MobilePlantViT-LDA: A Lightweight Hybrid CNN-Transformer Model for Plant Leaf Disease Detection

Vaishnavi Abbanaboina
Student,CSE
Neil Gogte Institute of Technology
Hyderabad, India
vaishnavi.abbanaboina21@gmail.com

Sreeja Vaitla
Student,CSE
Neil Gogte Institue of Technology
Hyderabad, India
sreejavaitla@gmail.com

Astha Kumari
Student,CSE
Neil Gogte Institue of Technology
Hyderabad, India
009asthakumari@gmail.com

Gangisetti Himasree
Student,CSE
Neil Gogte Institute of Technology
Hyderabad, India
himarm2006@gmail.com

Sangineedi Chaitanya Satya Balaji
Student,CSE
Neil Gogte Institute of Technology
Hyderabad, India
chaitanyasatyabalaji@gmail.com

***Abstract*—This paper presents MobilePlantViT-LDA, an intelligent mobile-based plant leaf disease detection system that accurately classifies diseases and recommends the appropriate treatment to farmers. The system integrates image preprocessing, lightweight convolutional feature extraction, attention enhancement, transformer-based encoding, and a classification layer to predict disease types with confidence scores. Achieving 97.48% accuracy in the PlantVillage data set, the model outperforms the baseline networks while maintaining efficiency for mobile deployment. The source code is available in the GitHub repository: https://github.com/SCSBalaji/MobilePlantViT-LDA**

***Index Terms*—plant disease detection, deep learning, vision transformer, mobile deployment, convolutional neural networks, attention mechanism**

I. Introduction

This paper focuses on the design of an intelligent mobile-based plant leaf disease detection system that can accurately classify diseases and recommend appropriate treatment to farmers. Integrates image preprocessing, lightweight convolutional feature extraction, attention enhancement, transformer-based encoding, and a classification layer to predict the disease type along with confidence score.

A. Project Overview

Plant diseases threaten global agriculture, causing economic losses and food insecurity. The FAO reports that nearly 40% of crop yield is lost annually to plant diseases. Timely, accurate disease identification is vital for sustainable agriculture and food security.

Recent progress in deep learning has made it possible to automatically recognize plant diseases from images [1]. Convolutional Neural Networks (CNNs) have been successful [14], but they mainly extract local features and often miss broader patterns in plant images [9]. Vision Transformers (ViTs) address this by capturing global spatial relationships, which helps improve classification accuracy [2], [4].

Traditional ViT models are effective but require extensive computational and memory resources [5]. This makes them difficult to deploy in resource-constrained settings like mobile or edge devices [1]. To address these issues, this work proposes MobilePlantViT-LDA. It is a lightweight hybrid Vision Transformer architecture optimized for efficient plant disease classification [1], [3], [10].

Our model uses Ghost Convolutions for efficient feature extraction [1], Coordinate Attention to focus on important disease areas [7], and Fused-Inverted Residual Blocks to improve how features are represented [11]. We also use Linear Differential Attention to lower computational demands and a Bottleneck Feed Forward Network (FFN) to keep the model small without losing accuracy [1], [13]. Together, these features help MobilePlantViT-LDA balance accuracy, speed, and efficiency, making it practical for use on low-power devices in real-world agriculture [1], [5], [12].

B. Background and Motivation

The identification of plant diseases has traditionally relied on manual inspection by agricultural experts, a process that is time-consuming, subjective, and often inaccurate under field conditions. The introduction of artificial intelligence and computer vision has revolutionized this domain by enabling rapid and objective disease recognition. However, most state-of-the-art deep learning models remain computationally intensive, limiting their practicality for on-site diagnosis, particularly in rural regions where computational resources are limited. Vision Transformers have demonstrated exceptional capabilities in visual understanding by leveraging self-attention mechanisms. Nevertheless, their high computational overhead poses a barrier to real-time applications. There is thus a growing need for lightweight ViT architectures that retain the accuracy of transformer-based models while minimizing resource consumption. This research is motivated by the necessity to design an optimized Vision Transformer that maintains high accuracy in disease detection while remaining computationally efficient. MobilePlantViT-LDA addresses this challenge through architectural simplifications, hybrid feature learning, and parameter reduction, thereby enabling the deployment of AI-driven plant disease recognition systems in practical agricultural settings.

II. Literature Survey

A. Introduction to ViT

Vision Transformers (ViTs) represent a major step forward in computer vision. They change how machines understand and interpret images. Building on the success of transformers in natural language processing, ViTs use the same attention-based method for visual data. Instead of analyzing an image with traditional convolutional layers, they break it into small patches, turn these patches into sequences, and input them into a transformer that learns relationships between different visual areas. The main strength of ViTs is their self-attention mechanism. This feature lets the model capture long-range dependencies across the whole image. Unlike convolutional neural networks (CNNs), which handle information locally within a fixed receptive field, ViTs can focus on multiple regions at once. This ability helps them balance fine details with the overall structure of an image, leading to excellent results in tasks like image classification, segmentation, and object detection. Typically, Vision Transformers (ViTs) begin with pre-training on large image datasets like ImageNet or JFT. During this phase, they learn various visual patterns and representations. This pre-training gives them a strong base for understanding general visual ideas. Later, these models can be fine-tuned for specific tasks, allowing them to adapt effectively to different problem settings. Due to this wide-ranging learning process that captures global relationships within images, Vision Transformers (ViTs) often demonstrate higher accuracy and greater flexibility than traditional CNNs across many computer vision applications. However, they require significant computational power and memory. This makes it hard to use them on low-power or mobile devices. To solve these problems, researchers have developed more efficient hybrid models, including MobileViT, EfficientFormer, and MobilePlantViT. These models mix convolutional layers with transformer components. They maintain the efficiency of CNNs while also using the deep understanding provided by transformers. As a result, Vision Transformers are increasingly used not just in traditional computer vision tasks but also in specialized areas like plant disease detection, medical imaging, and remote sensing, where capturing subtle visual details is crucial.

B. Existing Works

M.R.Tonmoy et al. [1] propose MobilePlantViT, a lightweight hybrid Vision Transformer for plant disease image classification on mobile and edge devices. The model combines depthwise separable convolutions, a hierarchical 1–2–4 group convolution structure, and CBAM-based attention to efficiently extract and refine disease-relevant features. Patch embeddings are generated using lightweight convolutions and processed by a transformer encoder employing linear self-attention to reduce computational complexity. With only 0.69M parameters, MobilePlantViT achieves high accuracy across multiple plant disease datasets and outperforms lightweight MobileViT variants with larger models. The results show that integrating hierarchical convolutions, explicit attention, and efficient transformer design enables accurate and resource-efficient plant disease recognition for real-world applications.

