IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 April 2023, accepted 24 April 2023, date of publication 5 May 2023, date of current version 12 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3273317

RESEARCH ARTICLE

A Deep Features Extraction Model Based on the Transfer Learning Model and Vision Transformer "TLMViT" for Plant Disease Classification

AMER TABBAKH AND SOUBHAGYA SANKAR BARPANDA
School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India
Corresponding author: Soubhagya Sankar Barpanda (soubhagya.barpanda@vitap.ac.in)

ABSTRACT This paper proposes a novel approach for extracting deep features and classifying diseased plant leaves. The agriculture industry is negatively impacted by plant diseases causing crop and economic loss. Accurate and timely diagnosis is crucial for managing and controlling plant diseases, as traditional methods can be costly and time-consuming. Deep learning-based tools effectively detect plant diseases depending on the qualitative of extracted features. In this regard, a hybrid model for plant disease classification based on a Transfer Learning-based model followed by a vision transformer (TLMViT) is proposed. TLMViT has four stages: 1) data acquisition, where the PlantVillage and wheat datasets are used to train and evaluate the proposed model, 2) image augmentation to increase the number of training samples and overcome the overfitting issue, 3) leaf feature extraction by two consecutive phases: initial features extraction by using pre-trained based model and deep features extraction by using ViT model, and 4) classification by using MLP classifier. TLMViT is experimented with using five pre-trained-based models followed by ViT individually. TLMViT performs accurately in plant disease classification, obtaining 98.81% and 99.86% validation accuracy for VGG19 followed by the ViT model on PlantVillage and wheat datasets respectively. Moreover, TLMViT is compared with pre-trained-based architecture. The comparison result illustrates that TLMViT achieved an enhancement of 1.11% and 1.099% in validation accuracy, 2.576% and 2.92% in validation loss compared with the transfer learning-based model for PlantVillage and wheat datasets respectively. Thereby proposed model proves the efficiency of using ViT for extracting deep features from the leaf.

INDEX TERMS Plant disease, image processing, deep learning, transfer learning, vision transformer.

I. INTRODUCTION

Plant diseases have become more prevalent in recent years due to globalization, trade, and climate change [1], [2]. These issues have reached pandemic proportions in several nations, which increased the likelihood of crop damage and, in turn, created a threat to people's access to adequate food and nutrition [3]. Specialists should ensure to safeguard agricultural plants. Parasitic organisms such as bacteria, fungi, viruses, roundworms (nematodes), and other plants can cause illnesses. Environmental conditions such as winter frosts or summer dryness and lack or excess of nutrients in the soil can be the potential causes of plant diseases [4]. Phytopathology is the scientific study of plant disease that focuses on ways to treat and avoid the conditions responsible for plant illnesses. To overcome plant illness, plants have to be diagnosed precisely. There are several methods available for diagnosing. Local plant clinics and agricultural groups have traditionally helped in disease detection. Still, the technique could be more effective as there are more possibilities of human errors, and it is difficult for humans to access plants across a wide area. Moreover, using software using machine learning techniques can improve the efficiency of classifying diseased plants.

Smartphones are being developed using different tools and technology [6]. Modern plant illness classification approaches can be adopted using smartphones due to

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

their incorporation of high computational techniques, high-resolution screens, and built-in accessories such as HD cameras, which will be more effective for plant disease detection. Plant diseases are effectively diagnosed using machine learning techniques [4] as accurate diagnostic tests can be carried out, which benefits in preserving the resource. Farmers can upload field images recognized by smartphones, and distinct software can be used for analyzing, diagnosing, and developing action plans.

Recently, image processing techniques with deep neural networks have been effectively used diversely and proven to be highly effective approaches in constantly monitoring the health of plants and identifying signs of diseases in their early stages [5]. A neural network considers an image of a diseased plant as an input and processes the image to produce a crop-disease pair. Creating a deep network in such a manner that the network topology, functions (nodes), and edge weights accurately map the input to output is challenging. While training deep neural networks, the network parameters are adjusted in such a manner that the mapping is enhanced better over the training period. This complex computational process has recently seen several conceptual and practical advancements that have dramatically increased its performance [7], [8]. One of these approaches is Vision Transformer, which takes the whole image and extracts its features by splitting the image into multi patches, and from each patch, the transformer encoder will extract the features. Extracting the features from the whole image may take much more time and extract unnecessary features, e.g., a background of the leaf. Above all, this paper presents a novel approach to classify plant disease using pre-trained deep learning model followed by ViT. The pre-trained model extracts the initial features from the base convolutional networks that are already fine-tuned and reduces the dimensionality of the image making the next stage less complex. Then the initialled features are passed as input to ViT to extract the most significant features (deep features). A simple MLP is used to classify these deep features to which classes belong. The main contributions of this paper are pointed out as follows:

1) Using a pre-trained model to reduce the dimensionality of the image, making the next stage less complex.
2) Extract the deep feature of the leaf using a combination of the pre-trained model and ViT making the classifier more accurate.
3) Experiment with five different combinations of transfer learning-based models (such as ResNet50, AlexNet, Inception V3, VGG16, and VGG19) with ViT.
4) Comparisons between the TLMViT and five other transfer learning-based models are made.

