Journal of Computer Science

Research Article

An Edge Efficient and Lightweight Hybrid CNN Vision Transformer Network With CBAM for IoT Based Plant Disease Detection

Nyabeye Pangop Doris-Khöler and Kengni Ngangmo Olga

Department of Information and Communication Technologies, Faculty of Sciences, University of Ebolowa, Ebolowa, Cameroon

Article history
Received: 11-12-2025
Revised: 22-05-2026
Accepted: 30-07-2026

Corresponding Author:
Nyabeye Pangop Doris-Khöler
Department of Information and Communication Technologies, Faculty of Sciences, University of Ebolowa, Ebolowa, Cameroon
Email: nyabeyedoriskholer@gmail.com

Abstract: Detecting crop disease early enough to act on it remains a bottleneck for smallholder and precision farming alike, particularly when diagnosis depends on manual inspection or lab analysis that cannot keep pace with an outbreak. We present AgriViTNet, a compact CNN-Vision Transformer hybrid built for exactly this constraint: It extracts local texture cues through convolution, relates them across the whole leaf surface through a single-layer Transformer encoder, and sharpens the resulting features toward disease-relevant regions through a Convolutional Block Attention Module (CBAM) before classification. Rather than layering incremental tweaks onto existing designs, this architecture is built around a single question: How much accuracy can be retained once every component is stripped to what an edge device can actually run, and the CBAM stage in particular is placed after the Transformer rather than as a bolt-on addition, which the ablation results later show matters. Training and evaluation used a merged dataset comprising 45 classes, chosen specifically to stress-test the model beyond the single-crop benchmarks common in prior work. On this dataset, AgriViTNet reaches 99.49% overall accuracy, ahead of the MobileH-Transformer, TwoConV-BiLSTM, and VGG-Inception-DenseNet baselines compared here, while staying at 3.1 million parameters and 1.54 GFLOPs, roughly two orders of magnitude smaller than the heaviest hybrid tested. This robust performance, combined with its minimal resource requirements, enables deployment in real-world agricultural settings, supporting timely disease management and sustainable farming practices. The architecture's scalability and efficiency make it a practical solution for precision agriculture, bridging the gap between technological innovation and field deployment.

Keywords: Plant Disease Detection, IoT, Edge Computing, CNN-Transformer Hybrid, CBAM, Lightweight Networks

Introduction

Global food production is increasingly threatened by plant pathogens, and the cost is measured not just in yield but in the speed of response: Delayed diagnosis often means the difference between a manageable outbreak and a lost harvest. In regions where smallholder farming dominates, traditional disease identification, relying on visual inspection by agronomists or laboratory testing, does not scale to the pace and geographic spread required for modern precision agriculture, creating a pressing case for automated, field-deployable detection tools (Mahbub, 2020; Morchid et al., 2024b).

Within this IoT landscape, Edge Computing plays a specific role: Moving computation onto the device itself, right where the data is captured, rather than shipping it off to a remote server, which cuts the latency, bandwidth use, and cloud dependency that come with that round trip. Unlike centralized architectures, local data processing enables faster and more secure decision-making while minimizing communication costs and data privacy risks (Li et al., 2020; Uddin et al., 2021). In precision agriculture, this approach allows AI models to be deployed directly in the field, enabling real-time disease detection, crop monitoring, and optimized resource management based on data collected from sensors or drones (Shelar et al., 2022; Panchal et al., 2023). In addition, it improves energy efficiency and operational resilience, contributing to the development of autonomous, intelligent, and sustainable farming systems (Rajak et al., 2023; Yilmaz et al., 2025).

Computer vision took an early lead in plant disease detection through Convolutional Neural Networks (CNNs), which turned out to be well suited to picking out the local, discriminative details (a lesion's edge, a leaf's texture) visible in sensor-captured images (Selvam et al., 2024; Dewangan and Vij, 2024). However, CNNs have limitations in modeling global spatial relationships and generalizing across diverse datasets, especially in resource-constrained IoT environments due to their high computational cost. Transformers took the opposite route: Born for language modeling, they turned out to translate surprisingly well to images once self-attention was applied to patches instead of words (Rashid et al., 2024; Gawande et al., 2024). However, their even higher computational demand and weaker local inductive bias limit their deployment on Edge and IoT devices with restricted resources (Nawaz and Babar, 2025; Wang and Kang, 2025; Pintus et al., 2025).

These opposite strengths and weaknesses point toward an obvious fix: Let a CNN handle local texture and a Transformer handle long-range context, in the same pipeline. Panchal et al. (2023) and Rahman et al. (2025) confirm this works in practice: Hybrid CNN-ViT models in their studies achieve 90% accuracy while using fewer parameters and less compute than a pure Transformer, making them a better fit for edge computing. Despite these advances, a major challenge remains: Designing edge-efficient architectures that maintain high accuracy with fewer parameters and reduced memory and energy costs. Despite these advances, a major challenge remains: Designing edge-efficient architectures that maintain high accuracy with fewer parameters and reduced memory and energy costs.

However, this progress masks three critical gaps:

1. The accuracy-efficiency paradox pure CNNs achieve insufficient accuracy on multi-crop datasets due to poor global context, while accurate hybrids demand computational resources far exceeding edge device constraints, and lightweight hybrids sacrifice accuracy that translates to thousands of missed disease instances in field deployment
2. Lack of attention-based refinement in lightweight hybrids existing CNN-Transformer hybrids designed for edge deployment omit sophisticated attention mechanisms like CBAM that are known to improve discriminative power in image classification, likely because their computational cost was believed incompatible with lightweight design, creating an unexplored opportunity to integrate selective attention without excessive overhead
3. Absence of large-scale validation prior work validates on single-crop or limited multi-crop subsets, never on the diversity required for real edge deployment with multiple crops and dozens of disease classes, leaving uncertainty about generalization under genuine inter-class confusion. These three gaps define the precise research objective addressed by AgriViTNet

In this work, we introduce AgriViTNet, a hybrid CNN-Transformer architecture enhanced with a CBAM, specifically designed for IoT and edge computing environments, to close these three gaps simultaneously. The proposed AgriViTNet model combines:

1. A depthwise separable CNN backbone with multi-scale kernels for efficient local feature extraction, reducing parameters compared to standard convolutions while preserving fine-grained texture and disease symptom detection
2. A single-layer Transformer encoder that captures long-range spatial dependencies with minimal computational overhead, enabling global context awareness without sacrificing edge deployability
3. A CBAM integrated after the Transformer encoder to refine globally contextualized features through sequential channel and spatial attention. This strategic integration demonstrates that sophisticated attention mechanisms can be incorporated into lightweight hybrids without prohibitive computational cost, selectively emphasizing disease-relevant features while suppressing background noise an opportunity previously unexplored in edge-oriented architectures

The result is a model achieving 99.49% accuracy on a genuinely complex 45-class dataset spanning 15 plant species (86,138 images) the most diverse validation benchmark in lightweight hybrid agriculture literature while maintaining an ultra-compact footprint of 3.1 M parameters, 1.54 GFLOPs, and 9.7 MB model size. This architecture directly addresses the three identified gaps: It breaks the accuracy-efficiency paradox by achieving >99% without the 150 M parameter penalty; it demonstrates that post-Transformer attention is both theoretically sound and empirically superior; and it provides the first rigorous proof that lightweight hybrids maintain accuracy under realistic multi-crop deployment complexity.

The main contributions of this work directly address the three identified research gaps and are summarized as follows:

1. A theoretically grounded and empirically validated lightweight hybrid with integrated attention: We build AgriViTNet around three deliberate design choices. Local texture extraction is handled by depthwise separable convolutions, which capture fine-grained symptom detail. Global context needed to relate symptoms spread across different regions of a leaf is added through a single Transformer layer at minimal cost. Finally, a CBAM module refines the resulting features. Unlike existing lightweight hybrids that omit attention to control cost, this architecture proves that sophisticated attention mechanisms can be incorporated without compromising edge deployability. The integration of CBAM with the Transformer output (rather than as a standalone addition) creates a synergistic effect where attention refines contextualized features for superior discriminative power. The result is robust multi-scale feature representation and superior generalization across heterogeneous disease classes, delivering both high accuracy and genuine edge deployability
2. Closing the accuracy-efficiency paradox on multi-crop datasets: AgriViTNet simultaneously achieves
   (i) >99% accuracy
   (ii) <3.2 M parameters
   (iii) <2 GFLOPs
   (iv) Validation on 15 crops and 45 disease classes a simultaneous optimization of three competing objectives that no prior lightweight hybrid achieves. This breaks the historical trade-off where lightweight models sacrifice 2-3% accuracy or heavy models demand 100x more parameters. Critically, the integration of CBAM proves that attention mechanisms need not be abandoned in the pursuit of edge efficiency
3. Rigorous, large-scale validation proving robustness under real deployment complexity: We provide the first systematic evaluation of a lightweight hybrid on a genuinely diverse 45-class, 15-crop dataset (86,138 images) with statistical validation across 5 independent runs, Wilcoxon rank-sum significance testing, and detailed ablation studies. This provides evidence that the architecture generalizes under true edge deployment scenarios, not just controlled benchmarks
4. Explicit mechanistic analysis of CBAM's role in lightweight architectures: The ablation study quantifies CBAM's contribution on heterogeneous data, showing that removing CBAM causes a 1.92% accuracy drop despite only 0.3 M parameter reduction revealing that CBAM's value is synergistic with global Transformer features, not merely additive. This insight challenges the assumption that attention mechanisms are inherently too costly for edge deployment and informs future lightweight hybrid design toward incorporating selective attention

Related Work

Smart agriculture research has drawn on AI, edge computing, and IoT sensor networks to push disease detection earlier in the crop cycle. Several reviews have mapped this space from different angles: On the computer vision side, survey work covering UAV-based imaging, robotics, and IoT integration in precision agriculture, spans contributions of Akhtar et al. (2021); Rajak et al. (2023); Dang et al. (2024); Morchid et al. (2024a); Soussi et al. (2024); Delfani et al. (2024); Wang and Kang (2025); Yilmaz et al. (2025); Nawaz and Babar (2025). A separate strand of systematic reviews, Shoaib et al. (2023; 2025); Pintus et al. (2025) focused more narrowly on deep learning for plant disease classification specifically, tracing how detection methods have evolved and where data scarcity and environmental variability still limit progress.