A. Ouamane, A. Chouchane, Yassine Himeur, and their team [2] present an optimized Vision Transformer (ViT) framework for accurate plant disease detection. Instead of proposing a new architecture, the core contribution of this work lies in a systematic optimization of ViT hyperparameters, including patch size, image resolution, embedding dimension, model depth, number of attention heads, and MLP dimension. Through extensive experiments on the PlantVillage dataset, the authors identify an optimal configuration (224×224 image size, patch size 16, embedding dimension 512, depth 6, 8 attention heads, and MLP dimension 1024) that achieves 99.77% accuracy. The model leverages standard multi-head self-attention to capture global disease patterns and incorporates saliency map visualizations to improve interpretability. Cross-dataset validation on Taiwan Tomato and BananaLSD datasets demonstrates strong generalization, while comparisons with CNN models such as VGG19 and AlexNet show superior accuracy with significantly fewer parameters and storage requirements. The results confirm that careful ViT parameter optimization can outperform traditional CNNs and state-of-the-art methods, making ViTs a powerful and adaptable solution for plant disease detection.

Q. Meng, Y. Li and Z. Wang et al. [3] propose a dual-branch hybrid architecture that combines Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) for accurate and lightweight crop disease classification. The model consists of a parallel CNN branch for local feature extraction and a Transformer branch for global context modeling, with a learnable weighted fusion mechanism that adaptively balances the contributions of both branches. To address the high computational cost of standard Transformers, the authors introduce an Aggregated Local Perception Feed-Forward Network (ALP-FFN), which injects spatial locality into the Transformer encoder using depthwise convolutions, along with a linear (separable) self-attention mechanism to reduce complexity. This design enables effective global–local feature interaction while maintaining a lightweight structure. The proposed model achieves 99.71% accuracy on the PlantVillage dataset and 98.78% on the Potato Leaf dataset with only 4.9M parameters and 0.62 GFLOPs, outperforming several CNN- and ViT-based state-of-the-art models. The results demonstrate that adaptive feature fusion and locality-aware Transformer design significantly improve both accuracy and efficiency for crop disease recognition in resource-constrained environments.

Murugavalli S and Gopi R [4] propose PLA-ViT (Precision Leaf Analysis with Vision Transformers), a transformer-based framework for accurate plant leaf disease detection in precision agriculture. The core idea of PLA-ViT is to replace purely CNN-driven feature extraction with Vision Transformer self-attention, enabling effective modeling of long-range spatial dependencies that are often missed by convolutional networks. The pipeline combines image pre-processing, patch embedding, positional encoding, and multi-head self-attention–based transformer encoders, followed by classification and disease localization modules. Transfer learning with pre-trained ViTs, along with data augmentation and adaptive learning rate scheduling, improves generalization on limited and imbalanced datasets. Experimental results show that PLA-ViT outperforms CNN-based and hybrid models such as DLMC-Net, CycleGAN, and ESDNN in terms of classification accuracy, disease localization performance, inference time, and computational efficiency. The integration of transformer-based global context modeling with IoT-enabled monitoring highlights PLA-ViT as a scalable and interpretable solution for real-time plant disease diagnosis in smart agriculture systems.

Guoqiang Li, Yuchao Wang, Qing Zhao, and their team propose [5] PMVT (Plant-based Mobile Vision Transformer), a lightweight Vision Transformer architecture designed for real-time plant disease identification on mobile devices. PMVT builds upon MobileViT by replacing standard convolution blocks with an inverted residual structure using a larger 7×7 convolution kernel, enabling better modeling of long-distance dependencies between spatially separated leaf regions. To further enhance feature discrimination, a Convolutional Block Attention Module (CBAM) is integrated into the ViT encoder, allowing the network to emphasize disease-relevant features while suppressing irrelevant background information. This combination improves global context modeling without significantly increasing computational cost. The model achieves strong performance across multiple agricultural datasets, including wheat, coffee, and rice, reaching 93.6% accuracy with only 0.98M parameters, and consistently outperforming lightweight CNNs such as MobileNetV3 and SqueezeNet. The results demonstrate that incorporating larger convolution kernels and attention mechanisms into MobileViT yields a highly efficient and accurate backbone suitable for deployment in resource-constrained agricultural applications.

Weidong Zhang et al. [6] propose DBCLNet, a dual-branch collaborative learning network for accurate crop disease identification. The key idea of DBCLNet is to jointly capture fine-grained and coarse-grained features using a dual-branch convolutional module with different kernel scales, enabling effective extraction of local details and global texture information. Each branch integrates a channel attention mechanism, using max pooling for fine-grained features and average pooling for coarse-grained features, to emphasize disease-relevant channels. Multiple dual-branch modules are stacked into a feature cascaded module (FCM), allowing the network to progressively learn more abstract representations. To address class imbalance in real-world agricultural datasets, the model employs a focal loss function instead of standard cross-entropy. Experimental results on the PlantVillage dataset show that DBCLNet achieves 99.89% accuracy while maintaining moderate computational complexity, outperforming CNN-, attention-, and transformer-based baselines. The results demonstrate that collaborative multi-scale feature learning combined with channel attention significantly improves crop disease recognition performance.

Z. Salman, A. Muhammad, and D. Han propose [7] a Vision Transformer with Mixture of Experts (ViT–MoE) framework for plant disease classification in real-world (“in-the-wild”) conditions. The key idea is to combine a ViT backbone for global feature extraction with a Mixture of Experts classifier, where multiple lightweight expert networks specialize in different visual characteristics such as texture, color, and disease severity, and a gating mechanism dynamically selects and weights experts per input. To improve robustness and prevent expert domination, the model introduces entropy, orthogonal, and usage regularization, ensuring balanced expert utilization and diverse feature learning. Extensive experiments on PlantVillage and PlantDoc datasets show that the proposed ViT–MoE significantly improves cross-domain generalization, achieving a 20% accuracy gain over standard ViT and 68% accuracy in PlantVillage-to-PlantDoc transfer, outperforming CNNs such as InceptionV3 and EfficientNet. These results demonstrate that adaptive expert routing combined with transformer-based global context modeling is highly effective for handling diverse and uncontrolled agricultural imaging conditions.

Sherihan Aboelenin et al. [8] propose a hybrid CNN–Vision Transformer framework for accurate plant leaf disease detection and classification. The model employs an ensemble of pre-trained CNNs (VGG16, Inception-V3, and DenseNet201) to extract rich and complementary deep features, which are concatenated and passed to a Vision Transformer (ViT) for modeling long-range spatial dependencies through multi-head self-attention. This design addresses the limitation of CNNs in capturing global context while leveraging their strength in local feature extraction. Evaluated on the PlantVillage Apple and Corn datasets, the proposed framework achieves 99.24% accuracy on Apple leaves and 98% on Corn leaves, outperforming standalone CNNs and recent hybrid approaches. The results demonstrate that feature-level ensemble learning combined with transformer-based global modeling significantly improves multi-class plant disease classification performance.

