MobilePlantViT: A Mobile-friendly Hybrid ViT for Generalized Plant Disease Image Classification

Moshiur Rahman Tonmoy, Md. Mithun Hossain, Nilanjan Dey, *Senior Member, IEEE*, M. F. Mridha, *Senior Member, IEEE*

***Abstract*—Plant diseases significantly threaten global food security by reducing crop yields and undermining agricultural sustainability. AI-driven automated classification has emerged as a promising solution, with deep learning models demonstrating impressive performance in plant disease identification. However, deploying these models on mobile and edge devices remains challenging due to high computational demands and resource constraints, highlighting the need for lightweight, accurate solutions for accessible smart agriculture systems. To address this, we propose MobilePlantViT, a novel hybrid Vision Transformer (ViT) architecture designed for generalized plant disease classification, which optimizes resource efficiency while maintaining high performance. Extensive experiments across diverse plant disease datasets of varying scales show our model's effectiveness and strong generalizability, achieving test accuracies ranging from 80% to over 99%. Notably, with only 0.69 million parameters, our architecture outperforms the smallest versions of MobileViTv1 and MobileViTv2, despite their higher parameter counts. These results underscore the potential of our approach for real-world, AI-powered automated plant disease classification in sustainable and resource-efficient smart agriculture systems. All codes will be available in the GitHub repository: https://github.com/moshiurtonmoy/MobilePlantViT**

***Index Terms*—plant disease, precision agriculture, vision transformer, attention mechanism, computer vision.**

I. Introduction

PLANT diseases represent a significant threat to global food security, impacting crop output and endangering the livelihoods of millions of people. According to the Food and Agriculture Organization (FAO), up to 40% of global crop production is lost annually due to pests and diseases, amounting to billions of dollars in economic losses and exacerbating food scarcity challenges [1]. Automated and accurate identification of these diseases is essential to mitigate crop damage and ensure sustainable agricultural productivity, and deep learning (DL) has revolutionized the field of automated plant disease recognition [2], [3] over the past decade. Convolutional neural network (CNN)–based models have demonstrated notable success in extracting discriminative features from complex leaf images, achieving competitive performance in classification tasks [4], [5]. In recent years, Vision Transformers (ViTs) have emerged as a compelling alternative to CNNs [6] and have shown remarkable performance gains across a wide range of computer vision tasks, including classification, suggesting their potential applicability in plant disease diagnosis [7].

However, deploying high-performance DL models to resource-constrained devices remains a major challenge. For instance, the high computational and memory demands of ViTs can be prohibitive in low-power scenarios [8] despite their state-of-the-art performance. Likewise, conventional CNN backbones also face trade-offs between model capacity and hardware limitations, spurring researchers into lightweight models. There has been significant progress toward developing compact models including ViTs [6], and models such as MobileViTv1 [9] and MobileViTv2 [10] are the prime examples. They incorporate efficient mechanisms for compressed feature representations and streamlined designs to narrow the performance gap between standard ViT and CNN counterparts [11], [12]. Nonetheless, these methods can remain computationally intensive in ultra-low-power devices, underscoring the need for optimizations that balance accuracy and maximum efficiency in ensuring precision agriculture accessibility globally.

In the context of scaling AI-based solutions for global precision agriculture advancement across underrepresented farming communities, tailored lightweight architectures become even more critical for cost-efficient automated agricultural tasks [13]. As high-end smart agriculture tools and automated systems often put significant financial barriers to users from low-income regions, introducing on-device smart systems integrated with tailored lightweight AI models demanding minimum computational overhead could be a feasible choice. With this motivation, we investigate lightweight yet generalized DL models in the plant disease classification domain and propose MobilePlantViT, a hybrid ViT architecture tailored for accurate plant disease image classification maintaining well balance between performance and efficiency. The key contributions of this study can be summarized as follows:

- We propose a generalized lightweight ViT for accurate plant disease image classification, with only 0.69 million parameters
- Our model achieved competitive accuracy across extensive experiments on multiple plant disease datasets
- Despite having fewer parameters, our model outperformed equivalent lightweight ViTs with higher parameter counts

The rest of this article is organized as follows: Section II summarizes past works, Section III presents methodology,

Moshiur Rahman Tonmoy is with the Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh (email: moshiurtonmoy.bb@gmail.com)

Md. Mithun Hossain is with the Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh (email: mhosen751@gmail.com)

Nilanjan Dey is with the Department of Computer Science and Engineering, Techno International New Town, Kolkata 700156, India (email: nilanjan.dey@tint.edu.in)

M. F. Mridha is with the Department of Computer Science, American International University-Bangladesh, Dhaka 1229, Bangladesh (email: firoz.mridha@aiub.edu)

Section IV provides the experimental results and discusses the key findings. Finally, Section V concludes this study with future directions.

