 reports the validation-set sensitivity analysis of the memory bank size. Table 11 reports the validation-set sensitivity analysis of the ECI temperature.

Table 10. Validation-set sensitivity analysis of the memory bank size K in KGMS. The ECI temperature is fixed at $\tau = 2.0$. Results are reported as mean $\pm$ std over three independent runs. The best result for each metric is shown in **bold**, and the final default setting is marked with $\dagger$.

Memory Size K	Acc (%)	F1 (%)
4	96.58 $\pm$ 0.13	96.03 $\pm$ 0.14
8	96.71 $\pm$ 0.09	96.15 $\pm$ 0.12
16	96.84 $\pm$ 0.11	96.18 $\pm$ 0.10
32 †	97.02 $\pm$ 0.06	96.55 $\pm$ 0.10
64	96.74 $\pm$ 0.10	96.22 $\pm$ 0.11

Table 11. Validation-set sensitivity analysis of the temperature τ in ECI. The memory bank size is fixed at $K = 32$. Results are reported as mean $\pm$ std over three independent runs. The best result for each metric is shown in **bold**, and the final default setting is marked with $\dagger$.

Temperature τ	Acc (%)	F1 (%)
0.1	96.52 $\pm$ 0.14	96.25 $\pm$ 0.16
0.25	96.68 $\pm$ 0.10	96.48 $\pm$ 0.12
0.5	96.81 $\pm$ 0.12	96.62 $\pm$ 0.09
1.0	96.89 $\pm$ 0.07	96.26 $\pm$ 0.08
2.0 †	97.02 $\pm$ 0.06	96.55 $\pm$ 0.10

Sensitivity analysis discussion. All hyperparameter sensitivity analyses were conducted on the validation set only, while the held-out test set was used exclusively for final model evaluation. We first analyzed the memory bank size K with the ECI temperature fixed at $\tau = 2.0$ (Table 10). The results show that $K = 32$ achieves the best validation accuracy and macro-F1 among the tested memory sizes, indicating that a moderately sized memory bank is sufficient to provide useful disease-related prototypes without introducing unnecessary redundancy.

We then fixed $K = 32$ and varied the temperature τ to evaluate the sensitivity of ECI (Table 11). The results show a moderate accuracy–F1 trade-off. Specifically, $\tau = 2.0$ achieves the highest validation accuracy (97.02%), whereas $\tau = 0.5$ obtains the highest macro-F1 score (96.62%). Compared with the best-F1 configuration ($\tau = 0.5$), $\tau = 2.0$ improves accuracy by 0.21 percentage points while reducing macro-F1 by only 0.07 percentage points. Therefore, we retain $\tau = 2.0$ as the final temperature setting because it provides the best validation accuracy while maintaining a competitive macro-F1 score.

Overall, the performance differences among $K \in \{16, 32, 64\}$ and $\tau \in \{0.5, 1.0, 2.0\}$ are relatively small, suggesting that EATMamba is not highly sensitive to these hyperparameters within reasonable ranges. The final test-set performance is reported separately in Tables 4 and 5.

4.6. Cross-Dataset Generalization

Because TLD contains one additional class (Powdery Mildew) that is absent from TLDD, cross-dataset generalization experiments were conducted on the intersection of the two label spaces. Specifically, the Powdery Mildew samples in TLD were excluded during cross-dataset training/testing, resulting in a shared 10-class classification protocol. The classifier head was reconfigured accordingly for the cross-dataset setting.

To further evaluate robustness under domain shift, we conduct cross-dataset generalization experiments. Models are trained on one dataset and directly evaluated on the other dataset without fine-tuning. This setting is more challenging than the standard within-dataset split because the source and target domains differ in image resolution, acquisition environment, background complexity, and illumination conditions [57].

As shown in Table 12, all models experience substantial performance degradation under cross-dataset evaluation, confirming the domain shift between controlled and in-the-wild acquisition conditions, which is a well-documented challenge in agricultural image analysis [58]. When trained on TLD (512×512 , controlled) and tested on TLDD (256×256 , in-the-wild), EATMamba achieves 76.63% accuracy and 77.14% F1, outperforming Vim by 5.68 and 5.54 percentage points, respectively. The reverse direction (TLDD $\rightarrow$ TLD) is more challenging, as the model must generalize from low-resolution in-the-wild images to high-resolution controlled images with different class distributions [59]; EATMamba still leads with 75.02% accuracy and 78.34% F1, surpassing Vim by 4.80 and 6.00 points. Cross-continental crop classification studies have similarly shown the importance of domain adaptation strategies for handling distribution shifts across diverse acquisition conditions [60]. These consistent gains suggest that evolution-inspired token refinement improves the transferability of disease-related representations by suppressing domain-specific background responses and emphasizing more stable lesion cues that persist across acquisition conditions [61]. Furthermore, recent studies have shown that deep architectures may fail to generalize across agricultural domains, highlighting the need for lightweight alternatives and robust feature learning strategies for hyperspectral and RGB-based crop monitoring [62].

Table 12. Cross-dataset generalization results on the shared 10-class label space. Models are trained on one dataset and directly tested on the other without fine-tuning. For evaluations involving TLD, the Powdery Mildew class is excluded to ensure label-space consistency with TLDD.