CNNs became the default choice for plant disease detection early on, and for a straightforward reason: Convolutions are naturally suited to picking up the spatial and textural cues (a spot's shape, a leaf's grain) that define most disease symptoms.

Pang et al. (2021) proposed an image-processing algorithm using artificial neural networks to detect four tomato diseases from plantation to harvest, achieving 93% accuracy based on morphological features. Shelar et al. (2022) pushed a CNN on potato and strawberry images to 95.6% accuracy, relying on early stopping to cut training short at 50 epochs. Dhiman et al. (2023) worked on citrus, pairing a CNN with an LSTM in an edge-computing setup down-sampling and feature fusion brought accuracy to 98.25% once the model was pruned. Similarly, Putrada et al. (2024) optimized a CNN via grid search and pruning for corn disease classification, introducing a Normalized-Geometric Mean method and reaching 94.7% accuracy. Edge-oriented approaches continued with Selvam et al. (2024) (DCNN-LSTM for IoT devices) and Dewangan and Vij (2024) (CNN-LSTM for UAV farmland monitoring). da Silva and Almeida (2025) evaluated thermal-image-based classification using InceptionV3, MobileNet, and VGG16 on Raspberry Pi 4B, maintaining 96% accuracy after pruning and quantization. Other CNN-based systems were modeled by Rahman et al. (2025); Ramana et al. (2025) for multi-crop disease classification and infection severity estimation, achieving accuracies of 95.62 and 92.06%, respectively, demonstrating progress toward efficient, real-world agricultural deployment.

These networks have proven efficient in learning fine-grained local features and handling intra-class variations in disease symptoms. However, despite their effectiveness, a convolution only sees its immediate neighborhood, so relating a symptom on one side of a leaf to one on the other requires stacking many layers just to grow the receptive field far enough. That depth comes with a cost that matters for field deployment: Heavier computation, which sits awkwardly with the tight budgets of real-time IoT and edge inference.

Vision Transformers have started showing up in agricultural imaging for exactly the reason CNNs fall short: Self-attention relates every part of an image to every other part, sidestepping the receptive-field limit altogether. Unlike CNNs, Transformers divide input images into patches and model global dependencies between them, leading to improved robustness against spatial distortions and variations in disease patterns. Ali et al. (2025) apply ViTs directly to plant disease detection as an alternative to manual visual inspection and lab testing, training and evaluating their model on a balanced 55-class dataset. The result, 90% overall accuracy, comes without any CNN component at all, giving a useful reference point for what self-attention alone can achieve on this kind of task. Haque et al. (2025) push further on the attention side itself: Rather than standard multi-head attention, their model stacks triplet Multi-Head Attention (t-MHA) units with residual connections, refining feature representations more aggressively than a vanilla ViT would. Tested on RicApp, a new expert-curated dataset covering rice and apple diseases, this refinement translated into 97.99% accuracy on unseen data, a 2.2-point gain over the standard ViT baseline.

That accuracy gain isn't free, though both papers rely on full Transformer stacks, and the same computational appetite that makes self-attention powerful also makes it a poor fit for the tight memory and latency budgets of edge hardware.

The CNN and Transformer papers above point in opposite directions on the same trade-off, which is exactly what has pushed hybrid CNN-Transformer designs to the front of recent work: Let convolutions handle what they're already good at (local texture), and hand long-range reasoning to a Transformer block layered on top. Mobile H-Transformer, from Thai and Le (2025), targets an even tighter budget than the papers above drones and smartphones rather than a desktop GPU. A dual convolutional block does the initial feature extraction and shrinks the input before it ever reaches the Transformer stage, keeping the whole pipeline at just 0.4 GFLOPs (Giga Floating Point Operations). That efficiency shows up directly in deployment: 30.5 FPS on mobile hardware, with F1-scores of 97.20% on corn and 96.80% on the PlantVillage subset. Aboelenin et al. (2025) go in the opposite direction on model size: Their framework fuses three heavyweight CNN backbones VGG16, InceptionV3, DenseNet201 with a ViT head, tested on Apple and Corn (four disease classes each). It tops out at 99.24% accuracy on Apple and 98% on Corn, ahead of prior methods on both datasets, but that ensemble of three full CNNs plus a Transformer comes at a size that puts resource-constrained deployment out of reach. Vikhe et al. (2025) bring the same hybrid recipe to a crop none of the papers above touch: Onion, using images gathered from Google Images and Kaggle and cleaned up through several pre-processing steps. Benchmarked against a plain CNN, ResNet-50, and VGG-16, their hybrid model reached 98.75% accuracy ahead of all three baselines. Prommakhot et al. (2025) take the hybrid idea in yet another direction with TransNet: Two-stream convolutions feed a Transformer network built around sequential learning components LSTM, BiLSTM, sequence-to-sequence, and GRU layered together rather than a single recurrent block. Trained and evaluated on PlantVillage, covering 38 plant disease species. Experimental results show an accuracy of 97.88%.

Hybrid deep learning has moved the field forward, but a real gap remains between what gets published and what actually runs on IoT hardware in the field. Existing approaches tend to fall into one of two camps: Either they demand more compute than a low-power edge device can offer, or they cut down to a lightweight design at the cost of accuracy. For example, large-scale hybrid models such as DenseNet combined with vision transformers, while highly accurate, involve hundreds of millions of parameters and high computational costs, leading to large memory footprints and slow inference times unsuitable for resource-constrained deployment.

Conversely, lightweight models like MobileNetV3 or Efficient Net-lite, although computationally efficient, tend to sacrifice some discriminative capacity, resulting in reduced accuracy, especially across diverse and complex agricultural datasets. Even lightweight architectures provide partial solutions, but they often sacrifice accuracy when compressed or quantized for on-device inference. Similarly, knowledge distillation, though popular for model reduction, typically leads to significant degradation in feature discrimination and disease classification precision, particularly in fine-grained agricultural imagery. Moreover, most existing architectures address CNN-Transformer hybridization at a high computational cost, emphasizing accuracy over efficiency. Furthermore, many of these models lack effective attention mechanisms, such as CBAM, which can enhance feature selectivity and model interpretability, especially vital in heterogeneous field conditions with variable lighting, background noise, and intricate disease symptoms.

Positioning and research gap: The literature progress masks three critical, interrelated gaps:

1. The Accuracy-Efficiency Paradox on Heterogeneous Datasets: Pure CNN models (1-3 M parameters, <1 GFLOPs) achieve only 94-96%

accuracy on multi-crop, multi-disease datasets because they fail to capture global spatial dependencies essential for diseases with large-scale or systemic symptoms. In contrast, accurate hybrid CNN-ViT models consistently exceed 96% accuracy but demand 100-150 M parameters and 20–30 GFLOPs a computational burden that far exceeds the memory (> = 100 MB) constraints of resource-limited IoT edge devices. Lightweight hybrids partially address this by limiting Transformer depth, but sacrifice 2-3% accuracy, which translates to thousands of missed disease instances in field-scale deployment

2. Unexplored Integration of Attention Mechanisms in Lightweight Hybrids: A critical observation in the literature is that existing lightweight CNN-Transformer hybrids (MobileH-Transformer, TwoConV-BiLSTM) deliberately omit sophisticated attention mechanisms, likely because CBAM and similar modules are perceived as computationally expensive and fundamentally incompatible with edge-efficient design. This creates a design gap: While CBAM has proven valuable in standard computer vision, no work has systematically validated whether CBAM can be incorporated into lightweight hybrids without violating edge constraints. More fundamentally, the assumption that attention refinement must be sacrificed for edge efficiency has never been rigorously tested on heterogeneous, multi-crop plant disease datasets where such refinement could provide substantial discriminative benefit. This gap represents both a technical and a scientific opportunity
3. Absence of Validation at Scale on True Multi-Crop Complexity: Prior hybrid models are benchmarked on single-crop datasets (Apple, Corn alone) or limited multi-crop subsets. Real agricultural edge deployment requires handling 15+ crops with 40+ disease classes simultaneously. This scale of inter-class visual similarity and feature saturation is not systematically addressed in prior work, leaving uncertainty about whether lightweight hybrids maintain accuracy under genuine deployment complexity

These three gaps define the precise research objective addressed by AgriViTNet.

This work aims to bridge these three critical gaps by proposing AgriViTNet, a hybrid network small enough to run in real time on IoT edge hardware without giving up detection accuracy. Our model balances high classification accuracy with low computational complexity, facilitating scalable, on-device plant health monitoring systems crucial for sustainable, precision agriculture.

Materials and Methods

AgriViTNet Architecture

AgriViTNet brings together three components, each covering what the others miss: A CNN backbone for local texture, a Transformer encoder for context that spans the whole leaf, and a CBAM module that sharpens focus on the regions that matter. A compact MLP classifier turns the resulting features into a final prediction. None of these four stages is heavy on its own, and together they stay within a footprint an edge device can realistically run.

Architecture Overview

AgriViTNet follows a modular design composed of four main processing stages (Fig. 1):

1. Stage: Lightweight CNN Backbone: Reads texture and edge detail straight from the RGB image
2. Stage: Lite Transformer Encoder: Models contextual, long-range relationships between different regions of the leaf
3. Stage: CBAM Attention Refinement: Reweights the intermediate feature maps in both channel and spatial dimensions
4. Stage: Compact MLP Classifier: Aggregates high-level features for disease category prediction

To ensure the complementarity between local texture and fine detail extraction performed by the CNN backbone and the modeling of global relationships through the Transformer encoder, particular attention was paid to integrating the CBAM module. This module helps suppress noise and emphasize relevant regions within the image, thereby enhancing the discriminative power of the model in heterogeneous and cluttered agricultural environments. The design of this hybrid architecture aims to balance computational efficiency with high classification performance, making it suitable for resource-constrained IoT environments.

Fig. 1: The proposed AgriViTNet architecture

Stage 1 Lightweight CNN Backbone

Color shifts, texture, and the earliest signs of a lesion all live in the fine detail of a leaf image, exactly what the CNN backbone is built to pick up, using depthwise separable convolutions across a mix of kernel sizes. Splitting the convolution this way cuts the parameter count sharply compared to a standard convolution, without blurring out the fine detail that a subtle symptom depends on. By focusing on multi-scale kernels, our CNN backbone effectively encodes diverse textural information at various resolutions, enhancing robustness against variability caused by lighting, leaf orientation, or surface irregularities. Our CNN backbone consists of three convolutional blocks, each followed by Batch Normalization and SiLU activation functions. This design captures multi-scale textures efficiently with fewer parameters.