Sumaya Mustofa et al. [9] present a comprehensive review of deep learning techniques for plant leaf disease detection, with a strong focus on the growing role of Vision Transformers (ViTs) in agricultural pathology. The paper systematically analyzes CNNs, DCNNs, ViT-based models, and hybrid architectures (CNN+ViT), along with specialized methods such as YOLO, RSNSR-LDD, SLViT, PlantXViT, and attention-enhanced transformers. Using PRISMA-based screening, the authors review studies published between 2018 and 2023 across multiple public datasets, comparing performance using metrics such as accuracy, precision, recall, and F1-score. The review highlights that ViTs consistently outperform traditional CNNs by capturing global contextual relationships, while hybrid CNN–ViT models offer a better balance between local feature extraction and global modeling. It also identifies key limitations in existing research, including dataset imbalance, dependence on publicly available data, and limited real-world generalization. Finally, the paper outlines future research directions such as hybrid transformer architectures, attention pruning, reinforcement learning, and privacy-preserving approaches to advance smart agriculture systems.

Kemal Celik [10] proposes a lightweight Vision Transformer–based framework for agricultural land use and land cover classification using remote sensing imagery. The work focuses on optimizing ViT architectures through a combination of model pruning and advanced data augmentation to achieve high accuracy under limited computational resources. Standard ViT-Base-16 and lightweight variants such as DeiT-Tiny and EfficientNet-B0 are evaluated on benchmark datasets including EuroSAT, NWPU-RESISC45, and SIRI-WHU, demonstrating strong global spatial modeling capability. A key contribution is the structured pruning of 50% of ViT encoder layers, which significantly reduces model complexity while maintaining competitive accuracy (97.9% on SIRI-WHU). Additionally, a hybrid augmentation strategy using CutMix and Cutout improves robustness and generalization, especially under limited data settings. The results confirm that optimized and compressed ViT models provide an effective and scalable solution for precision agriculture and real-world remote sensing applications.

Atika Apriani et al. [11] present a CNN-based automatic classification system for rice leaf disease detection aimed at supporting intelligent agricultural diagnostics. The model is trained on 15,030 rice leaf images spanning eight disease classes and one healthy class, using an 80:10:10 train–validation–test split with image normalization and data augmentation. The CNN architecture consists of three convolution–pooling blocks followed by fully connected layers, enabling effective hierarchical feature extraction without manual segmentation. Trained for 13 epochs using the Adam optimizer, the model achieves 89.30% training accuracy, 85.64% validation accuracy, and 84.52% test accuracy. Confusion-matrix analysis shows strong performance on visually distinct diseases such as Tungro and Bacterial Leaf Blight, while misclassifications mainly occur among diseases with similar visual patterns. The results demonstrate that standard CNN architectures can reliably classify rice leaf diseases but also highlight limitations in handling subtle inter-class similarities and complex field conditions.

S.Li, J.Zhang, and the team et al. [**?**] propose Plant-CNN-ViT, an ensemble-based deep learning framework for plant leaf classification under limited training data conditions. The key idea is to combine the complementary strengths of Vision Transformer (ViT), ResNet-50, DenseNet-201, and Xception by concatenating their deep feature representations before final classification. ViT captures global contextual dependencies through self-attention, while ResNet-50, DenseNet-201, and Xception provide robust hierarchical and fine-grained feature extraction using residual, dense, and depthwise separable convolutions, respectively. This ensemble design significantly improves feature diversity and generalization, especially on small datasets. Evaluated on four benchmark leaf datasets (Flavia, Folio Leaf, Swedish Leaf, and MalayaKew Leaf), Plant-CNN-ViT achieves near-perfect accuracy (up to 100%), outperforming standalone CNNs and ViT models. The results demonstrate that multi-model feature fusion effectively addresses data scarcity and boosts plant classification performance, albeit with increased computational cost.

Sankar Murugesan et al. [13] propose a Hybrid ConvNet–ViT framework for robust multiclass crop leaf disease classification across banana, cherry, and tomato leaves. The key contribution lies in fusing convolutional layers for local texture and edge extraction with Vision Transformer encoders for global contextual modeling, enabling effective recognition of visually similar and spatially distributed disease symptoms. The model uses a convolutional stem to generate dense feature maps, which are tokenized and processed through multi-head self-attention, followed by a lightweight classification head. Extensive experiments against state-of-the-art models such as EfficientNetV2, ConvNeXt, Swin Transformer, and ViT show that the proposed hybrid model consistently outperforms all baselines, achieving 99.29% test accuracy and strong precision, recall, and F1-scores under 5-fold cross-validation. Additionally, Grad-CAM visualizations confirm that the model attends to disease-relevant regions, improving interpretability and trustworthiness. The results demonstrate that ConvNet–Transformer fusion offers superior accuracy, robustness, and generalization for real-world plant disease diagnosis compared to standalone CNN or ViT architectures.

Muhammad Shafay et al. [14] present a comprehensive systematic review of deep learning–based plant disease detection using RGB and hyperspectral imaging (HSI) modalities, focusing on their evolution, comparative strengths, and real-world deployment challenges. Unlike prior surveys that study RGB and HSI in isolation, this work introduces a unified comparative framework that evaluates classical methods, machine learning, CNNs, and transformer-based architectures across 11 benchmark datasets. The analysis reveals a significant performance gap between lab conditions (95–99% accuracy) and field deployment (70–85%), highlighting the limitations of conventional CNNs under environmental variability. Transformer-based models, particularly Swin Transformers, demonstrate superior robustness, achieving 88% accuracy in real-world settings compared to 53% for CNNs. The review identifies key deployment constraints—including environmental sensitivity, high sensor costs, and interpretability requirements—and emphasizes the importance of lightweight architectures, cross-geographic generalization, explainable models, and multimodal RGB–HSI fusion. The paper provides evidence-based guidelines and future research directions aimed at translating plant disease detection systems from controlled research prototypes to scalable, real-world agricultural solutions.