Train Set	Test Set	Method	Acc (%)	F1 (%)
TLD	TLDD	Vim	70.95	71.60
TLD	TLDD	VMamba	73.18	74.37
TLD	TLDD	EATMamba	76.63	77.14
TLDD	TLD	Vim	70.22	72.34
TLDD	TLD	VMamba	72.68	74.90
TLDD	TLD	EATMamba	75.02	78.34

4.7. Grad-CAM Visualization Analysis

Grad-CAM visualizations are used to qualitatively analyze whether different models focus on disease-relevant regions. As shown in Figure 8, baseline models sometimes activate background regions, leaf edges, or non-discriminative textures, especially under cluttered in-the-wild scenes. In contrast, EATMamba produces more concentrated responses around lesion spots and disease-relevant leaf regions. This indicates that the proposed evolution-inspired token refinement modules help suppress irrelevant background responses and enhance disease-discriminative visual evidence. The visualization results are consistent with the quantitative improvements on TLDD, where background variation and domain shift are more pronounced.

From an agronomic perspective, the highlighted regions correspond to important visual symptoms used in tomato disease diagnosis. For Early Blight, the activated areas mainly cover necrotic lesions and concentric-ring patterns. For Late Blight, the responses are associated with irregular water-soaked and expanding necrotic regions. For Bacterial Spot and Septoria Leaf Spot, the model focuses more on small spot-like lesions, dark margins, and surrounding chlorotic halos.

These observations indicate that EATMamba does not merely improve classification accuracy, but also enhances attention toward biologically meaningful symptom regions.

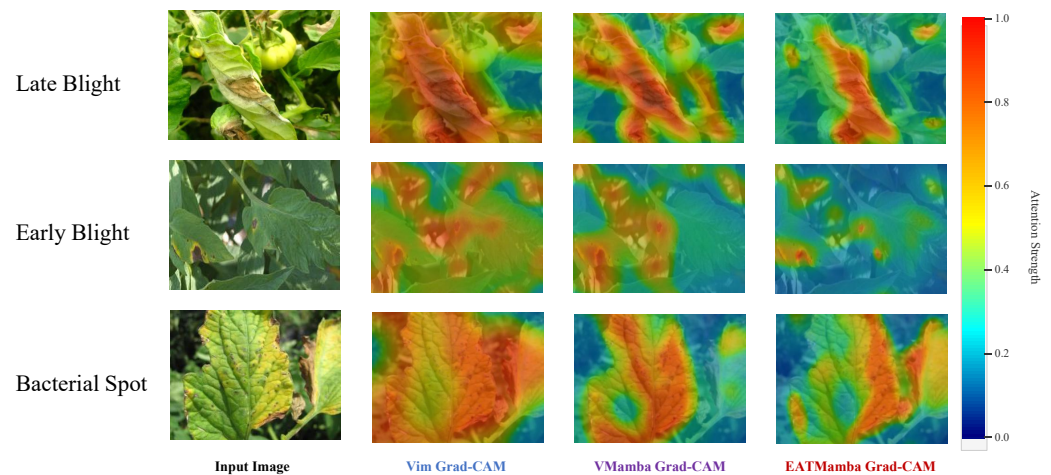


Figure 8. Grad-CAM visualization comparison among Vim, VMamba, and EATMamba. EATMamba focuses more consistently on disease lesions and pest-damaged regions, while baseline models may attend to background clutter or non-discriminative leaf areas.

Figure 9 presents the confusion matrices of EATMamba. Most categories are correctly classified with high diagonal responses. The remaining errors mainly occur between visually similar diseases, such as early blight and late blight, or bacterial spot and septoria leaf spot. These confusion patterns identify the main remaining errors of EATMamba and are consistent with the subtle inter-class differences in lesion shape, color, and texture. Together with the per-class F1 comparison in Table 13, where EATMamba achieves higher F1-scores than Vim-S and VMamba-T on visually ambiguous categories, the results suggest improved fine-grained feature discrimination.

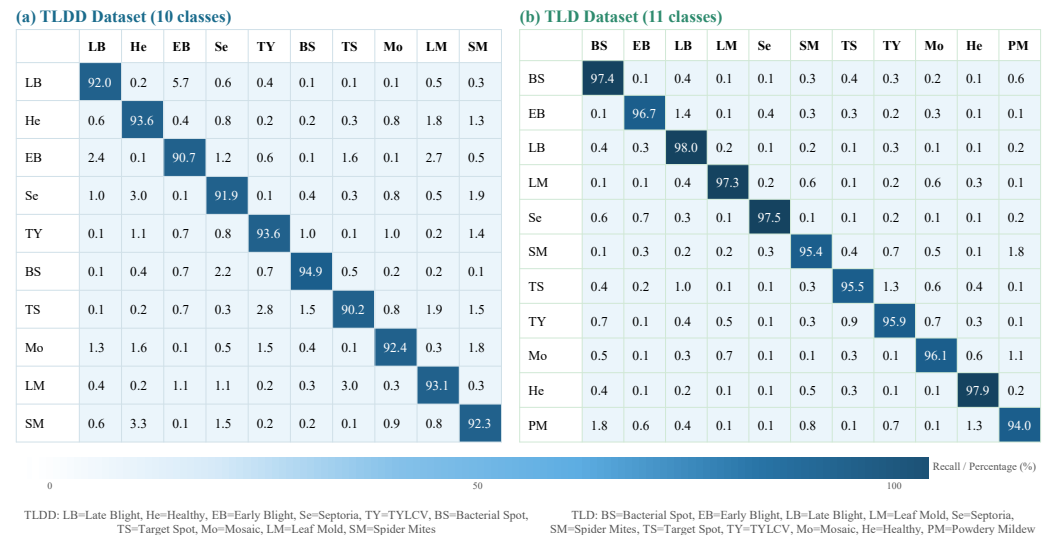


Figure 9. Row-normalized confusion matrices of EATMamba on the TLDD and TLD test sets. Values are reported as percentages and averaged over three independent runs. Diagonal entries indicate per-class recall; the unweighted diagonal mean corresponds to macro-Recall. Overall accuracy is computed from raw sample-level predictions in Tables 4 and 5, and may differ from the unweighted diagonal mean due to class imbalance and averaging across seeds. Darker colors indicate higher values.