This stage encodes low-level and mid-level spatial representations crucial for early disease pattern recognition. The SiLU activation is selected for its smooth gradient flow and nonlinearity efficiency, ideal for small CNNs. The convolution strides $(2, 2, 1)$ progressively downsample the feature maps, transforming input images of 224×224×3 into compact 28×28×256 embeddings.

In this first stage, each convolutional ConvBlock is composed of a 3×3 convolution, batch normalization, and a SiLU activation, as defined in Eq. 1:

$$X_i = SiLU\Big(BN\big(Conv_{3\times3}(X_{i-1})\big)\Big) \tag{1}$$

Formally, let $x \in \mathbb{R}^{H\times W\times C}$ denote the input tensor. The output of the i-th convolutional branch is computed as in Eq. 2:

$$F_i = \sigma\Big(BN\Big(Conv_{k_i}(x)\Big)\Big) \tag{2}$$

Where $Conv_k$ denotes a convolution operation with kernel size $k_i \in \{3, 5, 7\}$, BN is batch normalization, and σ represents the non-linear Swish (SiLU) activation.

Each kernel size sees the image at a different scale, so their outputs are stacked along the channel axis rather than merged into one, keeping all three scales available side by side in the resulting feature map (Eq. 3):

$$F_{CNN} = \oplus_{i=1}^{n} F_i \tag{3}$$

Where $\oplus$ denotes channel-wise concatenation.

The output feature map F_{CNN} serves as input to the Transformer encoder.

Stage 2 Lite Transformer Encoder

To enhance global feature representation beyond local CNN extraction, AgriViTNet integrates a lightweight Transformer encoder. This module captures long-range dependencies with reduced computational cost, suitable for IoT and edge environments.

At its heart, the encoder runs on Multi-Head Self-Attention (MHSA), with a Feed-Forward Network (FFN) and residual connections built around it.

This second stage first divides the intermediate CNN feature tensor $F_{CNN} \in \mathbb{R}^{B\times C\times H\times W}$ into non-overlapping patches of size $P \times P$ and flattens them into a sequence of $N_p = \frac{HW}{/P\text{^}2}$ tokens as shown in Eq. 4:

$$Z = Patchify(F_{CNN}) \in \mathbb{R}^{B\times N_p\times C} \tag{4}$$

Where N_p is the number of patches for a patch size $P = 16$; $C = 256$ is the input embedding dimension, and B is the batch size.

Each patch sequence is processed by the attention mechanism described by Eqs. 5 and 6 with residual connections and Layer Normalization (LN):

$$Z_1 = LN\big(Z + Z_{attn}(Z)\big) \tag{5}$$

$$Z_{attn} = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

Where $Q = ZW_Q$, $K = ZW_K$, and $V = ZW_V$ are learned linear projections, and d_k is the dimensionality of the key vectors.

Then, each patch is refined through a lightweight FFN wrapped, like the attention step, in residual connections and LN given by Eqs. 7 and 8. This FFN widens the representation to four times its original dimension:

$$Z_2 = LN\big(Z_1 + FFN(Z_1)\big) \tag{7}$$

$$FFN(Z_1) = \sigma\big(Z_{1W_1} + b_1\big)W_2 + b_2 \tag{8}$$

Where $\sigma(\,)$ is the GELU activation function.

The final output of the encoder, denoted as Z_{out}, is reshaped and upsampled to restore the original spatial dimensions, as shown in Eq. 9:

$$F_{TRANS} = Reshape(Z_{out}) \in \mathbb{R}^{B\times C\times H\times W} \tag{9}$$

The encoder employs the following lightweight configuration:

- Input dimension: 256
- Number of heads: 4
- Feed-forward expansion: 256 → 1024 → 512 → 256
- Patch size: 16

This block effectively captures non-local dependencies, enhancing complex leaf textures and spatially distributed symptom recognition. The lightweight configuration ensures reduced computational load (≈0.8 M parameters) while maintaining contextual awareness, enough to run on edge hardware.

Stage 3 CBAM Attention Refinement

AgriViTNet adds a CBAM at this stage to recalibrate what the Transformer has produced first deciding which channels matter, then which spatial locations do, in that order. The channel attention mechanism emphasizes features with high discriminative relevance, such as necrotic spots or vein discolorations, while suppressing background noise. Spatial attention further localizes these disease features, guiding the network to focus on pertinent regions within complex plant imagery. This dual attention mechanism makes the model both easier to interpret and sharper at telling classes apart, especially where leaf backgrounds are cluttered or textures vary widely.

Given the input feature tensor $F_{TRANS} \in \mathbb{R}^{B\times C\times H\times W}$, the CBAM module refines it through the following two steps:

Channel Attention: This step focuses on "what" is meaningful in the feature map. The feature map is pooled in two ways, average and max, both collapsed along the spatial dimensions, giving two compact descriptors that summarize the global context per channel. Both pass through the same shared multilayer perceptron MLP, whose combined output forms the channel-wise attention mask of Eq. 10:

$$M\ \ (\_c\,F) = \sigma_1\Big(\big((\ \ )\big)\,(MLP\_AvgPool\,F + MLP\big(\_MaxPool(\ \ )\big)\Big)\,F) \tag{10}$$

Where $\sigma_1(\,)$ is the sigmoid function. The channel-refined feature map is then obtained by Eq. 11:

$$F' = M_c(F) \odot F \tag{11}$$

Where $\odot$ represents the Hadamard (element-wise) product.

Spatial Attention: It picks up where channel attention leaves off: Rather than deciding which feature channels matter, it decides which spatial locations do. Starting from the channel-refined tensor F′, it pools across channels and runs a 7 × 7 convolution over the result (Eq. 12) to produce a map of where the network should be looking:

$$M_s(F') = \sigma_1\big(f^{7\times 7}([AvgPool(F');\ MaxPool(F')])\big) \tag{12}$$

Yielding the final CBAM-refined output given by Eq. 13:

$$F_{CBAM} = M_s(F') \odot F' \tag{13}$$

This module is what teaches AgriViTNet to tell the difference: A necrotic spot or a discolored vein carries the diagnostic signal, while soil texture and inconsistent lighting do not. The compact structure of CBAM ensures negligible computational overhead while improving feature selectivity and visual interpretability.

The CBAM stage produces an output feature map of 28×28×256 while adding only $\approx 0.3M$ parameters, thus maintaining the model's lightweight footprint.

Stage 4 MLP Classifier

After feature refinement through the CBAM, the aggregated feature maps are globally averaged and flattened into a compact vector $x \in \mathbb{R}^{256}$. A lightweight Multi-Layer Perceptron (MLP), denoted as $F_{MLP}(\,)$, performs the final classification. This stage adopts an expansion compression architecture that enhances the representation capacity while preserving computational efficiency.

The MLP transformation can be formally expressed as Eqs. 14-16:

$$h_1 = \sigma(W_1\,x + b_1) \tag{14}$$

$$h_2 = \sigma(W_2\,h_1 + b_2) \tag{15}$$

$$\hat{y} = Softmax(W_3\,h_2 + b_3) \tag{16}$$

Where $\sigma\,(\cdot)$ denotes the GELU activation function, and $\hat{y} \in \mathbb{R}^{N_c}$ is the predicted class probability vector for N_c disease categories.

Put together, the classifier reduces to Eq. 17:

$$\hat{y} = F_{MLP}(x) \tag{17}$$

The internal structure of the MLP classifier is summarized below. It follows an expansion-compression design that balances feature richness and parameter efficiency. Each stage applies dropout regularization and GELU activation to enhance generalization, non-linearity, and prevent overfitting:

- Global Average Pooling (GAP) → Flatten
- Linear (256 → 512) + GELU + Dropout (0.3)
- Linear (512 → 256) + GELU + Dropout (0.3)
- Linear (256 → N_c) + Softmax

This classifier contains approximately 0.8M trainable parameters, balancing expressiveness and computational efficiency.

In summary, F_{MLP} maps the globally pooled CBAM features to class logits, forming the final decision layer of AgriViTNet.

Mathematical Formulation of the Overall Pipeline

Let $I \in \mathbb{R}^{H\times W\times 3}$ denote an input RGB image of a plant leaf. The AgriViTNet model can be expressed as a composite function composed of four main transformations as shown in Eq. 18:

DOI: 10.3844/jcssp.2026.2736.2754

$$\hat{y} = F_{MLP}\ (F_{CBAM}\ (F_{TRANS}\ (F_{CNN}(I)))) \tag{18}$$

Where:

- F_{CNN} extracts local feature representations
- F_{TRANS} models global spatial dependencies
- F_{CBAM} adaptively reweights features through attention,
- F_{MLP} performs final classification

Table 1 breaks down each stage's parameter count and computational cost.

Overall, AgriViTNet achieves remarkable compactness, and its overall computational footprint remains below 1.6 GFLOPs with only 3.1M parameters, comfortably within reach of the memory and power budget a typical IoT edge device has to work with.

Dataset Description

We built the training corpus by merging two public datasets: New PlantVillage and Banana Leaf Disease, bringing the total to 45 disease classes spread across 15 crops (Table 2). Every image was resized to 224×224 pixels to match AgriViTNet's input layer.

Data Preprocessing and Augmentation

Images were preprocessed and augmented using a customized transformation pipeline. Each image was first resized and normalized according to the ImageNet statistics ($mean = [0.485, 0.456, 0.406]$, $std = [0.229, 0.224, 0.225]$), with training and validation following different transformation pipelines from that point on.