The rest of this paper is structured as follows. Section II of the study offers a brief review of the relevant research. Section IV introduces the background of the methods used. Section V outlines the workflow and methodology used, followed by the experimental parameters in Section VI. Experiment outcomes with extensive discussion are presented in Section VII. The summary of this research is addressed in section VIII.

II. RELATED WORK

Plant disease diagnostics using machine learning and deep learning have been a primary focus of recent research. The review papers [1], [2], [9], [10] provided a summary of different novel approaches, developed existing algorithms used to classify plant diseases, and sought to determine which classification method would be the most useful for this challenge. Different datasets have been utilized to classify various plant leaves. References [11], [12], and [13] proposed a model to classify a tomato leaves disease. At the same time, Dey et al. [14] concentrated on betel vines. References [15], [16], [17], and [18] utilized Cassava disease. References [19], [20], [21], [22], [23] focused on classifying potato leaf disease. References [24] and [25] performed the proposed work on a dataset of Guava leaves. Singh and Misra [26] focused on some of the most common leaves of banana, jackfruit, lemon, mango, potatoes, tomato, and sapota. Sharif et al. [27] experimented with the proposed work on the oranges infested with scale, Plant Village Dataset, Citrus Diseases Image Gallery Dataset, and their collected citrus dataset. Hassan and Maji [28] utilized the plant village dataset (which includes maize, potato, and tomato crops), the rice dataset, and the cassava dataset were all used in their experimental work for the study. In contrast, The CropDP-181 dataset was developed by Kong et al. [29] by combining three existing datasets totalling 123,987 images (from the AIChalle, Inaturalist, and IP102) that together contain 47 diseases and 134 pest categories. In [30], the dataset was collected from various sources, including (Apple plants, Wheat, Cotton, Maize, and Rice).

Machine learning models are developed to classify plant leaf disease and achieve good performance. In this regard, the following is a discussion of some methods. Mokhtar et al. [11] proposed an approach that identifies a diseased leaf without knowing the class of the disease. An essential feature of the tomato leaf is its surface texture, so GLCM is used where Energy, Contrast, Sum of Squares, Correlation, Entropy, Sum Entropy, Cluster Shade, Cluster Prominence, and Homogeneity are extracted. SVM is used as a classifier with different kernel functions. Bhargava et al. [31] applied different machine learning approaches, such as LR and SVM to classify varieties of leaves of vegetables and fruits. The dataset is segmented using grab-cut and fuzzy c-means clustering. Then the features were extracted by discrete wavelet transform, the histogram of gradients, Laws' texture energy, textural, statistical, and colorgeometrical with 114 features. PCA is used for feature selection to reduce the dimensionality of up to 30 features. The classification results show the significance of feature selection and the increased classification recognition rate. Sharif et al. [27] detected the lesion on leaves using an optimized weighted segmentation method on enhanced images obtained in preprocessing stage. Then the features

are extracted and fused using color (HSV, LAB, LUV, and HSI), geometric, and texture (18 GLCM features). The best features are selected using principal component analysis (PCA) estimation. The multi-class support vector machine (M-SVM) is used to classify citrus diseases. They conclude that enhancing the contrast of the lesion part will improve segmentation accuracy. Bhargava and Bansal [32] focused on segmenting the lesion from the mono-colored apples using two algorithms, e.g., ''Otsu'' strategy and k- means clustering. Then a combination of color coherence, Gabor wavelet, 14 geometric features, and 13 statistical & textural features that are extracted from the segmented image are used as features. The proposed model is trained using linear SVM. From the result, they conclude that the proposed model achieves a good performance compared with other kernels of the SVM classifier. In [21], the K-means algorithm extracts the infected regions from the leaf and then extracts the lesion features using GLCM. They considered feature extraction as a type of dimensionality reduction process. Pahurkar and Deshmukh [30] propose a model that optimizes variance by fusing GLCM, edge map, color map, and convolutional feature sets into an ensemble of features and then using particle swarm optimization (PSO) to choose the optimal features. The parametric variant classification techniques are also tuned with a Genetic algorithm (GA). Tabbakh and Barpanda [33] proposed an approach of a machine-learning model where wavelet transforms, GLCM methods, and statistical features are used to extract different combinations of leaf features. Then the extracted features are utilized for training and comparing six machine-learning models, e.g., SVM, AdaBoost, etc. They modified the GLCM approach to focus on extracting the features of the leaf part only. SMOTE method is used to handle the imbalanced dataset. The highest result is achieved by using LGBM with 94.39% accuracy.