Table 13. Per-class F1-score comparison on TLDD. Values are averaged over three independent runs using the same held-out test split as Table 5. The macro-average row is shown in **bold**.

Class	Vim-S	VMamba-T	EATMamba
Late Blight	90.70	91.20	91.47
Healthy	91.90	92.20	93.05
Early Blight	88.40	89.30	90.18
Septoria Leaf Spot	89.90	90.80	91.38
Tomato Yellow Leaf Curl Virus	91.70	92.20	93.02
Bacterial Spot	90.50	93.20	94.31
Target Spot	88.50	88.90	89.65
Tomato Mosaic Virus	90.30	91.10	91.85
Leaf Mold	92.20	92.55	92.58
Spider Mites	91.10	91.35	91.73
Macro-Average	90.52	91.28	91.92

Table 13 provides a class-level comparison on TLDD. The results are reported at the class level to show that overall accuracy alone may hide important differences among disease categories. Healthy, Tomato Yellow Leaf Curl Virus, Leaf Mold, and Bacterial Spot obtain relatively higher F1-scores because their visual patterns are more distinctive or are better captured by the learned representations. In contrast, Early Blight, Septoria Leaf Spot, and Target Spot remain more difficult to distinguish because they often share small necrotic lesions, chlorotic halos, and irregular spot boundaries. Compared with Vim-S and VMamba-T, EATMamba achieves consistent improvements across all categories, with particularly notable gains on visually ambiguous or small-lesion classes such as Early Blight, Septoria Leaf Spot, Bacterial Spot, Target Spot, and Tomato Mosaic Virus. This indicates that adaptive token refinement provides greater benefit when discriminative symptoms are sparse, subtle, and easily confused with background textures.

The per-class F1 values in Table 13 are averaged over three independent runs using the same TLDD test split as Table 5. The macro-average values are computed as the unweighted mean of the ten per-class F1 scores and are therefore consistent with the macro-F1 results reported in Table 5: 90.52% for Vim-S, 91.28% for VMamba-T, and 91.92% for EATMamba. Minor differences may only arise from rounding to two decimal places.

4.8. Feature Distribution Visualization

To further examine the discriminative ability of learned representations, we visualize the final class-token features using t-SNE [63]. This visualization is intended to compare the class separability and intra-class compactness of representations learned by different models. As shown in Figure 10, Vim-S and VMamba-T produce partially overlapped clusters for visually similar disease categories, whereas EATMamba yields more compact intra-class distributions and clearer inter-class separation. This observation suggests that evolution-inspired token refinement improves feature discriminability for fine-grained tomato disease recognition.