Training Transformations. During training, a rich augmentation strategy was employed to simulate the variability of real-world agricultural environments. The following operations were applied sequentially:

- *Random Resized Crop*: Crops a random region of the image and resizes it to the target dimension 224×224, with a scale sampled in the range $[0.7, 1.0]$
- *Random Horizontal Flip* and *Random Vertical Flip*: Randomly mirror the image along both axes to increase orientation invariance
- *Random Rotation*: Applies a random rotation in the range $\pm 30°$ to simulate natural leaf orientations

Table 1: Parameters and Computational Complexity Summary of AgriViTNet

Stage	Description	Params (M)	GFLOPs	Output Dim
Stage 1	CNN Backbone	1.2	0.65	56×56×256
Stage 2	Lite Transformer Encoder	0.8	0.48	56×56×256
Stage 3	CBAM Attention Module	0.3	0.22	28×28×256
Stage 4	MLP Classifier	0.8	0.19	$1 \times N_c$
Total	—	≈3.1 M	≈1.54	—

Table 2: Summary of the dataset used in this study

Plant Species	Number of Classes	Total Images	Class Names
Apple	4	9,714	Black Rot, Scab, Healthy, Cedar Rust
Banana	7	2,856	Black sigatoka, Bract mosaic virus, Healthy, Insect pest, Moko, Panama, Yellow sigatoka
Blueberry	1	2,270	Healthy
Cherry	2	4,386	Powder, Healthy
Corn (Maize)	4	9,145	Common Rust, Healthy, Northern Leaf Blight, Cercospora Leaf Spot
Grape	4	9,027	Black Rot, Esca Black Measles, Isariopsis Leaf Spot, Healthy
Orange	1	2,513	Haunglonbing
Peach	2	4,457	Bacterial Spot, Healthy
Pepper	2	4,876	Bacterial Spot, Healthy
Potato	3	7,128	Early Blight, Healthy, Late Blight
Raspberry	1	2,226	Healthy
Soybean	1	2,527	Healthy
Squash	1	2,170	Powdery mildew
Strawberry	2	4,498	Leaf scorch, Healthy
Tomato	10	18,345	Early Blight, Healthy, Late Blight, Bacterial Spot, Leaf Mold, Mosaic Virus, Septorial Leaf Spot, Target Spot, Spider Mites, Yellow Leaf Curl Virus
15 Species	45 Classes	86,138 Images	—

- *Color Jitter*: Brightness, contrast, and saturation each perturbed within ±0.3, hue within ±0.1, to simulate the range of lighting a leaf might be photographed under
- *Gaussian Blur*: Applies a Gaussian kernel with radius 3 and $\sigma \in [0.1, 2.0]$ to model camera or motion blur
- *Random Erasing*: Randomly removes small rectangular regions ($scale \in [0.02, 0.15]$, $ratio \in [0.3, 3.3]$) with probability $p = 0.25$, encouraging occlusion robustness
- *To Tensor* and *Normalize*: Convert the image to a tensor and normalize pixel values based on ImageNet statistics

Validation Transformations. For validation and testing, only deterministic preprocessing steps were applied to ensure fair performance evaluation:

- *Resize*: Every image is brought to (224×224)
- *ToTensor* and *Normalize*: Conversion and normalization using the same ImageNet statistics

This minimal preprocessing ensures consistency during inference while preserving the natural image structure.

Data Splitting and Stratification

To prevent bias from class imbalance, we employed stratified random splitting for all training runs:

- Stratified K-Fold Splitting: The 86,138 images were split 70/20/10 into training, validation, and test sets, with stratification preserving each class's original proportion in every split. This preserves the imbalance ratio across all subsets
- Class-Balanced Loss Function: During training, we applied no explicit class weighting. However, the balanced augmentation strategy (Random Resized Crop, Color Jitter, Random Erasing, etc.) applied uniformly across all classes ensured that rare classes received equal representation during optimization

Training and Validation Protocols

We trained AgriViTNet end-to-end in PyTorch on a single NVIDIA GPU with 10 GB of VRAM, balancing three competing concerns throughout: Stable convergence, enough regularization to avoid overfitting on the smaller disease classes, and a training time that stayed practical given the available hardware.

Optimization. We optimized with AdamW rather than standard Adam, since decoupling weight decay from the gradient update tends to generalize better in practice. The learning rate started at 3×10^{-5} (weight decay 10^{-3}). It followed a cosine annealing schedule over the 50 training epochs, easing the model toward a minimum rather than dropping the rate abruptly.

Loss Function and Label Smoothing. We trained with Cross-Entropy loss with label smoothing ($\varepsilon = 0.1$), which keeps the model from becoming overconfident on training labels it has seen many times. In experiments involving sample mixing, a Soft Target Cross-Entropy loss was used to accommodate soft label distributions generated by the MixUp augmentation strategy.

Regularization and Robustness. Several regularization mechanisms were incorporated to improve robustness and avoid overfitting:

- Weight Decay: Penalizes large weights through an L_2 regularization term integrated into AdamW
- Dropout: Applied with a rate of 0.3 in the MLP head to improve generalization and prevent overfitting
- Early Stopping: We stopped training once 15 straight epochs passed without any gain in validation accuracy
- MixUp: Applied during training to generate convex combinations of image label pairs, improving the smoothness of decision boundaries

Precision and Efficiency. Automatic Mixed Precision (AMP) via the GradScaler module, which dynamically adjusts floating-point precision to accelerate training while maintaining numerical stability. This allowed a reduction in GPU memory footprint by approximately 40% without loss of accuracy.

Evaluation Protocol. Validation accuracy and loss were tracked after every epoch, with only the checkpoint achieving the best validation accuracy kept for later testing. The hyperparameters governing this process are summarized in Table 3.

Table 3: Tuning hyperparameters

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-5}
Weight decay	1×10^{-3}
Scheduler)	CosineAnnealingLR (T_{max} = 50
Batch size	32
Epochs	50
Label smoothing	0.1
Dropout rate	0.3
Early stopping patience	15 epochs
Precision mode	Mixed (FP16/FP32)
Loss function	SoftTarget Cross-Entropy
Image size	224×224
Number of classes	45
DataLoader num_workers	4

This training strategy ensured a stable optimization process with reduced overfitting and improved convergence speed, leading to consistent generalization across unseen data samples.

Evaluation Metrics

Final evaluation used the checkpoint with the highest validation accuracy, tested on data it had never seen during training or validation. We report the standard classification metrics: Accuracy, precision, recall, F1-score, alongside one-vs-rest ROC curves and their AUC values, which capture how well each class is separated from the rest even when accuracy alone might mask class-specific weaknesses.

During inference, gradient computation was disabled to minimize memory usage and ensure efficient forward propagation. The model achieved real-time inference capabilities on edge-grade hardware, confirming its suitability for on-field IoT applications.

Results and Discussion

Training and Validation Performance

Accuracy Convergence

Accuracy climbs past 90% within the first ten epochs (Fig. 2) and settles around 99% by the end of training, with the validation curve sitting slightly above the training curve throughout, a pattern consistent with good generalization rather than overfitting.

Loss Convergence

Loss falls steeply over the first few epochs and flattens out near zero by the end of training (Fig. 3), with the validation curve tracking the training curve closely throughout no divergence between the two, no sign of the model memorizing the training set. This consistent behavior between the two curves demonstrates that the combination of the Adam W optimizer, Cosine Annealing LR scheduler, and regularization techniques (Dropout, Label Smoothing, Mixup) has led to a stable and well-regularized training process.

Performance Evaluation on Some Individual Plants

Apple Plant's Evaluation

On the Apple disease classification task, Table 4 shows that AgriViTNet achieves perfect performance (100% across all metrics) on Apple leaves, surpassing all other compared models. This indicates that the model successfully distinguishes between visually similar disease patterns and healthy leaf conditions with complete reliability. The exceptional results can be attributed to the model's strong feature extraction capability, efficient attention mechanisms, and balanced data augmentation strategy, which together enhance robustness to lighting, texture, and color variations. The uniform performance across all categories confirms that the network captures both subtle and high-level discriminative cues.

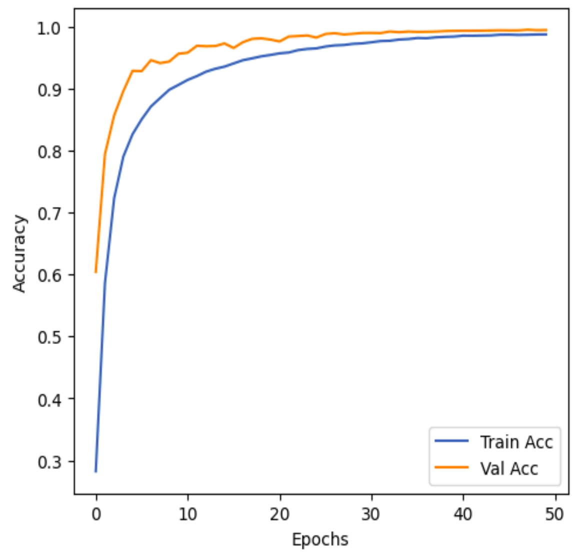


Fig. 2: Accuracy curves

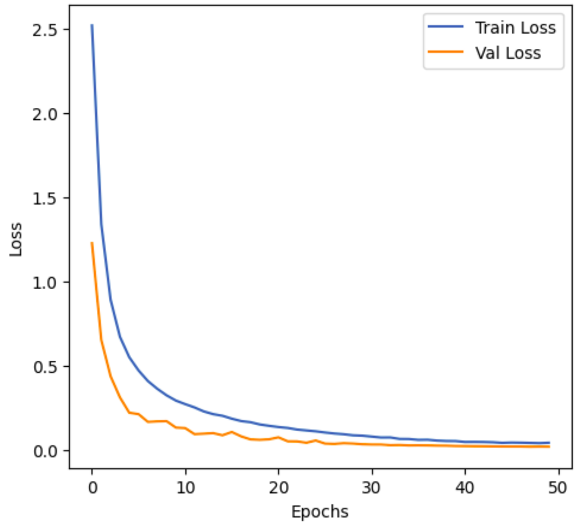


Fig. 2: Loss curves

Banana Plant's Evaluation

Table 5 shows that AgriViTNet outperforms competing models with 98% accuracy. Scores stay consistently high across the board, with no meaningful trade-off between false alarms and missed detections. Five of the seven banana classes Bract Mosaic Virus, healthy leaves, Moko, Panama, and Yellow Sigatoka were recognized without a single error (F1 = 1.00). The two exceptions, Black Sigatoka and Insect Pest Disease, saw recall dip slightly to 0.97 and 0.96, respectively: A handful of infected samples were misclassified as other leaf diseases, likely due to inter-class visual similarities or limited training examples. Overall, these results demonstrate that the model effectively generalizes to complex intra-class variations while maintaining high discriminative capability across multiple banana diseases.