II. Related Works

Over the years, researchers have focused on various aspects of automated plant disease recognition systems, particularly with the effectiveness of CNN-based architectures. Rehman et al. [14] employed CNNs to identify diseases in vegetables while overcoming obstacles like limited datasets, though limiting its scalability for underrepresented crops. Similarly, Demilie [15] compared multiple CNNs and reported up to 99.60% accuracy via ensemble approaches, however, ensemble methods increase computational overhead, potentially constraining edge deployment. CNNs were further shown to be effective by Gupta et al. [16] and Bezabih et al. [17], who combined VGG16 and AlexNet to classify pepper diseases with nearly perfect accuracy, but these relatively older architectures can be computationally heavier compared to modern variants. Additionally, Hassan and Maji [18], and Mzoughi and Yahiaoui [19] presented sophisticated CNN models that included segmentation methods to enhance accuracy while lowering complexity, yet such segmentation-based approaches often add extra preprocessing steps. To further improve classification performance, Wang et al. [20] and Nagaraju and Chawla [21] proposed T-CNN and DCNN-19, respectively, although these architectures still risk high inference times on low-power devices. According to Ojo and Zahid [22], class imbalance and data inconsistencies can be effectively reduced by combining ResNet-50 with preprocessing methods like CLAHE and GAN-based resampling, but these additional steps may complicate the pipeline and increase total computational overhead. Furthermore, Chen et al. [23] illustrated the advantages of deep transfer learning using a modified MobileNet-V2 to improve the detection of subtle lesion symptoms. The DIC-Transformer presented by Zeng et al. [24] achieves 85.4% accuracy while delivering comprehensive disease descriptions by combining Faster R-CNN for disease identification with a Transformer for picture caption creation. Nonetheless, this heavy two-stage pipeline can be computationally intensive for cost-effective applications. To improve feature extraction and prediction accuracy over models such as FCN-8s and DeepLabv3, Kalpana et al. [25] suggested an ensemble of Swin transformers and residual convolutional networks, although ensemble approaches typically demand greater memory usage and training time. Zhu et al. [26] introduced MSCVT, a hybrid model combining CNNs and ViT that achieved 99.86% accuracy on PlantVillage with reduced parameter complexity. Teki et al. [27] applied ViT and Shifted Window Transformer to thermal images of paddy leaves, achieving accuracies of 94% and 98%, respectively, but the requirement for thermal imaging devices and specialized hardware could restrict broad adoption. Borhani et al. [28] investigated lightweight ViT-based techniques that, despite being slower than conventional CNNs, achieved higher accuracy, underscoring a persistent trade-off between performance and resource usage in Transformer-based approaches. MFF-ADNet, which combines ViT, VGG16, and a variational autoencoder, was introduced by Bathula et al. [29] and outperforms current techniques by 7.18%, though this integration of multiple components can inflate memory requirements and training complexity. AppViT, a lightweight hybrid model that merges convolutional blocks with multi-head self-attention, was proposed by Wasi Ullah et al. [30] and achieved 96.38% precision on the Plant Pathology 2021-FGVC8 dataset. However, multi-head attention can be computationally heavy on extremely resource-limited devices. Chen et al. [31] merged transfer learning with a vision transformer to attain validation accuracies up to 99.86%, but it remains unclear how robust these models are under field conditions with non-uniform lighting or overlapping leaves. Li et al. [32] introduced PDLM-TK, which combines tensor features with knowledge distillation and lightweight residual blocks and obtained 96.19% accuracy and a 94.94% f1 score. Thakur et al. [33] combined ViTs with CNNs and produced 98.86% accuracy on the PlantVillage dataset with just 0.85M parameters. Finally, Muthireddy et al. [34] and Chougui et al. [35] demonstrated that adding handcrafted features and attention mechanisms improves classifier performance. Nonetheless creating such handcrafted features can be labor-intensive and may not generalize well to complex or newly emerging plant diseases.

Overall, past studies have highlighted the efficacy of CNN and ViT-based methods for various aspects of plant disease classification. However, these studies focused predominantly on maximizing accuracy and were often limited to specific plant species. Deploying separate DL models for different crops would be impractical and costly, underscoring the need for a tailored, lightweight architecture with strong generalization capabilities across multiple crops.

III. Methodology

Our proposed architecture comprises multiple components as illustrated in Figure 1. It begins with a DepthConv block as the stem layer which increases the channel dimension from 3 to 32 along with the very first feature extraction operation. A DepthConv block essentially employs Depthwise Separable Convolution [36], an efficient variant of the standard convolution which works in two steps: first, each input channel is convolved independently using a single spatial filter (Depthwise Convolution). Then, a 1×1 convolution (Pointwise Convolution) is applied to combine the outputs from the depthwise step and mix information across channels. Next, the GroupConv block comprises Group Convolution which can be considered as a generalized form of Depthwise Separable Convolution where each input channels are divided into user-defined groups, and each group is convolved separately using a dedicated set of filters, and the Depthwise Convolution is a special case where the number of groups equals the number of input channels. We set half the input channels as groups in each GroupConv block. Additionally, we imposed a residual connection within each block for effective information flow. These multi-step and residual operations significantly reduce the computational burden of standard convolution while enabling efficient feature extraction in deep networks.