Deep learning has proven to be an effective tool for image classification tasks, outperforming traditional methods such as support vector machines and random forests. Deep learning algorithms require large datasets to train models effectively. In [34], cropping the lesion portion from the original image and making them a new sample is used to overcome the lack of dataset size. Whereas Geetharamani and Pandian [35] augmented the dataset using various functions such as image flipping, scaling, rotation, principal component analysis (PCA) color augmentation, noise injection, and gamma correction. From the result, the authors observed that the data augmentation methods could improve the model performance. References [14], [26], and [36] used an image segmentation algorithm to identify damaged leaf areas to analyze better and have little processing effort to get optimum results. Dey et al. [14] found that using HVS color space on their dataset, the hue component shows clearly where a leaf has rotted. In addition, the hue component masks the background and the rest of the leaf region. Singh and Misra [26] segmented the lesion by masking the mostly green pixels. Chaturvedi et al. [36] proposed a modified firefly algorithm based on multilevel thresholding with fuzzy, Tsallis, and Kapur's entropy for various fruit leaf segmentation. Khalifa et al. [37] augmented the dataset (from 1,722 to 9,822 images of potato leaves) using Reflection, Zoom, and Gaussian noise functions. They proposed a simple CNN using two convolutional layers for feature extraction and two fully connected layers for classification achieving 98% accuracy as overall mean testing. Whereas, Rozaqi and Sunyoto [20] used four convolutions and one fully connected layer on the same dataset of potato leaf. They achieved 97% and 92% accuracy on training and validation data, respectively. Mittal and Gupta [38] focused on studies that show how creating a complete and unusual image of a diseased leaf and increasing the dataset may help the classification network perform better. A binary generator network is proposed to address the issue of how a generative adversarial network (GAN) generates a diseased leaf on a specific shape. Also, using edge-smoothing and Image pyramid methods to overcome the challenge of synthesising a complete lesion leaf image comprised of various synthetic edge pixels and network out pixels. Khamparia et al. [19] benefited the properties of the encoding part of the autoencoders model to extract useful features. They proposed a convolutional encoder network that only combines CNN with the encoding part. Moreover, 97.50% accuracy was achieved after 100 iterations. Liang et al. [39] presented a network-based estimate method for disease identification using ResNet50 and residual where Shuffle-Net-network v2's architecture is used to reduce computational complexity. Johnson et al. [22] developed an automatic method for identifying blight disease patches on potato leaves in field circumstances utilizing the Mask Region-based convolutional neural network (Mask R-CNN) architecture and a residual network as the backbone network. The Mask R-CNN model accurately distinguished between the infected area on the potato leaf and the similar-looking background soil patches that often affect binary classification results. The proposed work is experimented after converting the original dataset from RGB to YCrCb, XYZ, LAB, HSV, and HSL color spaces, giving a separate model for each color space. Hassan and Maji [28] have suggested a new deep-learning model using the inception layer and residual connection. The number of parameters in the proposed work is reduced by a margin of 70% using depthwise separable convolution, which directly impacts the computing cost.

Recent works [18], [40], [41], [42] applied the ViT concept in agricultural applications. Reedha et al. [40] highlighted how efficiently the convolutional-free ViT model, using the self-attention mechanism, interprets an image into a sequence of patches for processing by a standard transformer encode. They obtained a high performance even though the dataset is small. They justified that due to data augmentation, transfer learning, and a low number of classes. Wu et al. [41] passed the dataset into two ViT models parallelly, where a small patch size is used in one model, and a large patch size is used for another model. Fusion models combined

these two models to be fed into the MLP header. Concluding that, by combining different scales of the sequence of self-attention, the model can extract more information from images from various granularities. Thai et al. [18] applied ViT based on achieving a 90% F1-score. They used quantization to make the model three times smaller before deploying it on a Raspberry Pi 4 Model B. Yasamin et al. [42] combined a classical convolutional neural network (CNN) with ViT. They concluded that ViT gives good accuracy but decelerates the prediction, and this approach can recompense for the speed.