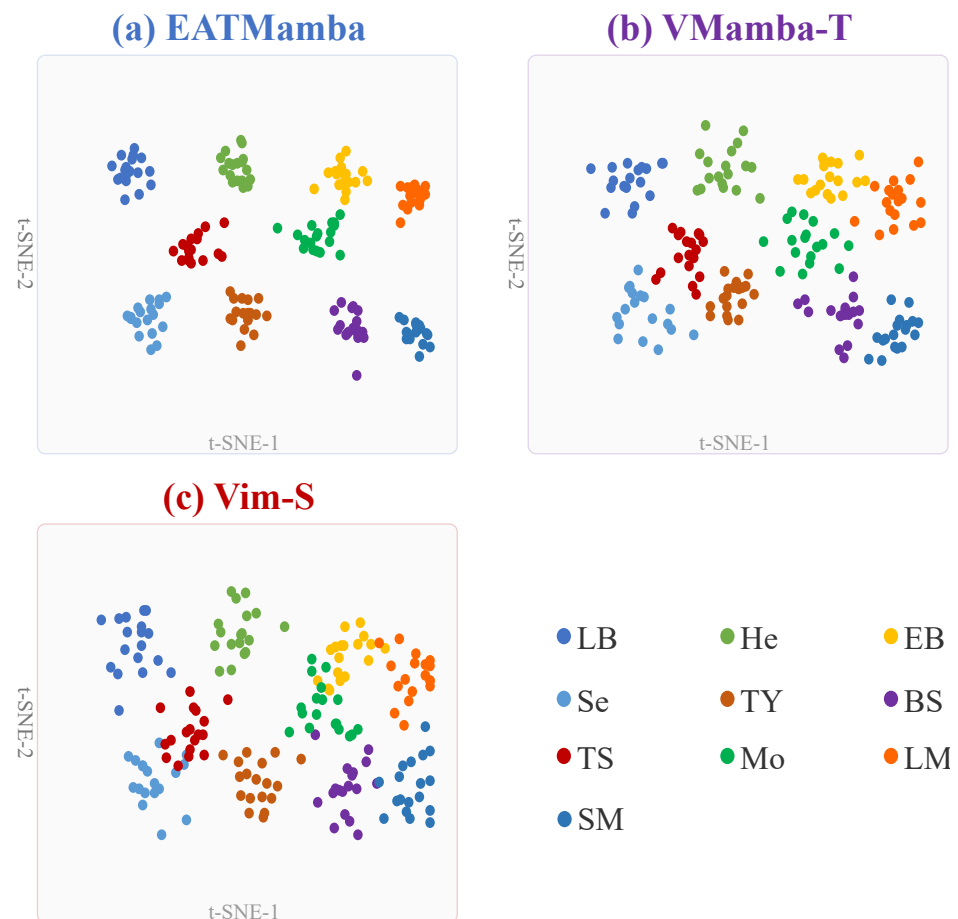


Figure 10. t-SNE visualization of final class-token embeddings on the TLDD test set. The visualization is used only for post-hoc qualitative analysis and is not involved in model training or quantitative evaluation. Each point denotes one test image, and colors denote different disease or pest categories. The three subplots correspond to (a) EATMamba, (b) VMamba-T, and (c) Vim-S. The axes t-SNE-1 and t-SNE-2 denote the two-dimensional nonlinear projection space and have no physical meaning. Class abbreviations: LB = Late Blight, He = Healthy, EB = Early Blight, Se = Septoria Leaf Spot, TY = Tomato Yellow Leaf Curl Virus, BS = Bacterial Spot, TS = Target Spot, Mo = Tomato Mosaic Virus, LM = Leaf Mold, and SM = Spider Mites.

5. Discussion

5.1. Why Evolution-Inspired Token Refinement Helps

The experimental results demonstrate that EATMamba consistently improves over the Vim-S backbone and other representative baselines. The improvement can be attributed to the complementary roles of ECI and KGMS. ECI performs token-level recombination, allowing spatially separated but semantically related disease cues to interact and reinforce each other. This is important for tomato leaves, where symptoms may appear as multiple scattered spots or irregular lesion regions. KGMS further introduces memory-guided candidate generation and differentiable soft selection, enabling the model to generate candidate refinements and selectively preserve useful changes.

These results suggest that the benefit of EATMamba does not only come from using a stronger backbone, but also from explicitly refining token-level evidence under noisy acquisition conditions. In RGB leaf images, disease-relevant regions are often sparse, weak, or partially corrupted by background clutter. The generate–recombine–select style refinement provides an inductive bias for strengthening discriminative tokens while suppressing nuisance responses. The ablation results confirm that both modules contribute to performance gains, while their combination produces the best results.

5.2. Relation to Attention, Gating, and Memory Mechanisms

It is natural to ask whether ECI and KGMS provide benefits beyond conventional attention, gating, or memory-based refinement. ECI shares a soft affinity computation with self-attention, since both compute projected queries and keys and derive token-wise weights. However, the functional interpretation differs. Standard self-attention directly aggregates contextual information for each token, whereas ECI formulates the affinity matrix as a crossover-inspired recombination distribution and fuses the original and recombined token representations.

KGMS is also related to memory retrieval and gating mechanisms. The memory bank provides prototype guidance, but KGMS does not directly replace tokens with retrieved memory features. Instead, the retrieved representation is used to guide mutation-inspired candidate generation, and a soft selection gate adaptively fuses the original and candidate representations. Therefore, KGMS can be viewed as a memory-guided candidate-generation and soft-selection module rather than a direct memory retrieval block.

The proposed modules are fully differentiable and optimized by standard backpropagation. They do not perform population-level stochastic search, non-gradient-based fitness evaluation, or discrete evolutionary selection. Thus, the contribution of EATMamba should be interpreted as an evolution-inspired neural refinement design, not as a classical evolutionary algorithm. More strictly matched comparisons with equally parameterized attention, gating, and memory-only variants would further clarify the source of improvement, and we leave such controlled architectural studies to future work.

5.3. Practical Necessity Under Visible-Symptom Conditions

This study focuses on RGB-based visible-symptom disease and pest classification rather than pre-symptomatic disease detection. The presence of visible symptoms, however, does not mean that accurate diagnosis is straightforward. Non-expert observers may notice that a leaf is abnormal, but fine-grained identification of the specific disease or pest remains challenging when different categories share similar visual cues. For example, Early Blight and Late Blight may both show necrotic regions and irregular discoloration, while Bacterial Spot and Septoria Leaf Spot may both present small spot-like lesions and chlorotic halos.

Therefore, the practical value of EATMamba lies not in simply detecting whether a leaf is diseased, but in providing scalable, consistent, and fine-grained classification under field conditions. Such a model can support non-expert farmers, reduce subjective manual diagnosis, and enable large-scale monitoring where continuous expert inspection is impractical.

5.4. Robustness Under Complex Acquisition Conditions

The performance gain of EATMamba is more pronounced on TLDD than on TLD, indicating that the proposed method is particularly effective under uncontrolled farm conditions. TLDD contains stronger illumination variation, background clutter, viewpoint changes, and scale differences. Under such conditions, disease-related tokens may be weakened by noisy background responses. ECI helps consolidate globally consistent symptom cues, while KGMS adaptively selects memory-guided refinements that are more likely to correspond to disease-relevant evidence. The Grad-CAM visualizations further support this interpretation, showing that EATMamba attends more consistently to lesion and pest regions rather than background textures. Therefore, the proposed evolution-inspired token refinement strategy improves not only classification accuracy but also localization behavior and robustness under domain shift.