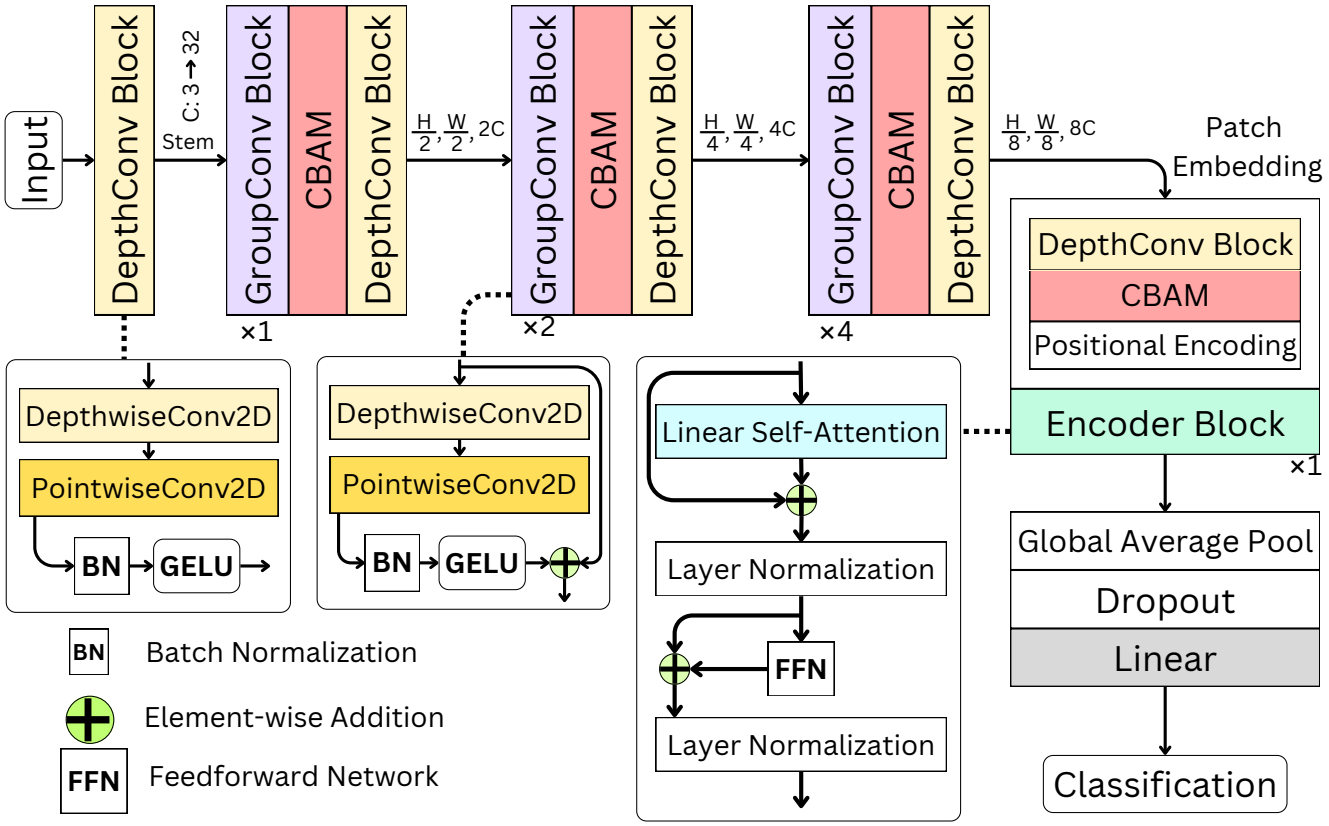


Fig. 1. Outline of the proposed MobilePlantViT. It consists of a ×1-×2-×4-×1 combination of GroupConv and Encoder blocks. The first DepthConv block acts as the stem layer, expanding the initial channel dimension from 3 to 32. The last DepthConv block serves as the patch embedding layer, while all intermediate DepthConv blocks function as spatial pooling layers with dimension reduction and channel expansion

The CBAM stands for Convolutional Block Attention Module [37] which enhances feature representation by sequentially applying channel attention and spatial attention, allowing the network to focus on important information while suppressing less significant features. The channel attention module emphasizes identifying the important features by learning a channel-wise attention map by utilizing global average pooling (GAP) & global max pooling (GMP) of the feature maps followed by a multilayer perceptron (MLP) with sigmoid activation to generate attention weights. Mathematically, let F be the feature maps, then the channel attention $M_c(F)$ is:

$$M_c(F) = \sigma(W_1(\delta(W_0(GAP(F)))+W_1(\delta(W_0(GMP(F))))))) \quad (1)$$

$$F' = M_c(F) \odot F \quad (2)$$

where σ is sigmoid function $(\frac{1}{1+e^{-x}})$, δ means ReLU activation which is $max(0, x)$, W_1 and W_0 are learnable MLP weights, $\odot$ is element-wise multiplication, and F' is the refined feature maps with channel attention. Similarly, the spatial attention module focuses on important regions in the feature map using a 7×7 convolution after applying max and average pooling across channels to generate a spatial attention map. Mathematically, let F' be the refined feature maps after the channel attention module, then the spatial attention $M_s(F')$ is:

$$M_s(F') = \sigma(f^{7\times7}([AvgPool(F'); MaxPool(F')])) \quad (3)$$

$$F'' = M_s(F') \odot F' \quad (4)$$

where $f^{7\times7}$ is a 7×7 convolution, and F'' is the refined feature maps output after CBAM. Then, we employed patch embedding with positional encoding over the extracted feature maps $(\frac{H}{8}, \frac{W}{8}, 8C)$ for the encoder block. The patch embedding splits the input feature maps into fixed-size patches and transforms each patch into a high-dimensional vector, forming a sequence of image patches. We used a DepthConv block with stride and kernel size equal to patch size to split the features into patches. In addition, we integrated a CBAM block into the patch extraction process. Positional encoding is also added to the patch embeddings to retain spatial information, as encoders do not naturally capture spatial relationships. Then, instead of the classic Transformer encoder, we integrated a lightweight encoder block comprising a linear self-attention block instead of the multi-head self-attention. We adapted the linear self-attention mechanism from the MobileViTv2 [10] as it enables efficient feature attention with linear complexity, suitable for resource-constraint devices. Given an input representation X, the encoder block computes the attention output:

$$X' = \text{LN}(X + \text{LinearAttention}(X)) \quad (5)$$

where LN represents layer normalization. In LinearAttention block, given an input sequence $X \in \mathbb{R}^{L\times d}$, where L is the sequence length, and d is the embedding dimension, we compute the query (Q), key (K), and value (V) representations as:

$$Q, K, V = W_{qkv}X + b_{qkv} \tag{6}$$

where $W_{qkv} \in \mathbb{R}^{(1+2d)\times d}$ is a learnable weight matrix, and b_{qkv} is a bias term. The $Q \in \mathbb{R}^{L\times 1}$, and $K, Q \in \mathbb{R}^{L\times d}$ are obtained via linear projection. The attention or context scores are computed using a softmax function applied along the sequence length, and then go through a dropout regularization layer:

$$\alpha = \text{Softmax(Q)} \tag{7}$$

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{8}$$

where x_i is the i-th input value and n is the total number of input values. Thus, $\alpha \in \mathbb{R}^{L\times 1}$ represents the normalized attention scores. Next, we compute the context vector as:

$$C = \sum_{i=1}^{L} \alpha_i \cdot K_i \tag{9}$$

where $C \in \mathbb{R}^{1\times d}$ is the aggregated K representation weighted by the attention scores. The output of the attention mechanism is then obtained by applying the GELU activation on the V, followed by element-wise multiplication with the expanded context vector. Next, a linear transformation is applied:

$$\hat{O} = W_{\text{out}}(\text{GELU}(V) \odot C) + b_{\text{out}} \tag{10}$$

Here, GELU activation is $x\Phi(x)$ where the $\Phi(x)$ is the standard gaussian cumulative distribution function [38]. $W_{\text{out}} \in \mathbb{R}^{d\times d}$ and b_{out} are trainable parameters. The X' is then processed by a feedforward network (FFN) of two linear layers with GELU and dropout regularization.