Based on what is discussed in the related work, it can be noted that CNN models represent the process on images as template matching, which extracts the neighbouring features and does not consider the relation between overall features. In contrast, ViT can extract and relate the features by giving the position to patches of pixels and interactions between them. ViT drawback is the need for a vast dataset and consuming much time due to its correlation between all pixels with keeping the image scale without reduction. Moreover, pre-trained models could be utilized significantly to extract the features from the leaf and simultaneously reduce the dimensionality of the image, which will be considered as initial features to be passed for the next stage to extract the deep features from them. Thus, combining both approaches could extract comprehensive and significant features to obtain an accurate model.

III. MOTIVATION AND OBJECTIVES

Most of the DL-based research work in the plant disease classification field, which uses the convolution process, may be seen as template matching, whereby the same filtering template is applied to several parts of the same image. However, the convolution layer represents only the connections between neighbouring pixels since convolution is a local action and is not able to encode the orientation and position of infected parts in the leaf. In case the new leaf has different positions of infection from a trained set, then the classifier would have a hard time classifying the diseased leaf.

In contrast, a transformer layer may overcome these issues by representing the interactions between all pixels, so the transformer is considered a global operation. In a transformer, the attention unit is an adaptive filter, and the filter weights are set according to how well two pixels compose. The modelling capabilities of this sort of computer module are superior. Since Vision Transformer is applied on the whole image to initialize the patches and process the attention patches, the time-consuming of using Vision Transformer depends on the size of the image and the number of patches. Hence, it is better to reduce the dimension of an image, whereas scaling down the dimension in an ordinary way will lead to the loss of some significant features of the plant leaf in the image. The transfer learning extracts the features and reduces the dimension of the image before the classification part of its model. In light of this, the aims listed below constitute the focus of the study:

1) Different transfer learning models are used to extract initial features and reduce the dimensionality of plant disease images instead of applying the Vit model on raw images with entire dimensions, which reduces the time consumption and trainable parameter numbers.
2) ViT model is used to extract the deep features of leaf images, i.e., interactions between all pixels that are extracted from the previous stage (Initial Features).
3) Different combinations of Transfer learning-based models with ViT have been experimented.
4) Different five transfer learning-based models are compared with the proposed work TLMViT.

IV. PRELIMINARIES

A. TRANSFER LEARNING

Transfer learning models or pre-trained models such as ResNet50, AlexNet, Inception V3, VGG16, and VGG19 are very active and useful in overcoming the flaws in deep learning [43]. Training a deep learning model with millions of parameters needs a lot of training samples that are only sometimes applicable to be collected and too much training time [44]. The transfer learning technique is utilized to transfer the knowledge from a pre-trained model to another model, which should be trained for a classification task. Hence, the training time is decreased because the model is not trained from scratch. Moreover, it is utilized to train the deep learning model on a small-size dataset and overcome the overfitting issue [45], [46]. Fine-tunning is the most popular architecture in transfer learning, where a large dataset is used to pre-train the model and then, the parameters of the pre-trained model are frozen and transferred to the target model for fine-tuning with the dense layers of the target model. Finally, the dense layers which have reasonable parameters should be trained on the dataset that belongs to the desired classification task [47]. Figure 1 shows the process of using the transfer learning model for training a new model to classify a new dataset.

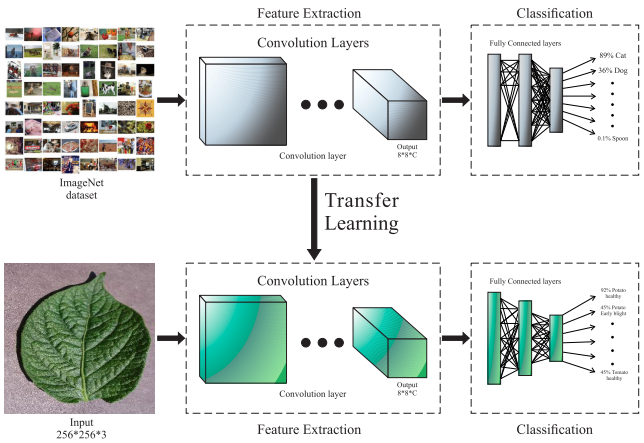


FIGURE 1. An illustration of using a Pre-Trained based model to classify a new dataset.