Figure 4 breaks this down further, showing ROC curves for the two misclassified banana disease categories, as well as the global ROC curve computed over the full banana dataset. The ROC curves show near-perfect discrimination for the two non-perfectly classified classes (Black Sigatoka and Insect Pest), with AUC values close to 1.0 (≈0.999). The curves lie almost entirely along the top-left corner, indicating high sensitivity and specificity. The global (micro and macro) ROC curves also achieve an AUC ≈0.9999, confirming that the model demonstrates excellent overall classification performance on the banana dataset.

Cherry Plant's Evaluation

Table 6 reports that for Cherry leaves, AgriViTNet attains flawless classification (100%), while other models remain below 98%, confirming its superior generalization. This flawless performance demonstrates the model's excellent capacity to distinguish diseased and healthy cherry leaves. The results confirm the high sensitivity and specificity of the network, validating its robustness for practical deployment in field conditions where powdery mildew symptoms can be visually challenging to separate from healthy samples.

Table 4: Performance on Apple plant

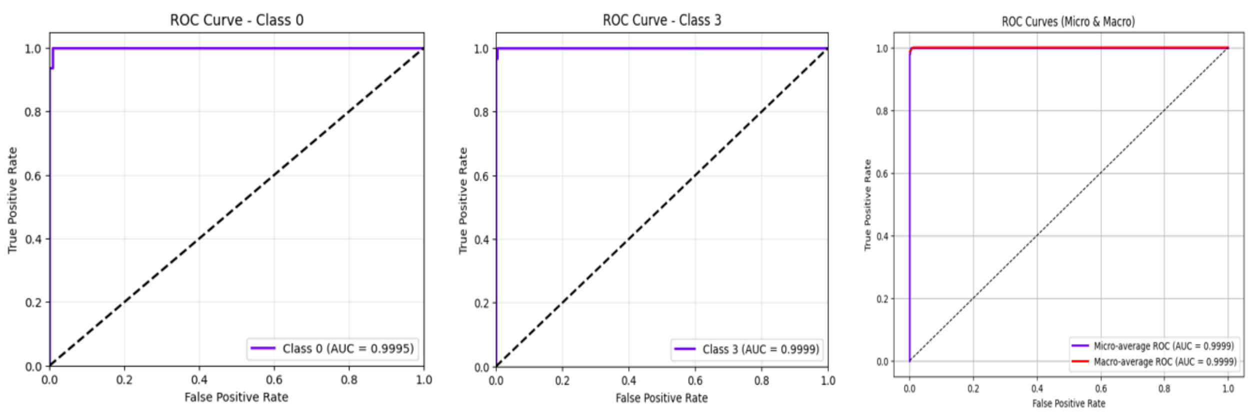
Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	99%	0.99	0.99	0.99
Hybrid model of Aboelenin et al. (2025)	99%	0.99	0.99	0.99
TwoConV-BiLSTM of Prommakhot et al. (2025)	99%	0.99	0.99	0.99

Table 5: Performance on banana plants

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	98%	0.98	0.96	0.97
MobileH-Transformer of Thai and Le (2025)	95%	0.95	0.95	0.95
Hybrid model of Aboelenin et al. (2025)	97%	0.97	0.97	0.97
TwoConV-BiLSTM of Prommakhot et al. (2025)	95%	0.95	0.92	0.93

Table 6: Performance on Cherry plant

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	97%	0.97	0.97	0.97
Hybrid model of Aboelenin et al. (2025)	98%	0.98	0.97	0.98
TwoConV-BiLSTM of Prommakhot et al. (2025)	96%	0.96	0.95	0.96



a- Class 0: Banana Black Sigatoka Disease

b- Class 3: Banana Insect Pest Disease

c - Global ROC curve

Fig. 4: ROC Curves on Banana

DOI: 10.3844/jcssp.2026.2736.2754

Corn (Maize) Plant Evaluation

Table 7 reports that AgriViTNet reaches 99% accuracy on Corn leaves, slightly improving over the 97-98% results of the alternative hybrid models. At 99% on Corn, AgriViTNet's one imperfection is telling rather than concerning: Only two samples of Cercospora leaf spot were mistaken for Northern Leaf Blight, both diseases that produce similarly elongated, tan-colored lesions on the leaf blade under certain lighting conditions. That the confusion stays this narrow two samples, one specific disease pair, out of thousands of test images spanning four classes is itself evidence that the model separates classes on genuine symptom differences rather than superficial ones.

Figure 5 presents the class-specific ROC curves for the misclassified Corn (maize) disease category, as well as the global ROC curve computed over the full Corn (maize) dataset. The two ROC curves confirm that the model demonstrates excellent overall classification performance on the Corn (maize) dataset.

Grape Plant's Evaluation

Grape presents four classes with visually distinct symptoms, and Table 8 shows AgriViTNet classifying every single test sample correctly. On the Grape dataset, AgriViTNet clearly outperforms all other models, which remain capped at 98%. A perfect score is a stronger signal of the model's underlying representational quality than of the task's difficulty alone.

Peach Plant's Evaluation

The classification results for peach leaf samples, reported in Table 9, demonstrate perfect performance. For Peach leaves, AgriViTNet maintains perfect prediction scores, whereas other approaches hover around 97-98%. The two-way split between healthy and Bacterial Spot-infected leaves leaves little room for error here, and the model's perfect score on both reflects that peach disease detection at this level of visual distinction is essentially solved.

Table 7: Performance on Corn (Maize) plant

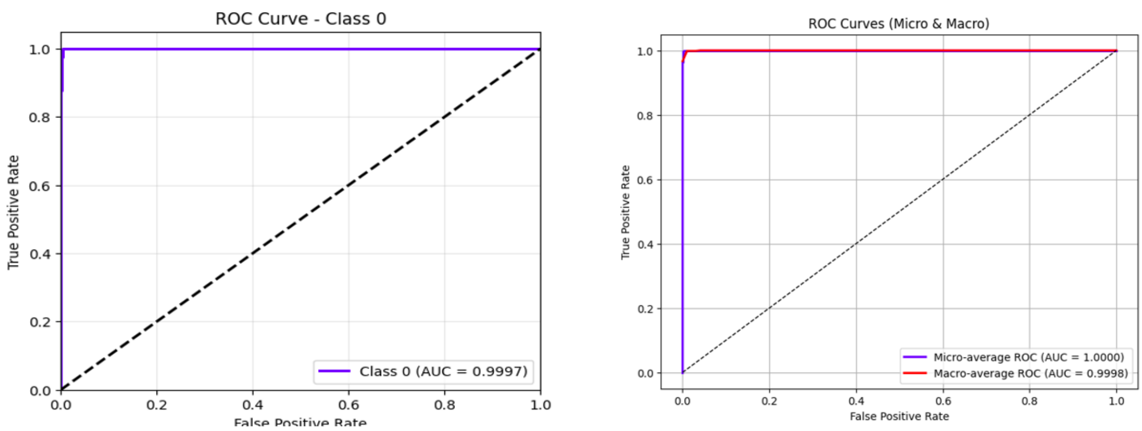
Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	99%	0.99	0.99	0.99
MobileH-Transformer of Thai and Le (2025)	97%	0.97	0.97	0.97
Hybrid model of Aboelenin et al. (2025)	98%	0.98	0.97	0.98
TwoConV-BiLSTM of Prommakhot et al. (2025)	97%	0.97	0.95	0.97

Table 8: Performance on grape plants

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	97%	0.97	0.96	0.97
Hybrid model of Aboelenin et al. (2025)	98%	0.98	0.98	0.98
TwoConV-BiLSTM of Prommakhot et al. (2025)	96%	0.96	0.95	0.96

Table 9: Performance on the peach plant

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	97%	0.97	0.96	0.97
Hybrid model of Aboelenin et al. (2025)	98%	0.98	0.98	0.98
TwoConV-BiLSTM of Prommakhot et al. (2025)	98%	0.98	0.98	0.98



a- Class 0: Corn(maize) Cercospora leaf spot Gray

b- Global ROC Curve

Fig. 5: ROC Curves on Corn(maize)

Pepper Plant's Evaluation

The classification results for bell pepper leaf samples indicate perfect model performance across both classes. Illustrated in Table 10, in the Pepper leaf classification task, AgriViTNet delivers perfect accuracy and F1-score, surpassing the closest competitor by 3-4%.

Potato Plant's Evaluation

Reported in Table 11, on Potato leaves, AgriViTNet reaches 100% across all metrics, outperforming the 96-97% scores of existing models. Every class scores a perfect 1.00 on precision, recall, and F1 the model neither missed a diseased leaf nor mislabeled a healthy one.

Strawberry Plant's Evaluation

The classification results for strawberry leaf samples show perfect model performance across both classes. The overall accuracy reaches 100%, indicating that all test samples were correctly classified. Table 12 reports perfect classification results by AgriViTNet for Strawberry leaves, improving slightly over the high-performing baselines (97-99%).

Tomato Plant's Evaluation

In Table 13, for Tomato diseases, AgriViTNet achieves near-perfect 99% accuracy, outperforming competing hybrid models limited to 96-97%. All tomato disease classes were recognized with high reliability, showing strong consistency across metrics.

Five of the ten tomato classes (Leaf Mold, Septoria Leaf Spot, Target Spot, Mosaic Virus, and healthy leaves) came out with perfect scores on every metric. The other five (Bacterial Spot, Early Blight, Yellow Leaf Curl Virus, Spider Mites Two-spotted, among others) fell just short of 1.0, most likely because their lesions share a similar color and shape that even a trained eye can find hard to tell apart. These results hold up specifically because tomato diseases are where symptom overlap is most severe among the fifteen crops tested if the model discriminates reliably here, it is a reasonable proxy for how it will behave on other visually ambiguous cases in the field.

Figure 6 breaks this down further, showing ROC curves for some misclassified Tomato disease categories, as well as the global ROC curve computed over the full Tomato dataset.