III. Proposed Work

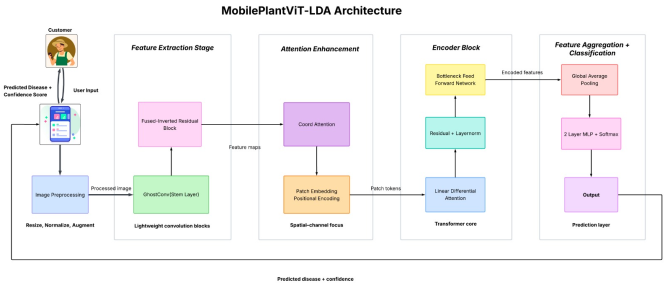
MobilePlantViT-LDA Architecture
Customer
Predicted Disease + Confidence Score
User Input
Feature Extraction Stage
Fused-Inverted Residual Block
GhostConv(Stem Layer)
Lightweight convolution blocks
Image Preprocessing
Resize, Normalize, Augment
Processed image
Feature maps
Attention Enhancement
Coord Attention
Patch Embedding Positional Encoding
Spatial-channel focus
Patch tokens
Encoder Block
Bottleneck Feed Forward Network
Residual + Layernorm
Linear Differential Attention
Transformer core
Encoded features
Feature Aggregation + Classification
Global Average Pooling
2 Layer MLP + Softmax
Output
Prediction layer
Predicted disease + confidence

Fig. 1: Architecture Diagram of the Proposed Work

A. Data Collection

Dataset for training our model was chosen with careful consideration. The PlantVillage dataset was selected because it contains over 54,000 leaf images spanning up to 38 classes across 26 plant species, including both healthy and infected leaves. This was used to train and validate the proposed deep learning model. This dataset was selected due to its large sample size, high-quality images, balanced class distribution, and benchmark status in agricultural disease research. It allows the model to learn a wide variety of infection patterns, improving prediction accuracy on real-world leaf scans. It was taken from Kaggle and many previous deep learning models such as MobilePlantViT used this dataset to train their models.

The dataset mainly consists of high-resolution RGB images, multiple plants under diverse disease conditions, and clear annotation of disease class for supervised learning. The diversity and scale of the PlantVillage dataset make it highly suitable for mobile-based leaf disease diagnosis. To ensure robustness and generalization, data augmentation was applied to simulate real farm scenarios and the dataset was split into three categories as shown in Table I.

TABLE I: Dataset Configuration

Property	Value
Source	PlantVillage (Kaggle) [16] [17]
Original Size	54,304 images
Classes	38
Preprocessing	Corruption check & pHash deduplication
Final Refined Size	49,370 images
Train (70.0%)	34,539
Val (15.0%)	7,388
Test (15.1%)	7,443

B. Preprocessing and Augmentation

The workflow begins with the insertion of raw image data, which undergoes a preprocessing pipeline to ensure model robustness.

The process of cleaning and preparing raw data into machine understandable format is known as pre-processing. It will change the data into a format that makes it easier for the model to understand and process. The model's performance is greatly influenced by the pre-processed data.

The main preprocessing method used was image resizing, normalization and augmentation where all images were normalized using the standard ImageNet statistics:

- **Mean:** (0.485, 0.456, 0.406)
- **Standard Deviation:** (0.229, 0.224, 0.225)

Augmentation was performed to artificially increase dataset diversity and improve model robustness to real-world farm conditions as shown in Table II.

TABLE II: Data Augmentation

Augmentation	Parameters
Horizontal flip	50%
Random rotation	±30°
Random crop	224 × 224 from 256
Affine transform	trans.=0.1, scale=(0.9,1.1), shear=10
Color jitter	bright.=0.2, cont.=0.2, sat.=0.2, hue=0.1
Random perspective	20% distortion, $p = 0.5$
Normalization	μ=[0.485,0.456,0.406], σ=[0.229,0.224,0.225]

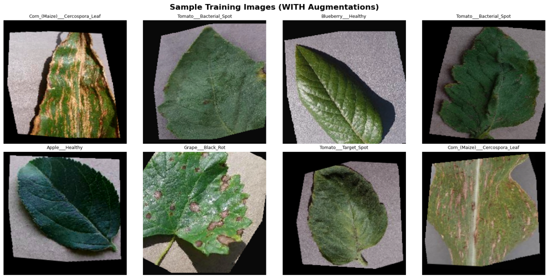

Fig. 2: **Sample Training Images (WITH Augmentations)** showing Corn cercospora leaf, Tomato bacterial spot, Blueberry healthy, Tomato bacterial spot, Apple healthy, Grape black rot, Tomato target spot and Corn cercospora leaf.

These transformations help the model generalize better by simulating natural leaf appearance variations such as different camera angles, lighting changes, scaling and orientation differences, partial occlusion, and background disturbance.

The sample images represent multiple plant species and disease types, including: Corn (Cercospora Leaf Spot), Tomato (Bacterial Spot / Target Spot), Grape (Black Rot), Apple (Healthy), Blueberry (Healthy).

IV. Methodology

A. Feature Extraction Stage

The Feature Extraction Stage serves as the backbone of the model and operates directly on the preprocessed input tensor $x_0 \in \mathbb{R}^{H \times W \times 3}$, where the image has already been resized and normalized. Its primary goal is to transform this input into a hierarchy of rich, compressed feature maps that capture visual patterns such as edges, textures, and high-level shapes while progressively reducing spatial resolution and increasing channel depth.

1) GhostConv (Stem Layer): The Stem layer acts as the initial feature extractor and performs the first spatial downsampling while expanding the channel dimension. Instead of using a standard convolution, it employs a Ghost module that decomposes feature generation into intrinsic and inexpensive ghost features. Formally, the stem produces an intrinsic feature map

$$U = W_s * x_0, \quad U \in \mathbb{R}^{sH \times sW \times C_{\text{int}}} \tag{1}$$

where W_s is a standard convolution kernel, s denotes the stride, and C_{int} represents the number of intrinsic channels. Cheap linear operations $\{\phi_k(\cdot)\}_{k=1}^{K}$ (e.g., depthwise filters or spatial shifts) are then applied to generate additional ghost feature maps:

$$G_k = \phi_k(U), \quad k = 1, \ldots, K. \tag{2}$$

These feature maps are concatenated with the intrinsic features to form the final stem output feature map:

$$F_{\text{stem}} = \text{Concat}\left(U, G_1, \ldots, G_K\right). \tag{3}$$

This mechanism significantly reduces the number of parameters and floating-point operations (FLOPs) compared to a single dense convolution that directly produces the same number of output channels.

2) Fused-Inverted Residual Blocks: After the stem stage, the feature tensor F_{stem} is passed through a lightweight convolutional block implemented as a *Fused-Inverted Residual (FIR)* unit, corresponding to the *Fused-Inverted Residual Block* shown in the architecture diagram. The inverted residual design follows a Narrow → Wide → Narrow transformation, where a low-dimensional input is first expanded, spatially processed, and then projected back to a compact bottleneck representation.

Given an input feature map $x \in \mathbb{R}^{H' \times W' \times C_{\text{in}}}$, the FIR block can be formulated as

$$z = \sigma(\text{BN}(W_{\text{fused}} * x)), \tag{4}$$

$$y = W_{\text{proj}} * z, \tag{5}$$

where W_{fused} denotes a single 3×3 convolution that jointly performs channel expansion and spatial feature extraction, W_{proj} represents a 1×1 convolution projecting the features to C_{out} channels, $\sigma(\cdot)$ is a nonlinear activation function (e.g., SiLU), and BN denotes batch normalization.