B. VISION TRANSFORMER (ViT)

The transformer was initially used to visualize backbone networks through the local relational network (LR-Net) [48]

and SASA [49]. Both models improved accuracy over ResNet while utilizing the same theoretical computing resource by restricting self-attention computation to a local sliding window. LR-Net has the same theoretical and computational complexity as ResNet, but it is much slower in reality. ViT model was proposed by Dosovitskiy et al. [50] in 2021 as an efficient image classifier. He suggested using the original transformer architecture in computer vision applications since it was previously applied only in natural language processing applications. In image classification, ViT performs better than standard CNN architectures when trained on a huge amount of image data. The input image is divided into patches, each one is flattened and merged across the channels of the image to produce a vector representation of each patch. The patch embeddings are calculated via linear projecting the vectors using a dense layer. The positional embeddings are generated, which help the ViT model to analyze the patches orderly and have a full view of the input image. Then, each patch is added to the corresponding positional embedding to obtain the input of the transformer's encoder. The encoder consists of one block executed several times such that the architecture of the block contains a multi-head self-attention followed by a dense layer. Finally, MLP classifies the input image based on the output of the transformer's encoder.

V. METHODOLOGY

This paper proposes a hybrid model for plant disease classification based on transfer learning models and vision transformers (TLMViT), as shown in Figure 2. TLMViT includes four stages:

1) Data acquisition: The PlantVillage and Wheat Rust datasets are used to train and validate the proposed model. Different three crops are considered in the PlantVillage dataset (pepper bell, potato, and tomato leaf).
2) Image augmentation: It artificially increases a dataset's size by applying random transformations to existing images. This allows the model to generalize better and reduce overfitting, as it has seen different variations of the same image.
3) Feature extraction: It is a process of identifying and extracting important information from raw data, like edges, corners, and textures in image data, which can be used as input for a classifier model. Moreover, two phases are consequently used to extract the features:
    a) First phase: a Pre-Trained based model is used to extract the initial features of the leaf, then pass them to the second stage.
    b) Second phase: ViT is utilized to extract the deep features from the initial features.
4) Classification: simple MLP classifier is used to be trained and evaluated.

Each stage is discussed in detail as follows:

A. DATA ACQUISITION AND IMAGE AUGMENTATION

In this work, PlantVillage and Wheat Rust datasets are used to evaluate the proposed model for extracting the deep features of leaves.

1) PlantVillage

The PlantVillage dataset is a collection of images of healthy and diseased plants created by the PlantVillage non-profit project. The dataset contains over 54,000 images of over 38 different crop species, focusing on cassava, tomato, pepper, and potato. Each image is labelled with the species of plant and the disease that is present if any. The dataset is freely available for computer vision and deep learning tasks such as Image classification, Object detection, and semantic segmentation. In this research, three different crops of the PlantVillage dataset [51] from Kaggle are utilized and downloaded for training and evaluating the proposed model. It comprises three main categories of plants: pepper bell, potato, and tomato leaves. It includes 20,638 images of diseased and healthy leaves from these three crops. Table 1 illustrates the differentiation of two bell pepper leaves, three varieties of potato leaves, and ten types of tomato leaves. Figure 3 shows samples of all 15 types of leaves included in the dataset.

From Table 1, it can be noted that some classes have less number of images compared to other classes. Moreover, the limitation of images while training the model leads to overfitting issues. In order to restrain the influence of imbalanced data and overfitting issues, data augmentation is applied to increase the samples [52]. The data augmentation, such as scaling, shearing, rotating, horizontal flip, zooming, and shifting [53], [54] are applied, and its parameters are noted in Table 5. From the initial set of 16511 training images, the data augmentation generated 33257 augmented training images.

2) WHEAT RUST

Wheat rust disease is a fungal disease that affects wheat plants, causing significant damage to crops and resulting in significant economic losses for farmers. The disease is caused by several species of fungi belonging to the Puccinia genus. There are three main types of wheat rust: stem rust, leaf rust, and stripe rust [55]. Stem rust is the most destructive type, as it attacks the stems of the wheat plant, weakening the plant and reducing yield. Leaf rust affects the leaves of the plant and can reduce photosynthesis, while stripe rust affects the leaves and stems of the plant. Wheat rust is spread by wind-borne spores, which can travel long distances, making it difficult to control. Preventing the spread of wheat rust is crucial to maintaining healthy crops and ensuring food security. Early detection and prompt treatment are essential for effectively managing the disease. In this work Wheat Rust dataset [42] is utilized for training and evaluating the proposed model. The dataset contains three classes (1128 Brown rust, 1348 Yellow rust, and 1203 Healthy wheat), as illustrated in Table 2. Figure 4 shows samples of the Wheat Rust dataset.

TABLE 1. Details of the leaves images used in the PlantVillage dataset.