$$\text{FFN}(X') = (W_2 \cdot \text{GELU}(W_1 X' + b_1) + b_2) \tag{11}$$

$$X'' = \text{LN}(X' + \text{FFN}(X')) \tag{12}$$

where $W_1 \in \mathbb{R}^{d\times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff}\times d}$ are the feedforward layer weights, and d_{ff} is the hidden dimension of the FFN. Residual connections are applied at both the attention and feedforward layers. The encoder gives us the output representation $X'' \in \mathbb{R}^{L\times d}$. To obtain a global feature representation, we apply GAP along the L (sequence length):

$$Z = \frac{1}{L}\sum_{i=1}^{L} X_i'' \tag{13}$$

where $Z \in \mathbb{R}^{B\times d}$ represents the pooled feature vector and B is the batch size. Dropout is applied to prevent overfitting. Lastly, a linear layer is applied to obtain the predicted logits for classification:

$$\hat{y} = W_{\text{c}} Z' + b_{\text{c}} \tag{14}$$

where $W_{\text{c}} \in \mathbb{R}^{d\times C}$ and $b_{\text{c}} \in \mathbb{R}^{C}$ are the learnable parameters of the classifier and the output $\hat{y} \in \mathbb{R}^{B\times C}$ represents the predicted logits which eventually go through a softmax to obtain class probabilities and we pick the class with the maximum value:

$$p_i = \text{Softmax}(\hat{y}) \tag{15}$$

$$\hat{c} = \arg\max_i p_i \tag{16}$$

where p_i represents the probability of class i ensuring $\sum_{i=1}^{C} p_i = 1$, and $\hat{c}$ is the predicted class.

IV. Results and Discussion

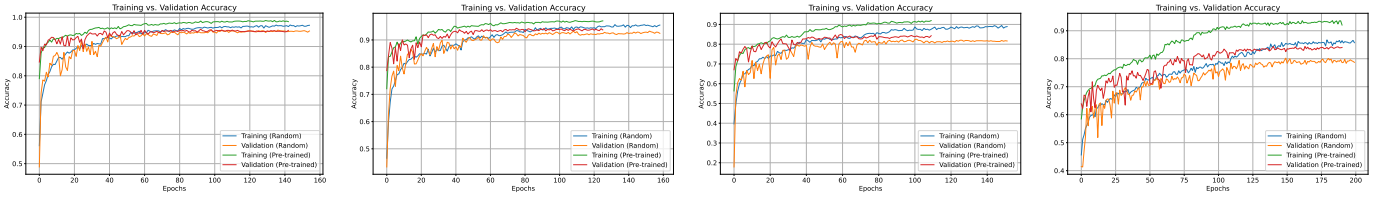
In this section, experimental setup and datasets are presented. Subsequently, the experimental results on the performance of the model are discussed highlighting the key findings from various aspects.

A. Experimental Data and Setup

We have employed 4 datasets ranging from large-scale to small-scale: PlantVillage [39], CCMT [40], Sugarcane [41], and Coconut [42]. The PlantVillage dataset consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and diseases. The CCMT dataset contains 6549, 7508, 5389, and 5435 disease and healthy raw images of Cashew, Cassava, Maize, and Tomato. Each of them has 5 classes except for Maize which has 7 classes. The Sugarcane dataset includes 6748 high-resolution leaf images classified into 9 disease categories and the Coconut dataset comprises 5798 images across 5 disease categories. Datasets were split into training, validation, and test sets in a 70-15-15 ratio. We applied learning rate reduction and early stopping based on validation accuracy. If accuracy did not improve for 10 consecutive epochs, the learning rate was reduced by half, and if the non-improvement trend continued for 50 epochs, training was halted and the best weights were retrieved. We implemented our model using PyTorch and all experiments were conducted in Google Colab's notebook environment with an NVIDIA Tesla T4 GPU. Table I summarizes the experimental setup, including augmentation techniques applied to the training images. All data were normalized using a mean of (0.485, 0.456, 0.406) and a standard deviation of (0.229, 0.224, 0.225).

B. Performance Analysis

Table II portrays the overall performance summary of our proposed model. The model demonstrated high classification accuracy across different datasets, with particularly strong performance on the PlantVillage and Coconut datasets, achieving accuracy rates of 99.57% and 99.20%, respectively. Among the CCMT datasets, the model exhibited robust results for CCMT-Cashew (95.04%), CCMT-Cassava (94.34%), and 92.76% test accuracy for Sugarcane. These results suggest that the model generalizes well to different crop types, even when dealing with real-world datasets that may contain variations in lighting, background, and disease severity. The f1-scores



(a) CCMT-Cashew (b) CCMT-Cassava (c) CCMT-Maize (d) CCMT-Tomato

Fig. 2. Effects of random and pre-trained weight initialization on the train vs. validation accuracy over epochs