When $C_{\text{in}} = C_{\text{out}}$ and the stride equals 1, a residual connection is employed to facilitate information preservation and stable gradient propagation, yielding

$$y = x + y. \tag{6}$$

The resulting output feature map $F_{\text{FIR}} = y$ is forwarded directly to the subsequent Attention Enhancement module, consistent with the single forward connection from the Fused-Inverted Residual Block to the Coordinate Attention stage in the architecture diagram. All of these blocks send an arrow to the right, linking to the Attention Amplification stage which means Multi-Scale Feature Extraction. It doesn't wait for the final output, but it extracts "Feature maps" at different resolutions (scales).

B. Attention Enhancement Stage

This module serves as a bridge between local CNN representations and the global Transformer encoder by operating directly on the output feature map of the Fused-Inverted Residual block. It refines the extracted features using Coordinate Attention, which enhances channel attention by embedding explicit positional information, enabling the network to attend to both *what* features are important and *where* disease-related cues appear on the leaf surface.

Let the input feature map be $F_{\text{FIR}} \in \mathbb{R}^{H' \times W' \times C}$. Coordinate Attention first applies two independent one-dimensional global pooling operations along the horizontal and vertical spatial directions:

$$f_h(c, x) = \frac{1}{H'} \sum_{y=1}^{H'} F_{\text{FIR}}(y, x, c), \tag{7}$$

$$f_v(c, y) = \frac{1}{W'} \sum_{x=1}^{W'} F_{\text{FIR}}(y, x, c), \tag{8}$$

which generate two context descriptors $f_h \in \mathbb{R}^{1 \times W' \times C}$ and $f_v \in \mathbb{R}^{H' \times 1 \times C}$ that preserve spatial information along a single axis.

These descriptors are passed through shared transformation layers and then separated into horizontal and vertical attention maps, denoted as a_h and a_v. The attention maps are subsequently broadcast and applied to the original feature map as

$$\tilde{F}(y, x, c) = F_{\text{FIR}}(y, x, c) \cdot a_h(c, x) \cdot a_v(c, y), \tag{9}$$

resulting in a refined feature tensor $\tilde{F}$, where channel responses are adaptively reweighted based on their spatial importance.

In the Patch Embedding and Positional Encoding stage, the refined feature map $\tilde{F}$ is converted into a sequence of patch tokens suitable for transformer processing. The feature map is partitioned into non-overlapping patches of size $P \times P$. Each patch is flattened and linearly projected using a learnable embedding matrix $W_p \in \mathbb{R}^{(P^2 C) \times D}$, producing patch-level tokens:

$$z_i = W_p \,\text{vec}\Big(\tilde{F}^{(i)}\Big), \quad z_i \in \mathbb{R}^D, \quad i = 1, \ldots, N, \tag{10}$$

where $N = \frac{H'W'}{P^2}$ denotes the total number of patches and D is the token embedding dimension.

To preserve spatial ordering information lost during flattening, a learnable positional encoding $p_i \in \mathbb{R}^D$ is added to each token, yielding the initial transformer input sequence:

$$x_i^{(0)} = z_i + p_i, \quad i = 1, \ldots, N. \tag{11}$$

C. Transformer Core (Encoder Block)

The transformer core captures long-range global dependencies among patch tokens that are difficult to model using convolutional operations alone. It is implemented as a lightweight transformer encoder composed of Linear Differential Attention (LDA), residual connections with layer normalization, and a bottleneck Feed Forward Network (FFN) optimized for mobile efficiency.

Given the input token sequence

$$X^{(0)} = \left[x_1^{(0)}, \dots, x_N^{(0)}\right] \in \mathbb{R}^{N \times D}, \tag{12}$$

each encoder layer produces an updated representation $X^{(\ell)}$ by sequentially applying LDA-based self-attention and a bottleneck FFN. Residual connections and layer normalization are employed around both sublayers to stabilize training and maintain effective information flow.

1) Linear Differential Attention: In standard Transformer architectures such as Vision Transformers (ViT), the self-attention mechanism is defined as

$$\mathrm{Att}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{13}$$

where $Q, K, V \in \mathbb{R}^{N \times d_k}$ represent the query, key, and value matrices, respectively. This formulation incurs a computational complexity of $\mathcal{O}(N^2)$ with respect to the number of patches N, which is impractical for deployment on resource-constrained edge devices.

Linear Differential Attention reformulates self-attention into a linear $\mathcal{O}(N)$ operation while enhancing discriminative capability through a differential mechanism. First, the query and key matrices are transformed using a kernel mapping function $\phi(\cdot)$:

$$\tilde{Q} = \phi(Q), \quad \tilde{K} = \phi(K). \tag{14}$$

This enables attention computation in a factored linear form:

$$\mathrm{LinAtt}(Q,K,V) = \tilde{Q}\left(\tilde{K}^\top V\right), \tag{15}$$

which avoids explicit construction of the $N \times N$ similarity matrix.

To further emphasize informative disease-related patterns, LDA computes two linear attention maps with distinct parameterizations, LinAtt_A and LinAtt_B, and combines them through subtraction:

$$\mathrm{LDA}(Q,K,V) = \mathrm{LinAtt}_A(Q,K,V) - \mathrm{LinAtt}_B(Q,K,V). \tag{16}$$

This differential formulation suppresses common background responses while amplifying salient disease-specific features.

Together, kernel-based linearization and differential attention reduce the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ and improve robustness to noise, enabling accurate and efficient global modeling of plant disease patterns on mobile and edge hardware.

Working Mechanism:

In the first step, Dual attention generation is used where instead of generating one attention map, the LDA block generates two separate linear attention maps.

$$\mathrm{Attention}_{\mathrm{LDA}}(Q,K,V) = \frac{\phi(Q_{\lambda_1})\left(K_{\lambda_1}^T V\right)}{\phi(Q_{\lambda_1})\sum K_{\lambda_1}^T} - \lambda \frac{\phi(Q_{\lambda_2})\left(K_{\lambda_2}^T V\right)}{\phi(Q_{\lambda_2})\sum K_{\lambda_2}^T} \tag{17}$$

Where: $\phi(\cdot)$ is the kernel function (typically $ELU(x) + 1$) used to linearize the complexity from $O(N^2)$ to $O(N)$. The λ parameter represents the differential scaling factor that helps cancel out common background noise. In the third step, irrelevant background information (like the sky or soil in a plant image) usually appears similarly in both maps. When you subtract them ($A - B$), this common noise cancels out or becomes close to zero.