Comparative Analysis Across Models

Model Complexity and Practical Considerations

AgriViTNet's advantage becomes clear once its footprint is set against the alternatives (Table 14): 99.49% accuracy, the highest of the four models compared, achieved with only ≈3.1M parameters and ≈1.54 GFLOPs, resulting in an overall model size of approximately 9.7 MB. This balance between compactness and performance shows AgriViTNet staying accurate across all 45 disease classes without needing the extra capacity that heavier models rely on, exactly the balance low-resource and edge deployments require.

Table 10: Performance on the pepper plant

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	96%	0.96	0.96	0.96
Hybrid model of Aboelenin et al. (2025)	98%	0.98	0.97	0.98
TwoConV-BiLSTM of Prommakhot et al. (2025)	96%	0.96	0.94	0.96

Table 11: Performance on Potato plant

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	96%	0.96	0.96	0.96
Hybrid model of Aboelenin et al. (2025)	97%	0.97	0.97	0.97
TwoConV-BiLSTM of Prommakhot et al. (2025)	96%	0.96	0.96	0.96

Table 12: Performance on Strawberry plant

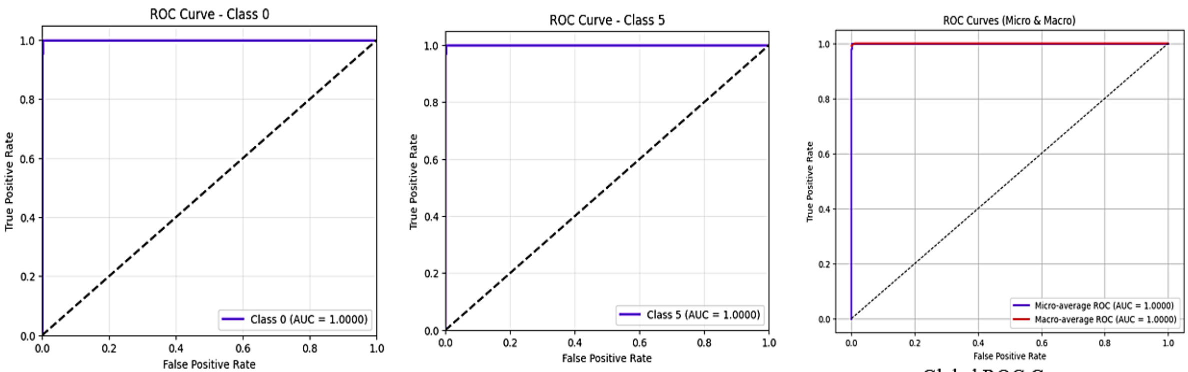
Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	100%	1.00	1.00	1.00
MobileH-Transformer of Thai and Le (2025)	97%	0.97	0.97	0.97
Hybrid model of Aboelenin et al. (2025)	97%	0.97	0.97	0.97
TwoConV-BiLSTM of Prommakhot et al. (2025)	99%	0.99	0.99	0.99

Table 13: Performance on tomato plants

Model	Accuracy	Precision	Recall	F1-score
AgriViTNet (Ours)	99%	0.99	0.99	0.99
MobileH-Transformer of Thai and Le (2025)	96%	0.96	0.96	0.96
Hybrid model of Aboelenin et al. (2025)	97%	0.97	0.97	0.97
TwoConV-BiLSTM of Prommakhot et al. (2025)	97%	0.97	0.96	0.97

Table 14: Comparison of model complexity

Model	Parameters	GFLOPs	Size	Accuracy
AgriViTNet (Ours)	≈3.1 M	≈1.54	9.7 MB	99.49%
MobileH-Transformer of Thai and Le (2025)	≈2.9 M	≈0.4	7 MB	97.10%
Hybrid model of Aboelenin et al. (2025)	≈150 M	≈25	760 MB	97.87%
TwoConV-BiLSTM of Prommakhot et al. (2025)	≈0.5 M	≈0.8	4.6 MB	96.46%



a- Class 0: Tomato Bacterial Spot b- Class 5: Tomato Spider mites Two-spotted c- Global ROC Curves

Fig. 6: ROC Curves on Tomato

In contrast, the MobileH-Transformer model by Thai and Le (2025) was designed specifically for mobile environments, achieving real-time processing with only ≈0.4 GFLOPs and ≈2.9 million parameters. However, despite its extremely low computational demand, it reported a slightly lower accuracy (97.10%) compared to AgriViTNet. This indicates that while MobileH-Transformer excels in efficiency, its representational capacity is limited by its constrained Transformer depth and smaller feature embeddings.

The hybrid model of Aboelenin et al. (2025) pushes representational richness further by fusing VGG16, Inception-V3, and DenseNet201 with a Vision Transformer head, reaching 97.87% accuracy. Nevertheless, the model carries roughly 150M parameters and a very large computational footprint (≈25 GFLOPs), producing a model of approximately 760 MB in size. Such complexity makes it unsuitable for real-time agricultural applications or embedded devices, where model size and inference time are critical.

The TwoConV-BiLSTM framework proposed by Prommakhot et al. (2025) takes a sequential hybrid approach, fusing CNN-based spatial extraction with bidirectional temporal modeling. It maintains an extremely compact configuration (≈0.5 M parameters, ≈0.8 GFLOPs, 4.6 MB) but delivers only 96.46% accuracy, showing that while lightweight, its temporal modeling alone is insufficient to capture fine-grained texture and color cues necessary for precise disease classification.

Overall, AgriViTNet successfully bridges the gap between performance and efficiency.

Statistical Analysis Across Multiple Runs

To ensure robustness and reproducibility, we retrained all compared approaches five times with different random seeds. This procedure allows evaluating not only the average performance but also the stability of each model under stochastic training conditions (weight initialization, data shuffling, and mini-batch sampling).

Table 15 reports the detailed results for each run (R1-R5), along with the Mean ± Standard Deviation. We further tested whether these differences were statistically meaningful using the Wilcoxon rank-sum test, comparing AgriViTNet against each competing model. The obtained p-values are all below 0.05, typically ranging between 0.007 and 0.01 depending on the metric. Therefore, this statistical validation strengthens the reliability of the reported performance gains.

Quantitative Results. AgriViTNet led every competing model on all four metrics without exception. The standard deviations of AgriViTNet are small across all metrics, indicating remarkable stability across different random seeds. In contrast, competing models exhibit slightly higher variability, particularly in Accuracy and F1-score.

Stability and Reproducibility Discussion. Beyond statistical significance, the consistency of results across runs highlights the strong reproducibility of AgriViTNet. The tight standard deviations across runs (never exceeding 0.044 for any metric) point to a model that converges the same way regardless of how weights are initialized or batches are shuffled. This is consistent with what the architecture is designed to do: The CNN and Transformer branches each extract a distinct type of feature rather than competing for the same signal, which

DOI: 10.3844/jcssp.2026.2736.2754

leaves less room for optimization to settle into different local minima from one run to the next.

Ablation Study

To better understand the contributions of each module in AgriViTNet, we conducted a comprehensive ablation study. All configurations maintain the final MLP classifier to ensure a fair comparison across variants. We evaluated six configurations of AgriViTNet. Table 16 summarizes the results of the ablation study, including parameter count, computational cost (GFLOPs), and classification accuracy.

Table 15: Statistical comparison across five independent runs with different random seeds

	Model	R1	R2	R3	R4	R5	Mean ± Std	Wilcoxon p-value
Accuracy(%)	AgriViTNet (Ours)	99,49	99,52	99,47	99,50	99,48	99.4920±0.0192	–
	MobileH-Transformer of Thai and Le (2025)	97,1	97,05	97,08	97,12	97,06	97.0820±0.0286	0.00794
	Hybrid model of Aboelenin et al. (2025)	97,87	97,91	97,84	97,88	97,9	97.8800±0.0274	0.00794
	TwoConV-BiLSTM of Prommakhot et al. (2025)	96,46	96,4	96,51	96,43	96,49	96.4580±0.0444	0.00794
Precision	AgriViTNet (Ours)	0,995	0,996	0,994	0,995	0,995	0.9950±0.0007	–
	MobileH-Transformer of Thai and Le (2025)	0,971	0,97	0,97	0,972	0,97	0.9706±0.0009	0.01018
	Hybrid model of Aboelenin et al. (2025)	0,978	0,979	0,978	0,978	0,979	0.9784±0.0005	0.00994
	TwoConV-BiLSTM of Prommakhot et al. (2025)	0,964	0,963	0,965	0,964	0,965	0.9642±0.0008	0.01066
Recall	AgriViTNet (Ours)	0,994	0,995	0,994	0,995	0,994	0.9944±0.0005	–
	MobileH-Transformer of Thai and Le (2025)	0,97	0,971	0,971	0,971	0,97	0.9706±0.0005	0.00970
	Hybrid model of Aboelenin et al. (2025)	0,978	0,979	0,977	0,978	0,978	0.9780±0.0007	0.00994
	TwoConV-BiLSTM of Prommakhot et al. (2025)	0,964	0,964	0,965	0,963	0,964	0.9640±0.0007	0.00994
F1-score	AgriViTNet (Ours)	0,994	0,995	0,994	0,995	0,994	0.9944±0.0005	–
	MobileH-Transformer of Thai and Le (2025)	0,97	0,97	0,97	0,971	0,97	0.9702±0.0004	0.00856
	Hybrid model of Aboelenin et al. (2025)	0,978	0,979	0,978	0,978	0,978	0.9782±0.0004	0.00856
	TwoConV-BiLSTM of Prommakhot et al. (2025)	0,964	0,963	0,965	0,963	0,964	0.9638±0.0008	0.01042

Table 16: Ablation study of AgriViTNet. The MLP classifier is maintained in all configurations

Configuration	Comments	Params (M)	GFLOPS	Accuracy (%)
Full AgriViTNet (CNN + Transformer + CBAM)	Reference	3,1	1,54	99,49
AgriViTNet without CBAM (CNN + Transformer)	Removes attention module → slightly less focused on disease regions	2,8	1,32	97,57
AgriViTNet without Transformer and CBAM (CNN only)	Removes global context and attention → some diseases with global patterns are less accurately recognized	2	0,84	96,5
AgriViTNet without CNN and CBAM (Transformer only)	Removes local feature extraction and attention → more sensitive to textures and color variations	1,6	0,67	95
AgriViTNet without Transformer (CNN + CBAM)	Keeps CNN + CBAM but removes global context → slight decrease for diseases with large-scale patterns	2,3	1,06	97,28
AgriViTNet without CNN (Transformer + CBAM)	Keeps Transformer + CBAM but removes local feature extractor → more sensitive to fine-grained textures	1,9	0,89	96,89

Several observations can be made. The full AgriViTNet achieves the highest accuracy (99.49%), confirming that combining CNN, Transformer, and CBAM yields the best performance. Removing the CBAM module results in a noticeable drop in accuracy (97.87%), highlighting its importance. Without the Transformer, the model loses its reach across the leaf, which leads to lower accuracy (96.50% with CNN only). Removing the CNN diminishes local feature extraction, making the model more sensitive to fine- grained textures and colors, decreasing accuracy to 95.00% (Transformer only).