TABLE I
EXPERIMENTAL SETUP AND DATA AUGMENTATION SUMMARY

Attribute	Value
Input Shape	$3 \times 224 \times 224$
Batch Size	64
Embed Dimension	256
FFN Dimension	512
Patch Size	00
Initial Learning Rate	$1e-3$
Learning Rate Reduction Rate	50%
Minimum Learning Rate	$1e-7$
Learning Rate Reduction Patience	10
Validation Accuracy Threshold	$1e-5$
Early Stopping Patience	50
Weight Decay	$1e-3$
Encoder Dropout Rate	30%
Classifier Dropout Rate	20%
Optimizer	Adam
Loss	Categorical Crossentropy
Horizontal Flip	50%
Random Rotate 90°	50%
Shift, Scale, Rotate (limits: 0.05, 0.05, 30°)	50%
Random Gamma	20%
Random Brightness/Contrast	30%
RGB Shift (shift limits: 15)	30%
CLAHE (clip limit: 4.0)	30%

for these datasets remain consistently high, with both macro and weighted averages exceeding 0.94 for CCMT-Cashew and CCMT-Cassava, further reinforcing the model's balanced performance across classes. The performance on the CCMT-Maize and CCMT-Tomato datasets was comparatively lower, with accuracy rates of 81.05% and 80.05%, respectively. However, the macro and weighted precision, recall, and f1-scores for these datasets still indicate reasonably effective classification performance, suggesting that the model can be further improved with additional training data or enhanced preprocessing techniques. To grab the whole picture, the One-vs-rest (OvR) AUC metric provides crucial insight into the discriminatory power of our model. We observe that it produced above 0.99 score for all the data except Maize and Tomato. Overall, these findings indicate that the proposed MobilePlantViT achieves state-of-the-art performance on standardized datasets while maintaining strong generalizability across diverse agricultural datasets.

To evaluate the impact of domain-specific pre-trained weight initialization on small-scale datasets, we initialized our model with weights pre-trained on the PlantVillage dataset and retrained it on the CCMT dataset. Our findings (presented in Table III) indicate that pre-training on relevant large-scale datasets (in our case PlantVillage) significantly enhances performance across small-scale datasets. The highest accuracy improvement was observed for CCMT-Maize and CCMT-Tomato, with increases of 4.00% and 4.50%, respectively. Precision, recall, and f1-score also followed a similar trend. These increases in the performance scores endorse that domain-specific pre-training provides valuable feature representations that generalize well to unseen datasets. Therefore, incorporating this strategy can positively impact the model performance, particularly for lightweight architecture and datasets with limited training samples. Figure 2 illustrates the training and validation accuracy and loss trends over epochs, highlighting the impact of random vs. pre-trained weight initialization. Models initialized with pre-trained weights show an immediate boost in both training and validation accuracy, leading to faster convergence in contrast to randomly initialized models. However, pre-trained weights can sometimes lead to overfitting (Figure 2d) and sufficient measures should be taken like regularization, early stopping, etc. Overall, leveraging pre-trained weights from relevant large-scale data enhances model performance and accelerates convergence which might come in handy for underperforming small datasets.

C. Ablation Study

We conducted an ablation study on the proposed model to evaluate the contribution of its key components to classification performance. The experiments were performed on the CCMT-Tomato dataset, as it proved to be the most challenging dataset in our study. Table IV summarizes the results of the ablation study. The findings reveal a substantial performance decline when either CBAM is removed or the GroupConv block formation is altered. In Case 1, where both CBAM and the proposed 1-2-4 GroupConv configuration were excluded, the model experienced the most significant drop in performance. Accuracy decreased by 9.32%, macro precision by 6.07%, weighted precision by 9.41%, macro recall by 15.3%, and macro f1-score by 13.47% compared to the proposed model, as shown in Table II. This demonstrates that both CBAM and the 1-2-4 GroupConv formation are crucial for enhancing feature extraction and representation learning. In Case 2, where CBAM was removed but the 1-2-4 GroupConv formation was retained, the performance drop was less severe than in Case 1 but still significant. Accuracy declined by 4.82%, with macro precision, weighted precision, macro recall, and macro f1-score decreasing by 3.17%, 4.85%, 8.66%, and 9.87%, respectively. This suggests that while the 1-2-4 GroupConv structure plays a key role in feature extraction, CBAM

TABLE II
PERFORMANCE SUMMARY OF THE PROPOSED MODEL OVER MULTIPLE DATASETS, EVALUATED IN MACRO AND WEIGHTED AVERAGE

Data	Test Size	Accuracy (%)	Precision (%)		Recall (%)		F1-score		AUC (OvR)	
			Macro	Weighted	Macro	Weighted	Macro	Weighted	Macro	Weighted
PlantVillage	8179	99.57	99.47	99.58	99.40	99.57	0.9943	0.9957	0.9999	0.9999
Coconut	874	99.20	99.47	99.21	99.35	99.20	0.9941	0.9920	0.9999	0.9999
Sugarcane	1022	92.76	88.81	92.92	88.85	92.76	0.8870	0.9274	0.9976	0.9984
CCMT-Cashew	987	95.04	95.71	95.02	95.78	95.04	0.9574	0.9502	0.9939	0.9927
CCMT-Cassava	1131	94.34	94.46	94.40	94.41	94.36	0.9442	0.9436	0.9952	0.9944
CCMT-Maize	802	81.05	82.87	81.41	81.42	81.05	0.8185	0.8110	0.9731	0.9668
CCMT-Tomato	872	80.05	79.58	79.86	76.36	80.05	0.7754	0.7973	0.9517	0.9414

TABLE III
SUMMARY OF THE PROPOSED MODEL'S PERFORMANCE WITH PLANTVILLAGE PRE-TRAINED WEIGHTS

Data	Accuracy (%)	Precision (%)		Recall (%)		F1-score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
CCMT-Cashew	96.45 (1.48↑)	97.02 (1.37↑)	96.45 (1.51↑)	96.96 (1.23↑)	96.45 (1.48↑)	0.9698 (1.24%↑)	0.9645 (1.51%↑)
CCMT-Cassava	95.23 (0.94↑)	95.48 (1.09↑)	95.25 (0.91↑)	95.63 (1.30↑)	95.23 (0.94↑)	0.9555 (1.20%↑)	0.9524 (0.93%↑)
CCMT-Maize	84.29 (4.00↑)	85.67 (3.38↑)	84.31 (3.56↑)	85.28 (4.74↑)	84.29 (4.00↑)	0.8541 (3.56%↑)	0.8425 (3.88%↑)
CCMT-Tomato	83.60 (4.50↑)	82.90 (4.20↑)	83.40 (4.44↑)	80.98 (6.12↑)	83.60 (4.50↑)	0.8188 (4.34%↑)	0.8345 (4.67%↑)