5.5. Efficiency and Practical Implications

EATMamba introduces additional token refinement operations on top of the Vim-S backbone. It is important to note that, while the Vim-S backbone itself processes tokens with linear complexity in sequence length, the ECI module introduces a quadratic $O(M^2D)$ term within each inserted ETD block. The proposed framework controls this overhead through three mechanisms: (i) sparse stage-wise insertion, with more ETD operations placed in deeper stages; (ii) a reduced projection dimension ($d = 64$) that lowers the constant factor in the quadratic term; and (iii) a small number of total ETD insertions ($K_s = [1, 1, 2, 2]$ across four stages). This design does not preserve the asymptotic linear complexity of the backbone, but it keeps the practical overhead moderate: the parameter count increases by 27.4% and FLOPs (measured at 224×224) increase by 20.8% relative to Vim-S. We report FLOPs at 224×224 following the standard fvcore convention for cross-model comparison; readers should note that actual FLOPs at higher resolutions (e.g., 512×512 used for TLD training) will differ. The efficiency comparison shows that EATMamba achieves a favorable accuracy–efficiency trade-off compared with Transformer-based baselines, though for ultra-high-resolution deployment, the quadratic ECI cost may require further mitigation through low-rank or sparse token-recombination approximations.

5.6. RGB-Only Imaging and Hyperspectral Extension

A limitation of this study is that it only considers RGB images. Hyperspectral and multispectral imaging can capture physiological and biochemical changes before visible symptoms appear, and may therefore be more suitable for pre-symptomatic or early-stage disease detection. However, such sensors are more expensive, require more complex acquisition and calibration procedures, and are less accessible for low-cost mobile or farm-scale deployment.

The goal of this work is therefore different: we focus on practical RGB-based visible-symptom classification using low-cost imaging devices. This setting is more compatible with scalable field monitoring, but it also limits the ability of the model to detect diseases before visible symptoms emerge. Future work will investigate whether the proposed evolution-inspired token refinement mechanism can be extended to multispectral or hyperspectral data.

5.7. Limitations and Future Work

Despite its effectiveness, EATMamba still has several limitations that should be considered when interpreting the results. A formal convergence or stability proof for the full nonlinear network is beyond the scope of this study, but the row-stochastic recombination distribution, residual fusion, and bounded soft selection provide an interpretable and controlled formulation. First, the ECI module introduces quadratic $O(M^2D)$ token interaction within each inserted ETD block, which means the overall framework does not retain the asymptotic linear complexity of the Vim-S backbone. Although the overhead is controlled by sparse stage-wise insertion and reduced projection dimension, future work may explore low-rank, sparse, or local–global token-recombination distributions to further improve scalability for ultra-high-resolution agricultural images. Second, the current study focuses on image-level classification and does not explicitly localize disease regions with pixel-level annotations. Incorporating weakly supervised localization or segmentation supervision may further improve interpretability and practical diagnosis [64]. Third, although two complementary tomato datasets are used, broader validation across different crops, sensors, seasons, and geographic regions is still needed to evaluate real-world generalization. Additionally, the class imbalance present in both datasets may affect model performance, and future work should investigate more specialized imbalanced learning strategies [65].

Finally, future work will investigate lightweight deployment on mobile and edge devices for real-time crop disease monitoring in resource-constrained agricultural environments.

6. Conclusions

In this paper, we proposed EATMamba, an effective evolution-inspired Vision Mamba framework for tomato leaf disease and pest classification. The proposed model is formulated as a differentiable token-refinement framework that combines efficient long-range modeling from state-space vision backbones with explicit token-level evidence refinement. By integrating the Evolutionary Crossover–Interaction module and the Knowledge-Guided Mutation–Selection module, EATMamba performs crossover-inspired token recombination, memory-guided candidate generation, and soft selection to enhance disease-relevant cues while suppressing background distractions. Experiments on the high-resolution TLD and the in-the-wild TLDD show that EATMamba achieves 96.89% and 92.87% accuracy, respectively, outperforming representative CNN-, Transformer-, and SSM-based baselines, including ConvNeXt, MambaVision, and EfficientVMamba. Ablation studies verify the complementary contributions of ECI and KGMS, while visualization analyses show that the proposed method focuses more consistently on disease and pest regions. These results demonstrate the potential of evolution-inspired differentiable token refinement for robust agricultural image recognition. Future work will explore multispectral and hyperspectral extensions, broader cross-region and cross-environment validation, more efficient token-recombination strategies, lightweight deployment on mobile and edge devices, and additional expert-validated datasets for real-world crop disease monitoring.

Author Contributions: Conceptualization, Y.H. (Yingbiao Hu), H.L. and W.K.; methodology, Y.H. (Yingbiao Hu), Y.H. (Yu He), Z.P. and W.K.; software, Y.H. (Yingbiao Hu); validation, Y.H. (Yingbiao Hu); formal analysis, Y.H. (Yingbiao Hu); investigation, Y.H. (Yingbiao Hu); resources, N.C., C.Y. and W.K.; data curation, Y.H. (Yingbiao Hu), N.C. and C.Y.; writing—original draft, Y.H. (Yingbiao Hu); writing—review & editing, Y.H. (Yingbiao Hu), H.L., Y.H. (Yu He) and W.K.; visualization, Y.H. (Yingbiao Hu), H.L. and Y.H. (Yu He); supervision, Z.P. and W.K.; project administration, W.K.; funding acquisition, W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Macao Polytechnic University under internal project number fca.f7f2.06fd.0.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the ongoing follow-up research of this project.