TABLE III: Comparison of LDA over other mechanisms

Feature	Standard	Linear	LDA
Complexity	High (N^2)	Low (N)	**Low** (N)
Focus Ability	High	Low	**Very High**
Noise Handling	Poor	Average	**Excellent**
Best For	GPUs/Servers	Mobile Tasks	**Edge Devices**

2) Residual Connections and Layer Normalization: In deep networks, gradients (learning signals) often get smaller and smaller as they travel back from the output to the input, eventually reaching zero (vanishing). The Residual connection creates a path for gradients to flow backward smoothly, allowing the earlier layers (like GhostConv stem) to learn effectively. This is done using:

$$Y = \mathrm{Concat}([X \cdot f_{\mathrm{primary}},\ \Phi_{i,j}(y_i')]) \tag{18}$$

Where: * y_i' is the i-th intrinsic feature map generated by a standard convolution $f_{primary}$,$\Phi_{i,j}$ represents the j-th linear transformation (e.g., a 3×3 depthwise convolution) used to generate the "ghost" maps.

Layer Normalization ensures that the inputs to the subsequent Feed Forward Network (FFN) lie within a stable and consistent range by normalizing features across the hidden dimension for each token independently. For a given token, it computes the mean μ and variance σ^2 of the feature activations and applies normalization as follows:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta, \tag{19}$$

where x denotes the input activation, μ and σ^2 are the mean and variance of the input features, respectively, ϵ is a small constant added for numerical stability, and γ and β are learnable scale and shift parameters.

In the attention sub-layer, Layer Normalization is applied after the residual connection, yielding

$$\mathrm{Output} = \mathrm{LayerNorm}(\mathrm{Attention}(x) + x), \tag{20}$$

which corresponds to the residual-plus-normalization structure illustrated after the Linear Differential Attention module in the architecture diagram.

3) Bottleneck Feed Forward Network: In standard transformer architectures, the Feed Forward Network (FFN) expands the input dimension d to a larger hidden size (typically $4d$), applies a nonlinear activation, and then projects the features back to dimension d. In contrast, the Bottleneck FFN in MobilePlantViT-LDA adopts a reduced expansion ratio (e.g., $2d$) and may incorporate depthwise separable convolutions to improve computational efficiency. The FFN operation is defined as

$$\mathrm{FFN}(x) = W_2\,\sigma(W_1 x)\,, \tag{21}$$

where $W_1 \in \mathbb{R}^{d\times d_{\text{bottleneck}}}$, $W_2 \in \mathbb{R}^{d_{\text{bottleneck}}\times d}$, and $d_{\text{bottleneck}} \ll 4d$. This bottleneck design substantially reduces parameter count and computational overhead while retaining sufficient nonlinear modeling capacity for accurate plant disease classification on mobile and edge devices.

TABLE IV: Comparison of Bottleneck over Standard FFNs

Feature	Standard FFN	Bottleneck FFN
Parameter Count	High	Low
Computational cost	Expensive	Cheap
Overfitting Risk	High on small datasets	Low
Performance	Great for servers	Great for mobile

D. Algorithm: Hybrid CNN–Attention–Transformer Model

1. Input Stage – User and Image Preprocessing

Input: The user provides an image (e.g., a leaf or crop photograph) through the mobile application interface.

Steps:

- Receive the raw RGB image from the user interface and convert it into a tensor $x_0 \in \mathbb{R}^{H\times W\times 3}$.
- **Preprocess** the image by resizing it to a fixed resolution (e.g., 224×224), normalizing pixel values to a standard range, and optionally applying data augmentation techniques (rotation, flipping, color jitter, etc.) during training.
- Output a preprocessed image tensor x_0' that is ready for feature extraction.

2. Feature Extraction Stage – GhostConv Stem and Fused-Inverted Residual Block

Goal: Extract informative convolutional features in a lightweight manner before passing them to the Coordinate Attention module.

Process:

- Pass x_0' through the **GhostConv (Stem Layer)**, which performs initial downsampling and channel expansion to produce the stem feature map F_{stem}.
- Feed F_{stem} into a single **Fused-Inverted Residual Block**, which expands channels, applies a fused 3×3 convolution for spatial feature extraction, and projects the features back to a bottleneck representation, yielding F_{FIR}.
- Forward F_{FIR} directly to the **Attention Enhancement Stage** (Coordinate Attention) as the sole convolutional feature map output of this stage, consistent with the MobilePlantViT-LDA architecture diagram.

3. Attention Enhancement – Coordinate Attention Module

Goal: Enhance the most informative spatial and channel-wise features in the single convolutional feature map produced by the feature extraction stage.

Steps:

- Feed the feature map F_{FIR} from the Fused-Inverted Residual Block into the **Coordinate Attention** module, which performs separate one-dimensional global pooling operations along the horizontal and vertical directions to preserve positional information.
- Coordinate Attention decomposes channel attention into spatially aware coordinate embeddings and generates horizontal and vertical attention maps that reweight the original feature map.
- The output is a refined, attention-enhanced feature map $\tilde{F}$ that emphasizes disease-relevant regions on the leaf while suppressing less informative background responses.

4. Patch Embedding and Positional Encoding

Goal: Convert the attention-refined spatial feature map into a sequence of patch tokens suitable for transformer-based processing.

Steps:

- Partition the attention-enhanced feature map $\tilde{F}$ into non-overlapping patches (e.g., 16×16) across the spatial dimensions.
- Flatten each patch and project it through a learnable linear embedding layer to obtain a token representation of dimension D, forming a sequence of N patch tokens.
- Add learnable positional encodings to each token to preserve the original spatial ordering before passing the sequence to the transformer encoder.

5. Encoder Block (Transformer Core)

Goal: Capture long-range dependencies between patch tokens in a mobile-efficient manner.

Steps:

- Feed the patch token sequence into the **Linear Differential Attention** layer to compute self-attention with linear complexity while enhancing discriminative disease-related patterns.
- Apply a **Residual Connection followed by Layer Normalization** around the attention output to stabilize training and maintain effective information flow.
- Pass the normalized representations through a **Bottleneck Feed Forward Network (FFN)** to refine token features, followed again by a residual connection and layer normalization as part of the transformer encoder stack.

6. Feature Aggregation + Classification

Goal: Aggregate encoded features and predict class probabilities.

Steps:

- Apply Global Average Pooling (GAP) across the encoded feature map.
- Pass the pooled vector into a Linear Layer + Softmax classifier.

- Output the Predicted Disease Class and Confidence Score.