TABLE IV
SUMMARY OF THE ABLATION STUDY HIGHLIGHTING THE KEY CONTRIBUTIONS OF THE CORE COMPONENTS OF THE MODEL

CBAM	GroupConv	Accuracy (%)	Precision (%)		Recall (%)		F1-score	
			Macro	Weighted	Macro	Weighted	Macro	Weighted
✗	1-1-1	72.59 (9.32↓)	74.75 (6.07↓)	72.35 (9.41↓)	64.68 (15.3↓)	72.59 (9.32↓)	0.6708 (13.47%↓)	0.7079 (11.22%↓)
✗	1-2-4	75.23 (4.82↓)	76.41 (3.17↓)	75.01 (4.85↓)	67.70 (8.66↓)	75.23 (4.82↓)	0.6989 (9.87%↓)	0.7385 (7.38%↓)
✓	1-1-1	77.64 (3.04↓)	77.69 (2.38↓)	77.56 (2.88↓)	73.53 (3.71↓)	77.64 (3.04↓)	0.7493 (3.37%↓)	0.7721 (3.16%↓)
✓	**1-2-4**	**80.05**	**79.58**	**79.86**	**76.36**	**80.05**	**0.7754**	**0.7973**

further refines feature selection and improves classification performance. Lastly, in Case 3, where CBAM was retained but the GroupConv formation was changed to 1-1-1, the model experienced a moderate decline in performance. Accuracy decreased by 3.04%, macro precision by 2.38%, weighted precision by 2.88%, macro-recall by 3.71%, and macro f1-score by 3.37%. These results highlight that while CBAM enhances feature representation, the hierarchical structure of the 1-2-4 GroupConv formation is a significant contributor to robust classification performance. Overall, the ablation study confirms that both the CBAM attention mechanism and the hierarchical 1-2-4 GroupConv block formation contribute synergistically to the model's performance. Removing either component results in performance degradation, with the worst-case scenario occurring when both CBAM and the 1-2-4 configuration are absent. Thus, our findings support the inclusion of both components in the proposed model to achieve optimal performance in plant disease classification.

D. Error Analysis

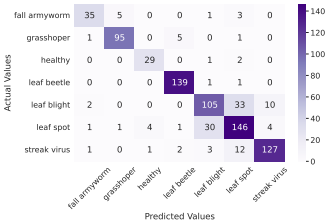
The most frequent misclassification for Maize occurred between *leaf blight, leaf spot*, and *streak virus*, and Tomatos involved *septorial leaf spot, verticillium wilt, leaf blight*, and *leaf curl*, observed via confusion matrix analysis presented in Figure 3a, 3b. As depicted in Figure 3c, 3d, one key reason for these misclassifications is the high similarity in the visual patterns of the affected areas. While lightweight models typically involve a trade-off between accuracy and resource efficiency, it is noteworthy that our model achieved competitive accuracy on the PlantVillage dataset (see Table II), which includes a subset of Tomato disease instances. The smaller training size and data quality likely played a significant role in these comparatively lower performances. To improve further, utilizing extensive augmentations to introduce more diverse large-scale training data, applying targeted augmentations on hard-to-classify samples, and domain-specific pre-training on large datasets could be beneficial. Additionally, contrastive learning and hierarchical classification approaches may also be useful in applicable contexts.

E. Performance Comparison With Equivalent ViTs

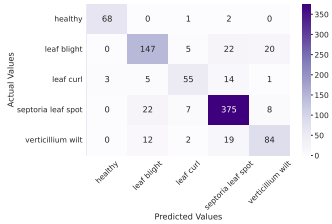
We conducted a comparative performance analysis against the lightweight versions of the MobileViT models with equivalent parameter constraints. We picked the MobileViTv1-XXS and the MobileViTv2-050 versions. The results are reported in Table V provides a summarized assessment. Our proposed model, MobilePlantViT, consistently outperformed MobileViT-XXS and MobileViTV2-050 across the experimental datasets. Despite having fewer parameters (0.69M) than both MobileViT-XXS (0.95M) and MobileViTV2-050 (1.12M), MobilePlantViT achieved the highest accuracy, precision, recall, and f1-scores. Notably, the performance improvement was more pronounced for the CCMT-Maize and CCMT-Tomato datasets, the challenging data segments of our experiments. Our model achieved a peak accuracy of 80.05%-83.60% on the CCMT-Tomato data, outperforming the MobileViTV2-050 producing an accuracy of 73.28% and MobileViT-XXS by 69.95%. For CCMT-Maize, our model