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

References

1. Jelali, M. Deep learning networks-based tomato disease and pest detection: A first review of research studies using real field datasets. *Front. Plant Sci.* **2024**, *15*, 1493322. [[CrossRef](#)] [[PubMed](#)]
2. Arshad, Z.; Javed, A.; Saudagar, A.K.J. ConvGeM-next: A deep learning framework for plant disease detection. *Front. Plant Sci.* **2026**, *17*, 1763739. [[CrossRef](#)] [[PubMed](#)]
3. Jiang, X.; Wang, J.; Xie, K.; Cui, C.; Du, A.; Shi, X.; Yang, W.; Zhai, R. PlantCaFo: An efficient few-shot plant disease recognition method based on foundation models. *Plant Phenomics* **2025**, *7*, 100024. [[CrossRef](#)] [[PubMed](#)]
4. Sreedevi, A.; Srinivas, K. Classification of Tomato Leaf Disease Using Ensemble Learning. In *Impending Inquisitions in Humanities and Sciences*; Varkolu, M., Mallesh, M.P., Oruganti, K.P., Eds.; CRC Press: Boca Raton, FL, USA, 2024; pp. 289–294. [[CrossRef](#)]
5. Benabbas, W.; Brahimi, M.; Akhrout, S.; Fortas, B. Rethinking Plant Disease Diagnosis: Bridging the Academic-Practical Gap with Vision Transformers and Zero-Shot Learning. *arXiv* **2025**. [[CrossRef](#)]
6. Devarajan, D.; Allafi, R.; Obayya, M.; Nemri, N. AI based real time disease diagnosis in plants using deep learning driven CNNs. *Sci. Rep.* **2026**, *16*, 4587. [[CrossRef](#)] [[PubMed](#)]

7. Souibgui, M.A.; Kessentini, Y. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1180–1191. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Kumar, T.; Brennan, R.; Mileo, A.; Bendeache, M. Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access* **2024**, *12*, 187536–187571. [\[CrossRef\]](#)
9. Xiang, J.; Ge, J.; Fu, C.; Huang, J.; Lu, M.; Zhang, Z.; Song, Y.; Yan, S. An environment-guided visual–temporal deep learning framework for early disease detection in greenhouse horticultural crops. *Front. Plant Sci.* **2026**, *17*, 1796407. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Liu, Z.; Zhang, Z.; Li, S. Precise Identification of Tomato Leaf Diseases: A VMamba-FCS Classification Model Based on Multi-Mechanism Synergistic Enhancement. *Agriculture* **2026**, *16*, 875. [\[CrossRef\]](#)
11. Ali, S.H.; Naeem, M.; Ali, H. Tomato Leaf Disease Dataset (6 Classes), 2026. Available online: <https://www.kaggle.com/datasets/syedhashirali260/tomato-leaf-disease-dataset-6-classes> (accessed on 21 July 2026).
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778. [\[CrossRef\]](#)
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2021. Available online: <https://openreview.net/forum?id=YicbFdNTTy> (accessed on 21 July 2026).
14. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE Computer Society: Los Alamitos, CA, USA, 2021; pp. 10012–10022. [\[CrossRef\]](#)
15. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE Computer Society: Los Alamitos, CA, USA, 2022; pp. 11976–11986. [\[CrossRef\]](#)
16. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Proceedings of the First Conference on Language Modeling (COLM)*, Philadelphia, PA, USA, 7–9 October 2024. Available online: <https://openreview.net/forum?id=tEYskw1VY2> (accessed on 21 July 2026).
17. Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*; *Proceedings of Machine Learning Research*: Cambridge, MA, USA, 2024; Volume 235, pp. 62429–62442. Available online: <https://proceedings.mlr.press/v235/zhu24f.html> (accessed on 21 July 2026).
18. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; Liu, Y. VMamba: Visual State Space Model. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2024. [\[CrossRef\]](#)
19. Zhang, J.; Li, X.; Wang, Y.; Wang, C.; Yang, Y.; Liu, Y.; Tao, D. EATFormer: Improving Vision Transformer Inspired by Evolutionary Algorithm. *Int. J. Comput. Vis.* **2024**, *132*, 3509–3536. [\[CrossRef\]](#)
20. Lange, R.; Tian, Y.; Tang, Y. Evolution Transformer: In-Context Evolutionary Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion; GECCO'24 Companion*; ACM: New York, NY, USA, 2024; pp. 575–578. [\[CrossRef\]](#)
21. Huang, Y.; Huang, T.; Du, J.; Qiu, J.; Liu, C.; Wan, F.; Qian, W.; Qiao, X.; Wang, L. DenseNet-CSL: An Enhanced Network for Multi-Class Recognition of Agricultural Pests, Weeds, and Crop Diseases. *Agriculture* **2026**, *16*, 394. [\[CrossRef\]](#)
22. Sun, C.; Liu, J.; He, B.; Huang, L.; Lv, L. SEAFEC: A spatial–edge adaptive convolution for multi-scale and boundary-aware plant disease and weed imagery. *Front. Plant Sci.* **2026**, *16*, 1695076. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Nguyen, T.T.; Nguyen, T.P. Efficient tick-shape networks of full-residual point-depth-point blocks for image classification. *Neurocomputing* **2024**, *596*, 127942. [\[CrossRef\]](#)
24. Alsubai, S.; Almadhor, A.; Al Hejaili, A.; Gadekallu, T.R. Intelligent smart sensing with ResNet-PCA and hybrid ML–DNN for sustainable and accurate plant disease detection. *Front. Plant Sci.* **2025**, *16*, 1691415. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Lv, X.; Wang, S.; Ye, D. CFNet: LiDAR-Camera Registration Using Calibration Flow Network. *Sensors* **2021**, *21*, 8112. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Cheng, Z.; Wu, Y.; Li, Y.; Cai, L.; Ihnaini, B. A Comprehensive Review of Explainable Artificial Intelligence (XAI) in Computer Vision. *Sensors* **2025**, *25*, 4166. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Felicita, S.A.M.; Kavitha, B.R. Enhancing multiclass plant disease classification using GAN-boosted vision transformer with XAI insights. *Front. Plant Sci.* **2026**, *16*, 1649399. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Gao, Z.; Lu, J.; Zhao, L.; Li, Q.; Chen, N.; Li, P. PCT-ViT: A vision transformer incorporating fine-grained perception enhancement and counterfactual token selection. *Digit. Signal Process.* **2026**, *170*, 105796. [\[CrossRef\]](#)
29. Roberts, A.G.; John, M.; Zhang, J.; Romano, D.; Sisman, M.; Choi, K.S.; Kim, H.; Sabuncu, M.R.; Nguyen, T.D.; Dimov, A.V.; et al. Spectral Vision Transformer for Efficient Tokenization with Limited Data. *arXiv* **2026**. [\[CrossRef\]](#)
30. Huang, Y.; Hechen, Z.; Zhou, M.; Li, Z.; Kwong, S. An Attention-Locating Algorithm for Eliminating Background Effects in Fine-Grained Visual Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 5993–6006. [\[CrossRef\]](#)