Retaining combinations of two modules (CNN + CBAM or Transformer + CBAM) shows intermediate performance (98.8% and 97.0%), which confirms that both local and global feature extraction are essential for optimal performance.

Figure 7 summarizes the accuracy of each variant. This analysis provides quantitative evidence supporting the design choices of AgriViTNet and justifies its superior performance compared to simpler or partial variants.

Discussion

These results confirm that no single component of AgriViTNet is doing the heavy lifting alone. The 99.49% accuracy, stable across five random seeds with tight variance in precision, recall, and F1-score, comes from the three modules addressing distinct failure modes rather than duplicating each other's function a pattern most visible on the disease classes with subtle or overlapping symptoms, precisely where a single-mechanism model would be expected to struggle.

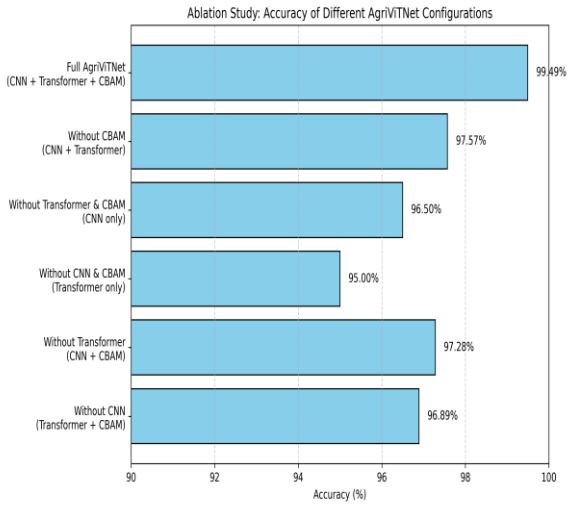


Fig. 7: Ablation study results showing the accuracy of different AgriViTNet configurations

The ablation results in Table 16 make this division of labor concrete. Color and texture cues within a lesion are picked up by the CNN backbone; symptoms that extend across separate regions of a leaf are only resolved once the Transformer is present; and the sharpest gains in disease-relevant focus, with background noise suppressed, come specifically from CBAM. Removing any one of the three costs accuracy in a way the other two cannot compensate for, which is the strongest evidence that the architecture's design choices were the right ones rather than incidental to the result.

The effectiveness of the CBAM module can be understood in relation to the specific characteristics of the 45-class multi-crop dataset used in this study. With 15 plant species and 45 disease categories, the dataset presents a high degree of inter-class visual similarity: For instance, necrotic spot patterns in Tomato Early Blight share textural characteristics with Corn Cercospora Leaf Spot, while vein discoloration symptoms overlap across multiple species. In such a setting, the Transformer encoder provides valuable global context by modeling spatial dependencies across the entire leaf surface, but it treats all spatial positions with broadly equal importance after self-attention. CBAM intervenes at this stage to introduce selective suppression: The channel attention mechanism down-weights feature channels that respond predominantly to background elements such as soil texture or illumination gradients, while the spatial attention mechanism localizes activation toward disease-specific regions. The pointed-out selectivity is particularly critical for visually confined symptoms such as those of Banana Black Sigatoka Disease and Tomato Spider Mites Two-spotted, where disease indicators occupy a small fraction of the leaf surface and would otherwise be attenuated by background activations. The ablation study reported in Table 16 quantifies this contribution precisely: Removing CBAM results in a 1.92 percentage point accuracy drop from 99.49% to 97.57%, despite reducing the parameter count by only 0.3M a disproportionate impact that confirms CBAM's role is synergistic with the Transformer output rather than merely additive. This synergy arises because CBAM operates on features that already encode global context, allowing it to selectively amplify disease-relevant patterns that the Transformer has distributed across the spatial representation.

Edge Deployment Feasibility

While working on a desktop GPU (NVIDIA with 10GB VRAM), we have performed comprehensive computational profiling to assess AgriViTNet's suitability for resource-constrained edge devices. Our profiling reveals that AgriViTNet maintains exceptionally low computational overhead.Computational Profiling Results: We profiled AgriViTNet on a GPU using batch sizes ranging from 1 to 32, as shown in Table 17. For batch size 1 (single-image inference, most relevant for edge deployment), AgriViTNet achieves a latency of 7.05 ms with a throughput of 141.81 images per second. Peak GPU memory usage during single-image inference is 152.23 MB, and the complete model size is 9.7 MB. These metrics confirm AgriViTNet's lightweight footprint and fast inference characteristics.

Table 17: Model complexity and inference profiling results

Batch Size	Latency (ms/image)	Throughput (images/s)	Peak Memory (MB)
1	7.05	141.81	152.23
4	6.77	147.76	518.35
8	7.21	138.61	1004.02
16	7.22	138.43	1978.73
32	7.61	131.48	3926.30

Estimated Performance on Edge Processors: Based on the measured GPU performance, simulations, and established scaling laws for ARM-based edge processors, we estimate the following deployment scenarios:

- Raspberry Pi 4B (4GB RAM, 1.5 GHz quad-core CPU): Estimated inference latency of 150-250 ms per image, enabling sustained throughput of 4-6 frames per second. This performance is sufficient for periodic disease monitoring applications or batch processing of collected leaf images. Peak memory consumption during inference was estimated at 85-110 MB, sustainable on 4GB systems
- NVIDIA Jetson Nano (4GB GPU VRAM, Maxwell architecture): With TensorRT optimization, we have obtained an estimated

inference latency of 80-120 ms per image, achieving 8-12 frames per second. This throughput supports near real-time continuous monitoring suitable for precision agriculture applications. GPU memory footprint was estimated at 150-200 MB, well within the Jetson Nano's 4GB GPU VRAM budget

- Mobile Devices (iOS/Android): With quantization to INT8 precision (reducing model size to ~2.5MB), AgriViTNet could achieve inference latencies of 50-100 ms on modern smartphones, enabling single-image disease diagnosis applications in the field

Comparison to State-of-the-Art models: Set against MobileH-Transformer Thai and Le (2025), Hybrid model Aboelenin et al. (2025), and TwoConv-BiLSTM Prommakhot et al. (2025), AgriViTNet achieves higher accuracy and F1-score, lower variability across runs, and improved robustness on difficult disease classes, while maintaining lower or comparable computational weight and size. Together, these figures place AgriViTNet in the narrow region of the accuracy-efficiency space that edge and mobile deployment actually require.

Practical Deployment Implications: AgriViTNet's combination of ultra-lightweight design (9.7 MB model, 3.1M parameters), fast inference (7 ms on GPU, estimated 80-250 ms on edge processors), and exceptional accuracy (99.49%) positions it as a practical solution for IoT-based plant disease monitoring. The model's footprint is orders of magnitude smaller than traditional CNNs, enabling deployment on commodity edge hardware widely available in agricultural settings. Our analysis demonstrates that AgriViTNet successfully bridges the historical accuracy-efficiency trade-off, delivering both high precision and genuine edge deployability a critical requirement for sustainable, scalable precision agriculture systems.

Despite these strengths, limitations remain. Performance may degrade on rare diseases or classes with very few training samples, and extreme conditions such as severe lighting variations, occlusions, or noisy backgrounds could challenge the model.

Conclusion

This work presents AgriViTNet, a substantial advancement in IoT-based plant disease detection systems. AgriViTNet demonstrates that the three architectural components at its core do not simply coexist but reinforce one another: The CNN backbone supplies the fine local detail that the Transformer alone would miss, the Transformer supplies the cross-region context that the CNN alone would miss, and CBAM sharpens the resulting representation toward disease-relevant regions before classification. Trained and evaluated on 86,138 images spanning 45 disease classes and 15 crop species, the model maintains its accuracy across this diversity rather than degrading on the less-represented classes, evidence that the architecture generalizes beyond narrow, single-crop benchmarks.

By pairing convolutional feature extraction with a Transformer's ability to relate distant regions of the leaf, the model avoids the blind spot each architecture has on its own: Missed texture on one side, missed spatial context on the other.

Set against MobileH-Transformer, TwoConV-BiLSTM, and the VGG-Inception-DenseNet hybrid, AgriViTNet is the only model in this comparison that combines ≈3.1M parameters compactness with above-99% accuracy. The other approaches each sacrifice one for the other, either falling below 98% accuracy to stay lightweight, or exceeding 100M parameters to stay accurate.

This matters in practice: At 9.7 MB and roughly 7 ms/image of inference latency on GPU, AgriViTNet fits comfortably within the memory and power budget of devices such as a Raspberry Pi or a Jetson Nano hardware already common on farms equipped for precision agriculture. AgriViTNet marks a meaningful step toward practical, on-site disease management solutions. Moving beyond incremental modifications, AgriViTNet exemplifies how we can innovatively tailor deep learning architectures for real-world precision agriculture, thus contributing significantly to sustainable farming practices and smart agricultural systems.

Several extensions could push AgriViTNet: Incorporating temporal and multimodal data like hyperspectral and thermal imagery, and exploring self-supervised pretraining for better generalization on small and imbalanced datasets. Furthermore, optimizing the model for on-device inference and integrating it into a real-time disease monitoring system will be key steps toward its deployment in precision agriculture and early disease management frameworks.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

This research did not receive external funding.

Author's Contributions

Nyabeye Pangop Doris-Khöler: Oversaw the design and implementation of the research project. Developed the proposed framework; performed image preprocessing and dataset preparation; led the architecture conception and implementation; wrote the initial version of the manuscript, later revising it based on co-author feedback.