TABLE V
PERFORMANCE COMPARISON OF LIGHTWEIGHT VITS HAVING EQUIVALENT PARAMETERS

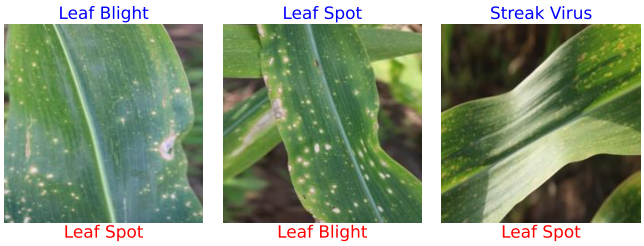
Model	Params (M)	Data	Accuracy (%)	Precision (%)		Recall (%)		F1-score	
				Macro	Weighted	Macro	Weighted	Macro	Weighted
MobileViTv1-XXS	0.95	CCMT-Cashew	94.02	94.82	94.01	94.88	94.02	0.9484	0.9401
		CCMT-Cassava	91.25	91.18	91.32	91.90	91.25	0.9152	0.9126
		CCMT-Maize	74.56	74.80	75.49	72.67	74.56	0.7324	0.7475
		CCMT-Tomato	69.95	70.76	69.58	65.06	69.95	0.6685	0.6921
MobileViTv2-050	1.12	CCMT-Cashew	94.43	95.22	94.43	94.76	94.43	0.9497	0.9441
		CCMT-Cassava	93.81	94.37	93.82	93.93	93.81	0.9414	0.9381
		CCMT-Maize	77.43	78.17	77.89	77.66	77.43	0.7783	0.7761
		CCMT-Tomato	73.28	73.83	72.75	68.35	73.28	0.7018	0.7246
MobilePlantViT	0.69	CCMT-Cashew	**95.04**	**95.71**	**95.02**	**95.78**	**95.04**	**0.9574**	**0.9502**
		CCMT-Cassava	**94.34**	**94.46**	**94.40**	**94.41**	**94.36**	**0.9442**	**0.9436**
		CCMT-Maize	**81.05**	**82.87**	**81.41**	**81.42**	**81.05**	**0.8185**	**0.8110**
		CCMT-Tomato	**80.05**	**79.58**	**79.86**	**76.36**	**80.05**	**0.7754**	**0.7973**



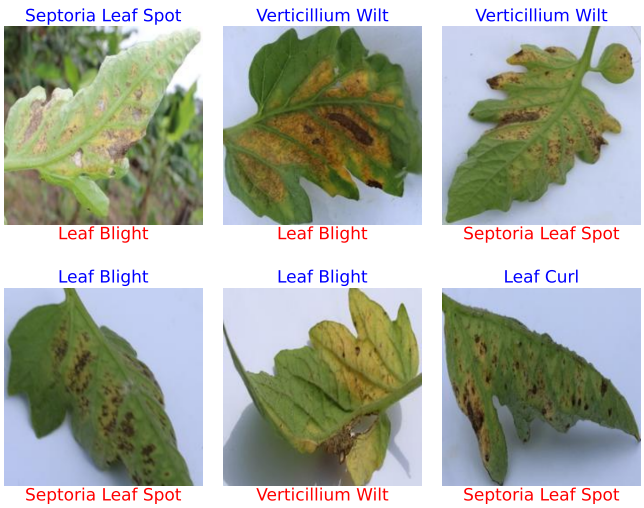
(a) CCMT-Maize Confusion Matrix



(b) CCMT-Tomato Confusion Matrix



(c) CCMT-Maize Misclassified Samples



(d) CCMT-Tomato Misclassified Samples

Fig. 3. Confusion matrices representing the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for Maize and Tomato, along with misclassified samples

achieved 81.05% accuracy with random weight initialization and 84.29% with pre-trained weight initialization, significantly outperforming MobileViTV2-050 (77.43%) and MobileViT-XXS (74.56%). Similar trends of producing lower performance scores were followed in the CCMT-Cashew and CCMT-Cassava as well. This significant performance superiority of our proposed model is reflected in all performance metrics, including macro and weighted precision, recall, and f1-score.

V. CONCLUSION AND FUTURE DIRECTION

In this study, we introduce MobilePlantViT, a novel DL model for effective and generalized plant disease image classification. Our model is computationally lightweight, making it well-suited for mobile and resource-constrained edge devices, with the broader goal of enhancing the accessibility and scalability of smart agricultural systems. To achieve efficiency without compromising accuracy, MobilePlantViT first extracts essential features via a stack of group convolutions that are fused with convolutional attention modules, progressively downsampling feature representations to reduce computational complexity before passing them to the encoder. Unlike conventional multi-head self-attention, which operates with quadratic complexity, our encoder leverages a self-attention mechanism with linear complexity, significantly improving scalability. Extensive experiments across multiple plant disease datasets, accompanied by an in-depth performance analysis from various perspectives demonstrate that MobilePlantViT achieves not only promising classification performance but also generalizes effectively across different instances of plant disease. In the future, further research could focus on adapting MobilePlantViT to other agricultural image classification tasks, expanding its applicability within precision agriculture. Additionally, exploring domain-specific pre-training could lead to improved performance and better generalizability, particularly for handling rare diseases or diverse environmental conditions, ultimately enhancing the model's robustness across various agricultural domains.

REFERENCES

[1] Food and A. O. of the United Nations, "Faostat," Rome, 2018, pp. 403-403. [Online]. Available: http://faostat.fao.org
[2] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv preprint arXiv:1409.1556.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[5] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, no. 1, p. 3289801, 2016.

[6] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[7] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9640–9649.

[8] X. Liu, Y. Song, X. Li, Y. Sun, H. Lan, Z. Liu, L. Jiang, and J. Li, "Ed-vit: Splitting vision transformer for distributed inference on edge devices," *arXiv preprint arXiv:2410.11650*, 2024.

[9] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=vh-0sUt8HlG

[10] ——, "Separable self-attention for mobile vision transformers," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=tBl4yBEjKi

[11] H. Cao, Z. Qu, G. Chen, X. Li, L. Thiele, and A. Knoll, "Ghostvit: Expediting vision transformers via cheap operations," *IEEE Transactions on Artificial Intelligence*, 2023.

[12] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5270–5279.

[13] K. Sharma and S. K. Shivandu, "Integrating artificial intelligence and internet of things (iot) for enhanced crop monitoring and management in precision agriculture," *Sensors International*, vol. 5, p. 100292, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666351124000147

[14] M. u. Rehman, J. Liu, A. Nijabat, M. Faheem, W. Wang, and S. Zhao, "Leveraging convolutional neural networks for disease detection in vegetables: A comprehensive review," *Agronomy*, vol. 14, no. 10, p. 2231, 2024.