31. Li, Y.; Wang, H.; Luo, H. Semantic Feature Integration Network for Fine-Grained Visual Classification. 2023. Available online: <https://www.researchsquare.com/article/rs-3220012/v1> (accessed on 21 July 2026).
32. Yang, H.; Zhang, Q.; Hu, J.F.; Lai, J. A Training-Free Framework for Text-to-Image Person Re-Identification via Query-Prototype Matching. *Pattern Recognit.* **2026**, *179*, 113705. [CrossRef]
33. Han, K.; Hua, W.; He, L. What's next in agri-vision: A review of Vision Transformers and Vision Mamba in crop management. *Comput. Electron. Agric.* **2026**, *245*, 111551. [CrossRef]
34. Gu, A.; Goel, K.; Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces. In *Proceedings of the International Conference on Learning Representations (ICLR)*; OpenReview.net: Amherst, MA, USA, 2022. Available online: <https://openreview.net/forum?id=uYLFoz1v1AC> (accessed on 21 July 2026).
35. Yang, C.; Chen, Z.; Espinosa, M.; Ericsson, L.; Wang, Z.; Liu, J.; Crowley, E.J. PlainMamba: Improving Non-Hierarchical Mamba in Visual Recognition. In *Proceedings of the 35th British Machine Vision Conference (BMVC)*, Glasgow, UK, 25–28 November 2024. Available online: https://bmva-archive.org.uk/bmvc/2024/papers/Paper_755/paper.pdf (accessed on 21 July 2026).
36. Hatamizadeh, A.; Kautz, J. MambaVision: A Hybrid Mamba-Transformer Vision Backbone. In *Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2025; pp. 25261–25270. [CrossRef]
37. Huang, T.; Pei, X.; You, S.; Wang, F.; Qian, C.; Xu, C. LocalMamba: Visual State Space Model with Windowed Selective Scan. In *Computer Vision—ECCV 2024 Workshops*; Del Bue, A., Canton, C., Pont-Tuset, J., Tommasi, T., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15633, pp. 12–22. [CrossRef]
38. Pei, X.; Huang, T.; Xu, C. EfficientVMamba: Atrous Selective Scan for Light Weight Visual Mamba. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 6443–6451. [CrossRef]
39. Yu, W.; Wang, X. MambaOut: Do We Really Need Mamba for Vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE Computer Society: Los Alamitos, CA, USA, 2025; pp. 4484–4496. [CrossRef]
40. Suleman, H.; Wasim, S.T.; Naseer, M.; Gall, J. Distillation-free Scaling of Large State-Space Models for Images and Videos. *Int. J. Comput. Vis.* **2026**, *134*, 232. [CrossRef] [PubMed]
41. Zhu, X.; Liu, J.; Cao, J.; Wang, B. MambaML: Exploring State Space Models for Multi-Label Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE Computer Society: Los Alamitos, CA, USA, 2025; pp. 4743–4753. [CrossRef]
42. Zhang, J.; Nguyen, A.T.; Han, X.; Trinh, V.Q.H.; Qin, H.; Samaras, D.; Hosseini, M.S. 2DMamba: Efficient State Space Model for Image Representation with Applications on Giga-Pixel Whole Slide Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE Computer Society: Los Alamitos, CA, USA, 2025; pp. 3583–3592. [CrossRef]
43. Wang, S.; Zeng, Q.; Yuan, G.; Ni, W.; Li, C.; Duan, H.; Xie, N.; Xiao, F.; Yang, X. Generalized zero-shot pest and disease image classification based on causal gating model. *Comput. Electron. Agric.* **2025**, *230*, 109827. [CrossRef]
44. Tang, J.; Hu, J.F.; Liang, T.; Lin, X.; Sun, J.; Zheng, W.S.; Lai, J. Human Motion Prediction via Continual Prior Compensation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 5131–5146. [CrossRef] [PubMed]
45. Mamun, A.A.; Zhang, M.; Ahmedt-Aristizabal, D.; Hayder, Z.; Awrangzeb, M. ConMamba: Contrastive vision Mamba for plant disease detection. *Pattern Recognit.* **2026**, *176*, 113177. [CrossRef]
46. Stanley, K.O.; Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. *Evol. Comput.* **2002**, *10*, 99–127. [CrossRef] [PubMed]
47. Song, X.; Lv, Z.; Fan, J.; Xiong, D.; Lv, J.; Liu, J.; Sun, Y. Evolutionary Multiobjective Spiking Neural Architecture Search for Image Classification. *IEEE Trans. Evol. Comput.* **2026**, *30*, 106–120. [CrossRef]
48. Stanley, K.O.; Clune, J.; Lehman, J.; Miikkulainen, R. Designing Neural Networks through Neuroevolution. *Nat. Mach. Intell.* **2019**, *1*, 24–35. [CrossRef]
49. Ma, L.; Cui, J.; Liu, X.; Yang, B. Neural Architecture Search with Graph Evolutionary. *SSRN* **2024**. Available online: <https://ssrn.com/abstract=4822229> (accessed on 21 July 2026). [CrossRef]
50. Ghosh, D.; Ghosh, S.; Jana, N.D.; Mallipeddi, R. Intelligent neural architecture search via Taguchi design and language model-based differential evolution for agricultural image recognition. *Expert Syst. Appl.* **2026**, *309*, 131135. [CrossRef]
51. Feng, Y.; Lv, Z.; Chen, H.; Gao, S.; An, F.; Sun, Y. LRNAS: Differentiable Searching for Adversarially Robust Lightweight Neural Architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 5629–5643. [CrossRef] [PubMed]
52. Jalil, H.; Qazi, A.; Iqbal, A. NeuRN: Neuro-inspired Domain Generalization for Image Classification. *arXiv* **2025**. [CrossRef]
53. Mondal, S.; Chowdhury, R.; Debnath, A. Machine Learning Models for Early Detection of Plant Diseases Using Multispectral Imaging: A Comprehensive Review. *Public Health—Open J.* **2026**, *11*, 353–362.
54. Hu, Y.; Li, H.; Yang, C.; Chen, N.; Pan, Z.; Ke, W. Challenges and Opportunities in Tomato Leaf Disease Detection with Limited and Multimodal Data: A Review. *Mathematics* **2026**, *14*, 422. [CrossRef]
55. Li, J.; Feng, Q.; Yang, J.; Zhang, J.; Yang, S. Few-shot crop disease recognition using sequence-weighted ensemble model-agnostic meta-learning. *Front. Plant Sci.* **2025**, *16*, 1615873. [CrossRef] [PubMed]