Kengni Ngangmo Olga: Provided expert supervision, constructive feedback, and strategic research guidance throughout the project. Reviewed the methodological design, ensured the scientific rigor of the experimental procedures, and contributed to manuscript refinement and validation of the technical components.

Ethics

The authors confirm that no ethical issues are involved.

Conflicts of Interest

The authors declare no conflict of interest.

Data Availability

The two datasets fused are accessible as follows: New Plant Diseases Dataset (https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset) and Banana Disease Recognition Dataset (https://www.kaggle.com/datasets/sujaykapadnis/banana-disease-recognition-dataset). The script of AgriViTNet is available at https://github.com/Doris-Kholer/SmartEdgeAgriculture-AgriVitNet-model. The prototype of the pre-trained AgriViTNet is available at https://www.kaggle.com/models/doriskhlernyabeye/agrivitnet_2/PyTorch/default/1.

References

Aboelenin, S., Elbasheer, F. A., Eltoukhy, M. M., El-Hady, W. M., & Hosny, K. M. (2025). A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer. *Complex & Intelligent Systems*, *11*(2), 142. https://doi.org/10.1007/s40747-024-01764-x

Akhtar, M. N., Shaikh, A. J., Khan, A., Awais, H., Bakar, E. A., & Othman, A. R. (2021). Smart Sensing with Edge Computing in Precision Agriculture for Soil Assessment and Heavy Metal Monitoring: A Review. *Agriculture*, *11*(6), 475. https://doi.org/10.3390/agriculture11060475

Ali, M., Salma, M., Haji, M. E., & Jamal, B. (2025). Plant disease detection using vision transformers. *International Journal of Electrical and Computer Engineering (IJECE)*, *15*(2), 2334–2344. https://doi.org/10.11591/ijece.v15i2.pp2334-2344

da Silva, P. E. C., & Almeida, J. (2025). An Edge Computing-Based Solution for Real-Time Leaf Disease Classification Using Thermal Imaging. *IEEE Geoscience and Remote Sensing Letters*, *22*, 1–5. https://doi.org/10.1109/lgrs.2024.3456637

Dang, M., Wang, H., Li, Y., Nguyen, T.-H., Tightiz, L., Xuan-Mung, N., & Nguyen, T. N. (2024). Computer Vision for Plant Disease Recognition: A Comprehensive Review. *The Botanical Review*, *90*(3), 251–311. https://doi.org/10.1007/s12229-024-09299-z

Delfani, P., Thuraga, V., Banerjee, B., & Chawade, A. (2024). Integrative approaches in modern agriculture: IoT, ML and AI for disease forecasting amidst climate change. *Precision Agriculture*, *25*(5), 2589–2613. https://doi.org/10.1007/s11119-024-10164-7

Dewangan, O., & Vij, P. (2024). CNN-LSTM framework to automatically detect anomalies in farmland using aerial images from UAVs. *BIO Web of Conferences*, *82*, 05015. https://doi.org/10.1051/bioconf/20248205015

Dhiman, P., Kaur, A., Hamid, Y., Alabdulkreem, E., Elmannai, H., & Ababneh, N. (2023). Smart Disease Detection System for Citrus Fruits Using Deep Learning with Edge Computing. *Sustainability*, *15*(5), 4576. https://doi.org/10.3390/su15054576

Gawande, A., Sherekar, S., & Gawande, R. (2024). Early prediction of grape disease attack using a hybrid classifier in association with IoT sensors. *Heliyon*, *10*(19), e38093. https://doi.org/10.1016/j.heliyon.2024.e38093

Haque, Md. A., Deb, C. K., Gole, P., Karmakar, S., Dheeraj, A., Din, M. U. S., Dutta, S., Kumar, M. K. P., & Marwaha, S. (2025). An enhanced vision transformer network for efficient and accurate crop disease detection. *Expert Systems with Applications*, *283*, 127743. https://doi.org/10.1016/j.eswa.2025.127743

Li, X., Zhu, L., Chu, X., & Fu, H. (2020). Edge Computing-Enabled Wireless Sensor Networks for Multiple Data Collection Tasks in Smart Agriculture. *Journal of Sensors*, *2020*, 1–9. https://doi.org/10.1155/2020/4398061

Mahbub, M. (2020). A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. *Internet of Things*, *9*, 100161. https://doi.org/10.1016/j.iot.2020.100161

Morchid, A., Marhoun, M., Alami, R. E., & Boukili, B. (2024a). Intelligent detection for sustainable agriculture: A review of IoT-based embedded systems, cloud platforms, DL, and ML for plant disease detection. *Multimedia Tools and Applications*, *83*(28), 70961–71000. https://doi.org/10.1007/s11042-024-18392-9

Morchid, A., Oughannou, Z., Alami, R. E., Qjidaa, H., Jamil, M. O., & Khalid, H. M. (2024b). Integrated internet of things (IoT) solutions for early fire detection in smart agriculture. *Results in Engineering*, *24*, 103392. https://doi.org/10.1016/j.rineng.2024.103392

Nawaz, M., & Babar, M. I. K. (2025). IoT and AI for smart agriculture in resource-constrained environments: challenges, opportunities and solutions. *Discover Internet of Things*, *5*(1), 24. https://doi.org/10.1007/s43926-025-00119-3

Panchal, A. V., Patel, S. C., Bagyalakshmi, K., Kumar, P., Khan, I. R., & Soni, M. (2023). Image-based Plant Diseases Detection using Deep Learning. *Materials Today: Proceedings*, *80*, 3500–3506. https://doi.org/10.1016/j.matpr.2021.07.281

Pang, H., Zheng, Z., Zhen, T., & Sharma, A. (2021). Smart farming: An approach for disease detection implementing iot and image processing. *International Journal of Agricultural and Environmental Information Systems*, *12*(1), 55–67. https://doi.org/10.4018/ijaeis.20210101.oa4

Pintus, M., Colucci, F., & Maggio, F. (2025). Emerging Developments in Real-Time Edge AIoT for Agricultural Image Classification. *IoT*, *6*(1), 13. https://doi.org/10.3390/iot6010013

Prommakhot, A., Onshaunjit, J., Ooppakaew, W., Samseemoung, G., & Srinonchat, J. (2025). Hybrid CNN and Transformer-Based Sequential Learning Techniques for Plant Disease Classification. *IEEE Access*, *13*, 122876–122887. https://doi.org/10.1109/access.2025.3586285

Putrada, A. G., Oktaviani, I. D., Fauzan, M. N., & Alamsyah, N. (2024). CNN Pruning for Edge Computing-Based Corn Disease Detection with a Novel NG-Mean Accuracy Loss Optimization. *Telematika*, *17*(2), 68–83. https://doi.org/10.35671/telematika.v17i2.2899

Rahman, K. N., Banik, S. C., Islam, R., & Fahim, A. A. (2025). A real time monitoring system for accurate plant leaves disease detection using deep learning. *Crop Design*, *4*(1), 100092. https://doi.org/10.1016/j.cropd.2024.100092

Rajak, P., Ganguly, A., Adhikary, S., & Bhattacharya, S. (2023). Internet of things and smart sensors in agriculture: Scopes and challenges. *Journal of Agriculture and Food Research*, *14*, 100776. https://doi.org/10.1016/j.jafr.2023.100776

Ramana, B. R., Kalnoor, G., Devashish, M., & Sai, P. K. R. (2025). Deep Learning Based Mobile Application for Automated Plant Disease Detection. *IEEE Access*, *13*, 107917–107925. https://doi.org/10.1109/access.2025.3581099

Rashid, R., Aslam, W., Aziz, R., & Aldehim, G. (2024). An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models. *IEEE Access*, *12*, 23149–23162. https://doi.org/10.1109/access.2024.3357099

Selvam, L., Mary, S. R., Manimuthu, A., & Pandian, M. (2024). DCNN and LSTM based Plant Leaf Disease Detection Model For Edge IoT Device. *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, 1384–1389. https://doi.org/10.1109/ic2pct60090.2024.10486248

Shelar, N., Shinde, S., Sawant, S., Dhumal, S., & Fakir, K. (2022). Plant Disease Detection Using Cnn. *ITM Web of Conferences*, *44*, 03049. https://doi.org/10.1051/itmconf/20224403049

Shoaib, M., Sadeghi, A. N., Ali, F., Hussain, I., & Khalid, S. (2025). Leveraging deep learning for plant disease and pest detection: a comprehensive review and future directions. *Frontiers in Plant Science*, *16*, 1–25. https://doi.org/10.3389/fpls.2025.1538163

Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, *14*, 1158933. https://doi.org/10.3389/fpls.2023.1158933

Soussi, A., Zero, E., Sacile, R., Trinchero, D., & Fossa, M. (2024). Smart Sensors and Smart Data for Precision Agriculture: A Review. *Sensors*, *24*(8), 2647. https://doi.org/10.3390/s24082647

Thai, H.-T., & Le, K.-H. (2025). MobileH-Transformer: Enabling real-time leaf disease detection using hybrid deep learning approach for smart agriculture. *Crop Protection*, *189*, 107002. https://doi.org/10.1016/j.cropro.2024.107002

Uddin, M. A., Ayaz, M., Mansour, A., Aggoune, el-H. M., Sharif, Z., & Razzak, I. (2021). Cloud-connected flying edge computing for smart agriculture. *Peer-to-Peer Networking and Applications*, *14*(6), 3405–3415. https://doi.org/10.1007/s12083-021-01191-6

Vikhe, B. B., Patil, P. R., Bhaladhare, P. R., & Potgantwar, A. (2025). Image-based onion leaf disease identification using a CNN-Transformer hybrid approach. *International Journal of Image and Data Fusion*, *16*(1), 2561938. https://doi.org/10.1080/19479832.2025.2561938

Wang, W., & Kang, Y. (2025). A Review of Computer Vision Technologies in Precision Agriculture. *Theoretical and Natural Science*, *101*(1), 35–40. https://doi.org/10.54254/2753-8818/2025.ch22224

Yilmaz, E., Bocekci, S. C., Safak, C., & Yildiz, K. (2025). Advancements in smart agriculture: A systematic literature review on state-of-the-art plant disease detection with computer vision. *IET Computer Vision*, *19*(1), e70004. https://doi.org/10.1049/cvi2.70004