SL. No.	CLASS NAME	Number of Images	SL. No.	CLASS NAME	Number of Images
1	Tomato mosaic virus	373	9	Tomato Bacterial spot	2127
2	Tomato Yellow Leaf Curl Virus	3208	10	Tomato healthy	1591
3	Tomato Target Spot	1404	11	Pepper bell Bacterial spot	997
4	Tomato Septoria leaf spot	1771	12	Pepper bell healthy	1478
5	Tomato Early blight	1000	13	Potato Late blight	1000
6	Tomato Leaf Mold	952	14	Potato healthy	152
7	Tomato Spider mites Two spotted spider mite	1676	15	Potato Early blight	1000
8	Tomato Late blight	1909			
		TOTAL : 20,638			

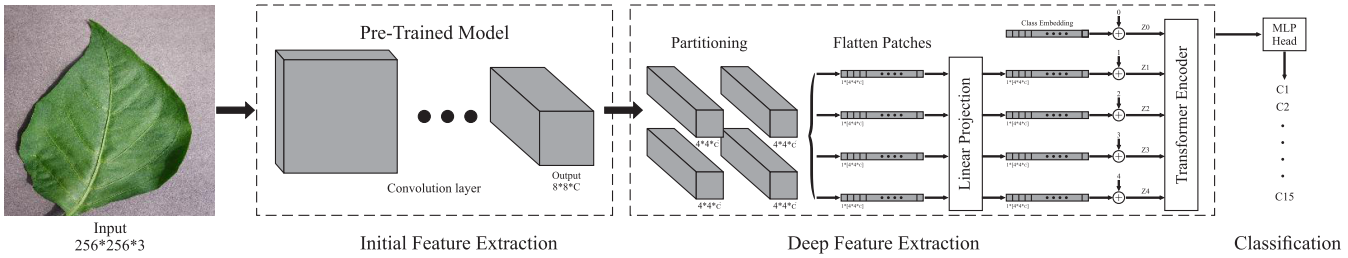


FIGURE 2. The architecture of the proposed model TLMViT.

TABLE 2. Details of the leaves images used in the Wheat Rust dataset.

SL. No.	CLASS NAME	Number of Images
1	Brown Rust	1128
2	Yellow Rust	1348
3	Healthy	1203
	TOTAL : 3679	

B. FEATURES EXTRACTION

The proposed model uses two consecutive phases for desirable feature extraction. In the first phase, the pre-trained model is used for the initial extraction of leaf features and for reducing the dimensionality of images, leading to decreased time while applying the second phase. In the second phase, the ViT model is applied on initial features for deep extraction of features.

1) 1ST PHASE (INITIAL FEATURES EXTRACTION)

In this phase, a pre-trained model is utilized to extract initial features (IF) from the input images, where different pre-trained models, namely (AlexNet, Res-Net 50, VGG-16, VGG-19, and Inception-V3) are experimented on the proposed work to observe which model will be superior. The weights of all pre-trained models are initialized by using the ImageNet dataset and then used to extract the features from the PlantVillage and Wheat Rust datasets as described below:

a: ResNet 50

Residual Network is referred to as ResNet. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun initially described this novel neural network in [56]. There are several distinct implementations of ResNet, all of which use the same basic structure but have various layers. Resnet50 is shorthand for the version of Resnet that supports 50 neural network layers. The dimension of the default input of ResNet50 is 224*224*3. Five different stages based on convolution and Residual Networks are applied, and after each stage, Max-Pooling Layers are applied, which leads to reducing the dimension of the image by 2. The output of the final stage of the default input gives a 7*7*2048 size. Whereas the dimension of the PlantVillage dataset is 256*256*3, the final stage of ResNet50 gives 8*8*2048. These are used as the initial feature and pass it as input to deep feature extraction using ViT.

b: AlexNet

Alex Krizhevsky and his colleagues [53] published their findings in a research article titled Imagenet Classification with Deep Convolution Neural Network, in which they suggested the model. There are eight layers in the Alexnet model, involving five layers based on convolution with a combination of max pooling, followed by three fully connected layers. Relu activation is used for each of these layers, excluding the output layer. In this work, only the first five layers are used to extract the initial features from the PlantVillage dataset, and the last three layers are eliminated. The dimension of input images is 256*256*3, and the output of the last layers is 8*8*256, as mentioned in Table 3.

c: VGG16 AND VGG19

VGG contains a combination of convolutional and three fully connected layers. The name of the VGG model depends on how many layers are used in the model [57], e.g., 13 convolutional and three fully connected layers produce VGG16, whereas 16 convolutional and three fully connected layers