56. Yuan, X.; Zhang, Z.; Xu, X.; Wang, Z.; Lin, C.W. From Calibration to Refinement: Seeking Certainty via Probabilistic Evidence Propagation for Noisy-Label Person Re-Identification. *arXiv* **2026**. [[CrossRef](#)]
57. Wu, Z.; Guo, J.; Hou, W.; Zhou, K.; Cao, K.; Jung, H. Unseen-Crop Plant Disease Classification via Disentangled Representation Learning. *Electronics* **2026**, *15*, 1553. [[CrossRef](#)]
58. Hu, X.; Chen, S.; Duan, Q.; Ahn, C.K.; Shang, H.; Zhang, D. Domain Adaptation for Big Data in Agricultural Image Analysis: A Comprehensive Review. *arXiv* **2025**. [[CrossRef](#)]
59. Wang, Q.; Bu, P.; Breckon, T.P. Unifying Unsupervised Domain Adaptation and Zero-Shot Visual Recognition. *arXiv* **2019**. [[CrossRef](#)]
60. Li, S.; Liu, L.; Huo, J.; Li, S.; Yin, Y.; Ma, Y. A Class-Aware Unsupervised Domain Adaptation Framework for Cross-Continental Crop Classification with Sentinel-2 Time Series. *Remote Sens.* **2025**, *17*, 3762. [[CrossRef](#)]
61. Ghafar, A.; Chen, C.; Ahmad, I.; Qasim, M.; Ahmed, S.; Ahmed Rajput, F.; Zulfikar, U.; Malik, T. Deep transfer learning for comprehensive diagnosis of cotton leaf pathologies. *Microb. Pathog.* **2026**, *211*, 108257. [[CrossRef](#)] [[PubMed](#)]
62. Pankajakshan, P.; Padmasanan, A.; Sundar, S. Deep Architectures Fail to Generalize: A Lightweight Alternative for Agricultural Domain Transfer in Hyperspectral Images. *Sensors* **2026**, *26*, 174. [[CrossRef](#)] [[PubMed](#)]
63. Wong, K.Y.; Chung, F.I. Visualizing Time Series Data with Temporal Matching Based t-SNE. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8. [[CrossRef](#)]
64. Zhang, B.; Zhang, Z.; Zhang, X. YOLOv11-4ConvNeXtV2: Enhancing Persimmon Ripeness Detection Under Visual Challenges. *AI* **2025**, *6*, 284. [[CrossRef](#)]
65. Chen, W.; Yang, K.; Yu, Z.; Shi, Y.; Chen, C.L.P. A survey on imbalanced learning: Latest research, applications and future directions. *Artif. Intell. Rev.* **2024**, *57*, 137. [[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.