[15] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *Journal of Big Data*, vol. 11, no. 1, p. 5, 2024.

[16] R. S. Singla, A. Gupta, R. Gupta, V. Tripathi, M. S. Naruka, and S. Awasthi, "Plant disease classification using machine learning," in *2023 International Conference on Disruptive Technologies (ICDT)*. IEEE, May 2023, pp. 409–413.

[17] Y. A. Bezabh, A. O. Salau, B. M. Abuhayi, A. A. Mussa, and A. M. Ayalew, "Cpd-ccnn: classification of pepper disease using a concatenation of convolutional neural network models," *Scientific Reports*, vol. 13, no. 1, p. 15581, 2023.

[18] S. M. Hassan and A. K. Maji, "Plant disease identification using a novel convolutional neural network," *IEEE Access*, vol. 10, pp. 5390–5401, 2022.

[19] O. Mzoughi and I. Yahiaoui, "Deep learning-based segmentation for disease identification," *Ecological Informatics*, vol. 75, p. 102000, 2023.

[20] D. Wang, J. Wang, W. Li, and P. Guan, "T-cnn: Trilinear convolutional neural networks model for visual detection of plant diseases," *Computers and Electronics in Agriculture*, vol. 190, p. 106468, 2021.

[21] M. Nagaraju and P. Chawla, "Plant disease classification using dcnn-19 convolutional neural networks," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. IEEE, September 2021, pp. 1–6.

[22] M. O. Ojo and A. Zahid, "Improving deep learning classifiers performance via preprocessing and class imbalance approaches in a plant disease detection pipeline," *Agronomy*, vol. 13, no. 3, p. 887, 2023.

[23] J. Chen, D. Zhang, and Y. A. Nanehkaran, "Identifying plant diseases using deep transfer learning and enhanced lightweight network," *Multimedia Tools and Applications*, vol. 79, pp. 31 497–31 515, 2020.

[24] Q. Zeng, J. Sun, and S. Wang, "Dic-transformer: interpretation of plant disease classification results using image caption generation technology," *Frontiers in Plant Science*, vol. 14, p. 1273029, 2024.

[25] P. Kalpana, R. Anandan, A. G. Hussien, H. Migdady, and L. Abualigah, "Plant disease recognition using residual convolutional enlightened swin transformer networks," *Scientific Reports*, vol. 14, no. 1, p. 8660, 2024.

[26] D. Zhu, J. Tan, C. Wu, K. Yung, and A. W. Ip, "Crop disease identification by fusing multiscale convolution and vision transformer," *Sensors*, vol. 23, no. 13, p. 6015, 2023.

[27] V. M. Teki, R. A. Ragaven, N. Manoj, V. Vipul, and S. Sarath, "A comparison of two transformers in the study of plant disease classification," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, July 2023, pp. 1–6.

[28] Y. Borhani, J. Khoramdel, and E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer," *Scientific Reports*, vol. 12, no. 1, p. 11554, 2022.

[29] B. Nagachandrika, R. Prasath, and I. P. Joe, "An automatic classification framework for identifying type of plant leaf diseases using multi-scale feature fusion-based adaptive deep network," *Biomedical Signal Processing and Control*, vol. 95, p. 106316, 2024.

[30] W. Ullah, K. Javed, M. A. Khan, F. Y. Alghayadh, M. W. Bhatt, I. S. Al Naimi, and I. Ofori, "Efficient identification and classification of apple leaf diseases using lightweight vision transformer (vit)," *Discover Sustainability*, vol. 5, no. 1, p. 116, 2024.

[31] A. Tabbakh and S. S. Barpanda, "A deep features extraction model based on the transfer learning model and vision transformer "tlmvit" for plant disease classification," *IEEE Access*, vol. 11, pp. 45 377–45 392, 2023.

[32] X. Zhang, K. Liang, and Y. Zhang, "Plant pest and disease lightweight identification model by fusing tensor features and knowledge distillation," *Frontiers in Plant Science*, vol. 15, p. 1443815, 2024.

[33] P. S. Thakur, S. Chaturvedi, P. Khanna, T. Sheorey, and A. Ojha, "Vision transformer meets convolutional neural network for plant disease classification," *Ecological Informatics*, vol. 77, p. 102245, 2023.

[34] V. Muthireddy and C. V. Jawahar, "Plant disease classification using hybrid features," in *International Conference on Computer Vision and Image Processing*. Springer Nature Switzerland, November 2022, pp. 477–492.

[35] A. Chougui, A. Moussaoui, and A. Moussaoui, "Plant-leaf diseases classification using cnn, cbam and vision transformer," in *2022 5th International Symposium on Informatics and its Applications (ISIA)*. IEEE, November 2022, pp. 1–6.

[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[38] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," *CoRR*, vol. abs/1606.08415, 2016. [Online]. Available: http://arxiv.org/abs/1606.08415

[39] G. G. and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790619300023

[40] P. K. Mensah, V. Akoto-Adjepong, K. Adu, M. A. Ayidzoe, E. A. Bediako, O. Nyarko-Boateng, S. Boateng, E. F. Donkor, F. U. Bawah, N. S. Awarayi, P. Nimbe, I. K. Nti, M. Abdulai, R. R. Adjei, M. Opoku, S. Abdulai, and F. Amu-Mensah, "Ccmt: Dataset for crop pest and disease detection," *Data in Brief*, vol. 49, p. 109306, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923004250

[41] S. Thite, Y. Suryawanshi, K. Patil, and P. Chumchu, "Sugarcane leaf dataset: A dataset for disease detection and classification for machine learning applications," *Data in Brief*, vol. 53, p. 110268, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340924002373

[42] ——, "Coconut (cocos nucifera) tree disease dataset: A dataset for disease detection and classification for machine learning applications," *Data in Brief*, vol. 51, p. 109690, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923007692