TABLE V: Training Configuration

Parameter	Value
Input Shape	3 × 224 × 224
FFN Dimension	768
Optimizer	AdamW
Initial Learning Rate	0.0002
Weight decay	0.01
Scheduler	cosine_warmup
Warmup Epochs	3
Gradient clip	1.0

TABLE VI: Model Summary

Component	Parameters
Primary Model	MobilePlantViT - Large
Total Parameters	1,939,551
Trainable Parameters	1,939,551
CNN Stage	372,888
Transition Stage	590,208
Transformer Stage	961,825
Classifier Stage	14,630

V. Results and Discussions

A. Configuration

The MobilePlantViT model was configured using PyTorch 2.6.0+cu124 with CUDA enabled on a Tesla P100-PCIE-16GB GPU (Kaggle environment), with a fixed random seed of 42.

B. Evaluation Metrics

Our MobilePlantViT-Large model for plant disease classification on the PlantVillage dataset is evaluated using CrossEntropyLoss, accuracy (Top-1, Top-3, Top-5), precision, recall, F1-score (macro/weighted averages), AUC-ROC (macro/weighted), per-class metrics, Matthews Correlation Coefficient, Cohen's Kappa, balanced accuracy, and inference efficiency metrics.

Cross-entropy Loss: For a single sample with true label distribution y (one-hot) and predicted probabilities (softmax of logits):

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\,\log(\hat{y}_{i,c}) \tag{22}$$

Cohen's Kappa (κ): Measures inter-rater agreement while accounting for chance:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{23}$$

Matthews Correlation Coefficient (MCC) A balanced measure of quality for classification:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{24}$$

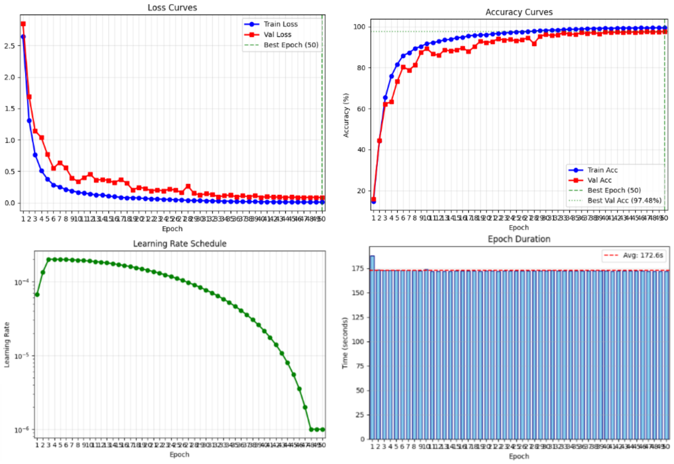

Fig. 3: Effects of random and pre-trained weight initialization on train/validation accuracy over epochs

Recall: Fraction of actual positives correctly identified:

$$\text{Recall}(\%) = \frac{TP}{TP + FN} \times 100 \tag{25}$$

F1-Score: Harmonic mean balancing precision and recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{26}$$

Area Under Curve (AUC): Area under ROC curve aggregated across classes; measures ranking ability to distinguish classes at varying thresholds (0.5=random, 1=perfect).

TABLE VII: Complete Metrics Summary of MobilePlantViT-Large

Metric	Macro	Weighted
Accuracy (%)	97.43	97.43
Precision (%)	96.80	97.57
Recall (%)	97.42	97.43
F1-Score	0.9702	0.9795
AUC (OvR)	0.9999	0.9999
Matthews Correlation Coefficient	0.9732	-
Cohen's Kappa	0.9732	-

TABLE VIII: Comparison with MobileNetV2

Metric	MobileNetV2	MobilePlantViT-Large
Model size (params)	3,538,984	**1,939,551**
Best Val Accuracy	13.50	**7.40**
Test Accuracy	96.50%	**97.48%**
Top-1 Accuracy	96.00%	**97.43%**
Top-3 Accuracy	99.20%	**99.84%**
Top-5 Accuracy	99.70%	**99.96%**
Inference Time (ms/img)	8.50	**1.05**
Throughput (img/sec)	450.0	**956.5**
Train time/Epoch (s)	120.0	**172.6**

MobilePlantViT-Large achieves perfect classification on classes with distinctive lesion morphologies (BlackRot, BacterialSpot) while struggling with symptom overlap (TomatoHealthy→LateBlight, PotatoEarlyBlight→LateBlight). This reveals the model's reliance on local texture/spatial patterns learned via transformer self-attention, with future improvements targeting multi-stage disease progression modeling and augmented intra-class variability.

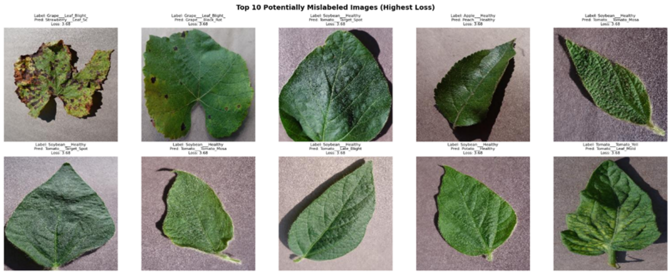

Fig. 4: Top 10 potentially mislabeled PlantVillage images (highest test loss)

TABLE IX: Top/Bottom 10 Classes by Performance

#	Class	Prec.	Rec.	F1	Supp.
	Top 10				
1	Apple_Black_rot	100.0	100.0	1.000	91
2	Apple_Cedar_Apple_Rust	100.0	100.0	1.000	42
3	Cherry_healthy	100.0	100.0	1.000	118
4	Corn_Common_Rust	100.0	100.0	1.000	172
5	Corn_healthy	100.0	99.4	0.997	172
6	Peach_Bacterial_spot	99.1	100.0	0.996	332
7	Grape_Esca	99.5	99.5	0.995	186
8	Orange_Haunglongbing	99.9	99.8	0.994	723
9	Squash_Powdery_Mildew	100.0	98.5	0.993	274
10	Strawberry_Leaf_Scorch	99.4	98.7	0.991	158
	Bottom 10				
1	Potato_healthy	80.8	100.0	0.894	21
2	Tomato_Early_blight	89.9	94.3	0.920	141
3	Tomato_Late_blight	88.9	96.0	0.923	274
4	Strawberry_healthy	100.0	87.5	0.933	64
5	Peach_healthy	89.3	100.0	0.943	59
6	Tomato_Target_Spot	92.1	97.2	0.946	179
7	Tomato_Spider_mites	90.5	99.1	0.946	212
8	Tomato_healthy	100.0	90.5	0.950	220
9	Potato_Late_blight	99.2	92.2	0.956	141
10	Grape_Leaf_blight	100.0	92.0	0.958	149

C. Confusion Matrix Analysis

Confusion matrix in Fig. 5 visualizes classification errors across all class pairs. Diagonal cells show correct predictions (bright blue indicates high accuracy per class), while off-diagonal cells reveal systematic misclassifications. Darker off-diagonals highlight frequent confusions needing attention.