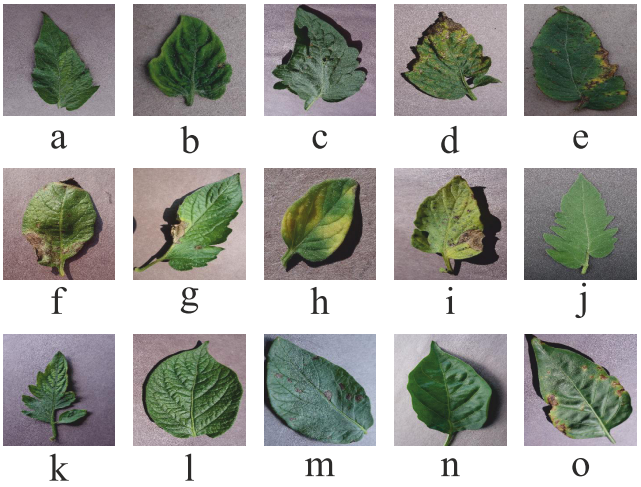


FIGURE 3. Image Samples of PlantVillage Dataset: (a) Tomato Spider mites Two spotted spider mite, (b) Tomato Yellow Leaf Curl Virus, (c) Tomato Target Spot, (d) Tomato Sep-toria leaf spot, (e) Tomato Bacterial spot, (f) Potato Late blight, (g) Tomato Late blight, (h) Tomato Leaf Mold, (i) Tomato Early blight, (j) Tomato healthy, (k) Tomato mosaic virus, (l) Potato healthy, (m) Potato Early blight, (n) Pepper bell healthy, and (o) Pepper bell Bacterial spot.

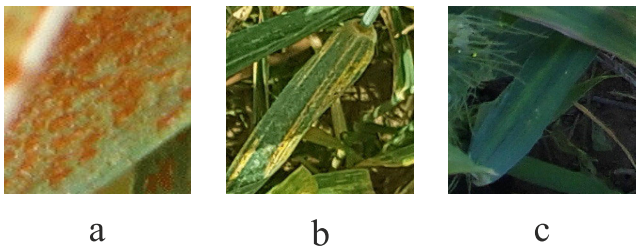


FIGURE 4. Image Samples of Wheat Rust Dataset: (a) Brown Rust, (b) Yellow Rust, and (c) Healthy wheat.

produce VGG19 [58]. A simple 3*3 convolution kernel is utilized across all layers to deepen the network and reduce the number of parameters. A 224*244 RGB image is used as VGG's default input. Moreover, the last three fully connected layers of both VGG16 and VGG19 models are not used in our work. Table 4 shows the default input of both models (VGG16 and VGG19), which is 224*224*3 and generates 7*7*512 as an output of the last layer, whereas the input size of the PlantVillage dataset is 256*256*3, so the last layer gives 8*8*512 an output size as shown in Table 4. This output is considered the initial features of the models to be forwarded to the next phase to extract the deep features.

d: INCEPTION V3

In the image classification competition at the 2014 ILSVRC, Google presented a network named GoogLeNet that could achieve the performance of human experts on the ImageNet database [59]. An improved version of GoogLeNet got inspired to generate the Inception-v3 model. It uses various sizes of receptive kernels. By utilizing zero padding, the convolution operation's output size is maintained. Finally, filter concatenation produces the feature maps that will be

TABLE 3. Details of the feature map size for default and our input size of the AlexNet model.

Layer	Filter size	Stride	Size of feature map of default input	Size of feature map of our input
Input	-	-	227 * 227 * 3	256 * 256 * 3
Conv 1	11 * 11	4	55 * 55 * 96	64 * 64 * 96
Max Pool 1	3 * 3	2	27 * 27 * 96	32 * 32 * 96
Conv 2	5*5	1	27 * 27 * 256	32 * 32 * 256
Max Pool 2	3 * 3	2	13 * 13 * 256	16 * 16 * 256
Conv 3	3 * 3	1	13 * 13 * 384	16 * 16 * 384
Conv 4	3 * 3	1	13 * 13 * 384	16 * 16 * 384
Conv 5	3 * 3	1	13 * 13 * 256	16 * 16 * 256
Max Pool 3	3 * 3	2	6 * 6 * 256	8 * 8 * 256

TABLE 4. Details of the feature map size for default and our input size of the VGG16 and VGG19 models.

VGG16				VGG19			
layer	Filter size	default input of VGG16 Model	PlantVillage input for VGG16 Model	layer	Filter size	default input of VGG19 Model	PlantVillage input for VGG19 Model
Input		224*224*3	256*256*3	Input		224*224*3	256*256*3
2 Conv	3*3*64	224*224*64	256*256*64	2 Conv	3*3*64	224*224*64	256*256*64
Max Pool	stride 2	112*112*64	128*128*64	Max Pool	stride 2	112*112*64	128*128*64
2 Conv	3*3*128	112*112*128	128*128*128	2 Conv	3*3*128	112*112*128	128*128*128
Max Pool	stride 2	56*56*128	64*64*128	Max Pool	stride 2	56*56*128	64*64*128
3 Conv	3*3*256	56*56*256	64*64*256	4 Conv	3*3*256	56*56*256	64*64*256
Max Pool	stride 2	28*28*256	32*32*256	Max Pool	stride 2	28*28*256	32*32*256
3 Conv	3*3*512	28*28*512	32*32*512	4 Conv	3*3*512	28*28*512	32*32*512
Max Pool	stride 2	14*14*512	16*16*512	Max Pool	stride 2	14*14*512	16*16*512
3 Conv	3*3*512	14*14*512	16*16*512	4 Conv	3*3*512	14*14*512	16*16*512
Max Pool	stride 2	7*7*512	8*8*512	Max Pool	stride 2	7*7*512	8*8*512
Total layers	16			19			