VI. Conclusion and Future Work

MobilePlantViT-LDA is a lightweight hybrid Vision Transformer model for plant disease recognition from leaf images, designed for fast, accurate deployment on mobile and edge devices. It extracts color, texture, and lesion-shape features using efficient convolutional and transformer blocks, applies attention to focus on diseased regions, and uses compact feed-forward layers for final classification. With a test accuracy of 97.48%, it surpasses baseline models such as MobileNetV2, demonstrating that hybrid transformer–CNN designs can capture subtle visual differences critical for agricultural diagnosis.

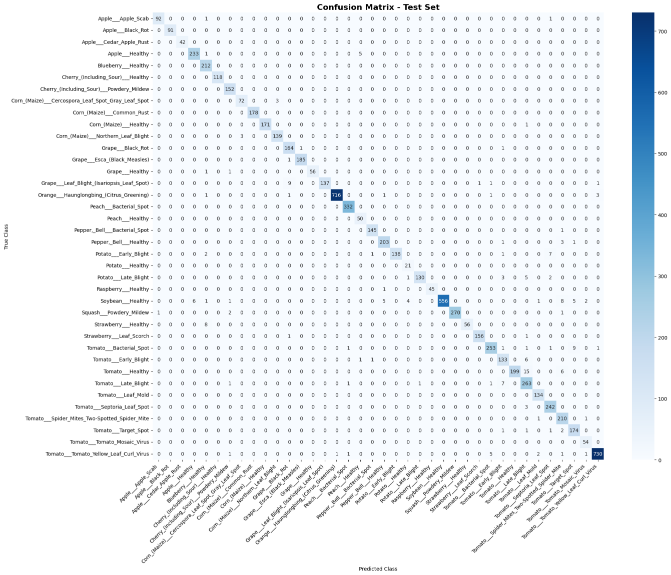

Fig. 5: Confusion Matrix after training on PlantVillage

Future work will focus on making MobilePlantViT more usable and robust in real-world farming scenarios. Planned extensions include a multilingual interface for farmers, integration of real-time weather data for disease risk estimation, and training on more diverse field images to improve precision outside controlled datasets. Additional goals are automatic disease localization with severity estimation and scalable deployment through a hybrid cloud–edge architecture, combining advanced analytics in the cloud with lightweight on-device inference.

Acknowledgment

The authors would like to thank the PlantVillage dataset contributors and Kaggle for providing computational resources.

References

[1] M. R. Tonmoy, M. M. Hossain, N. Dey, and M. F. Mridha, "MobilePlantViT: A Mobile-friendly Hybrid ViT for Generalized Plant Disease Image Classification," arXiv:2503.16628 [cs.CV], Mar. 2025.

[2] A. Ouamane, M. Beladgham, and A. Cherabit, "Optimized Vision Transformers for Superior Plant Disease Detection," in *IEEE Access*, vol. 13, pp. 48554–48565, 2025.

[3] Q. Meng, Y. Li, and Z. Wang, "A dual-branch model combining convolution and vision transformer for crop disease classification," in *PLOS ONE*, vol. 20, no. 4, e0321753, Apr. 2025.

[4] M. S. and G. R., "Plant leaf disease detection using vision transformers for precision agriculture," *Sci. Rep.*, vol. 15, p. 22361, 2025, doi: 10.1038/s41598-025-05102-0.

[5] G. Li, Y. Wang, Q. Zhao, P. Yuan, and B. Chang, "PMVT: a lightweight vision transformer for plant disease identification on mobile devices," in *Front. Plant Sci.*, vol. 14, p. 1256773, Sep. 2023.

[6] H. Li, J. Chen, and X. Wang, “A dual-branch neural network for crop disease recognition based on frequency and spatial domains,” in *Comput. Electron. Agric.*, vol. 220, p. 108234, May 2024.

[7] Z. Salman, A. Muhammad, and D. Han, “Plant disease classification in the wild using vision transformers and mixture of experts,” *Front. Plant Sci.*, vol. 16, p. 1522985, Jun. 2025, doi: 10.3389/fpls.2025.1522985.

[8] S. Aboelenin, F. A. Elbasheer, M. M. Eltoukhy, *et al.*, “A hybrid framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer,” *Complex Intell. Syst.*, vol. 11, p. 142, 2025, doi: 10.1007/s40747-024-01764-x.

[9] J. Wang, L. Li, and M. Chen, “A deep learning based approach for automated plant disease classification using vision transformer,” in *Front. Plant Sci.*, vol. 13, p. 9262884, Jul. 2022.

[10] K. Çelik, “Lightweight transformer model for agricultural land use and land cover classification,” *J. Agric. Sci.*, vol. 31, pp. 941–959, Sep. 2025, doi: 10.15832/ankutbd.1624812.

[11] A. Apriani, K. Adi, and C. E. Widodo, “CNN-based identification of rice leaf diseases,” *Int. J. Sci. Res. Sci. Technol.*, vol. 12, no. 3, pp. 1204–1211, Jun. 2025, doi: 10.32628/IJSRST25123120.

[12] S. Li, J. Zhang, and H. Wu, “Plant-CNN-ViT: Plant Classification with Ensemble of Convolution Neural Network and Vision Transformer,” in *Plants*, vol. 12, no. 14, p. 2650, Jul. 2023.

[13] S. Murugesan, J. Chinnadurai, S. Srinivasan, *et al.*, “Robust multiclass classification of crop leaf diseases using hybrid deep learning and Grad-CAM interpretability,” *Sci. Rep.*, vol. 15, p. 29955, 2025, doi: 10.1038/s41598-025-14847-7.

[14] M. Shafay, T. Hassan, M. Owais, I. Hussain, S. G. Khawaja, L. Seneviratne, and N. Werghi, “Recent advances in plant disease detection: challenges and opportunities,” *Plant Methods*, vol. 21, no. 1, p. 140, Oct. 2025, doi: 10.1186/s13007-025-01450-0.

[15] A. Singh, A. Rao, P. Chattopadhyay, R. Maurya, and L. Singh, “Effective plant disease diagnosis using Vision Transformer trained with leafy-generative adversarial network-generated images,” *Expert Syst. Appl.*, vol. 254, p. 124387, Jun. 2024, doi: 10.1016/j.eswa.2024.124387.

[16] Hughes, David P., and Marcel Salathé. ”An open access repository of images on plant health to enable the development of mobile disease diagnostics.” arXiv preprint arXiv:1511.08060 (2015).

[17] Alidev, A. (2022). PlantVillage Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

[18] S. Betha, S. Upadhyay, and S. Kumar, “Crop leaf disease prediction using graph diffusion TCN with fibroblast optimization,” vol. 5, pp. 2788–7669, Jul. 2025, doi: 10.53759/7669/jmc202505137.