used for the classification part. Feng et al. [60] represent the structure of the Inception V3 model in detail, the default size of an input image in the Inception V3 model is 299*299*3, and the final output size is 8*8*2,048.

In contrast, the image size in the PlantVillage dataset is 256*256*3, which is less than the default, leading to less output size of the feature map. Moreover, the output size of the features map for all previous pre-trained models that were used in our experiments gives 8*8*c (c is 2048, 256, 512, 512 for ResNet 50, AlexNet, VGG16, and VGG19 models, respectively) when using PlantVillage dataset. To maintain consistency throughout all experiments, the images of PlantVillage dataset are rescaled to the size of the default input of inception V3, which is 299*299*3, and fed as input to extract the initial features in size of 8*8*2,048 to be used in the second stage of the proposed model.

In contrast, the image dimensions of the Wheat Rust dataset are in different scales, so it may be difficult to compare and analyze the data accurately. To address this issue, rescale function is used to ensure that all inputs (image dimensions) are of the same shape. The image dimensions of the Wheat Rust dataset are rescaled as the same image dimensions of PlantVillage to have the same output dimensions in the 1st phase of feature extraction with the size of 8*8*c.

2) 2ND PHASE (DEEP FEATURES EXTRACTION)

In this phase, the Vit model is applied on the initial features that are extracted from the previous phase to extract deep features of leaves images, where ViT is described as follows: As mentioned in the first phase, the features are extracted by using transfer learning model, such that the extracted features IF are represented by a 2D image $I \in R^{L*W*C}$ dimensions, where L is the length, W is the width of image and C is the number channels. The final dimensions of

the initial features that are extracted by pre-trained model e.g, AlexNet, VGG-16, VGG-19, Res-Net, and Inception-V3 are $R^{8*8*256}$, $R^{8*8*512}$, $R^{8*8*512}$, $R^{8*8*2048}$, $R^{6*6*2048}$ respectively. Then these features are fed as input to ViT model to be processed according to the steps that are shown in Figure 2.

The first step is image partitioning, such that the extracted features are split into non-overlapping patches. The initial feature dimensions from the previous phase are $8*8*c$. Since the dimension is not too big $(8*8)$, and splitting it into lots of patches will give less dimension with less information on features, so the number of patches is chosen by taking the half size of the IF with $N(P) = 4$. Each patch P has size $4*4*C$ where C is the number of IF's channels and $(4*4)$ are the dimensions of P. The process of image partitioning is shown in equation [1].

$$IF = P_1||P_2||P_3||P_4; \quad P \in R^{4*4*C} \text{ and } IF \in R^{8*8*C} \tag{1}$$

where $C = 256, 512, 512, 2048, and 2048$ for AlexNet, VGG-16, VGG-19, Res-Net, and inception respectively. Then, every patch is flattened to a 1D vector such that, the size of the vector is $S = 4*4*C$. Hence, four flattened patches (vectors) are obtained with size S. Next, the patch embeddings PE_i are calculated by linear projecting on the four vectors using dense layer that has 64 units as shown in equation [2].

$$PE_i = Project(P_i) = Dense_{64}(P_i); \quad P_i \in R^{4.4.C*1}, \quad PE_i \in R^{64}, \ i = 1, \cdots 4 \tag{2}$$

where i is the index of patch and the dimension of PE_i is 64.

The second step: in this step, the positional information of patches Z_i is retained by adding the patch embeddings PE_i to the position embeddings PE_i. The aim of this step is to process the patches as per their positions in the images (IF), and the process of getting positional information is shown in equation [3].

$$positional\ information(Z_i) = PE_i + Pos\_E_i; \quad i = 1, \cdots 4 \tag{3}$$

where PE_i and Pos_E_i have the same dimension (64*1), and i is the index of patch. Position embeddings Pos_E_i are generated by mapping integer numbers which represent the positions of patches into vectors with size 64. Finally, the class token Z_0 is embedded and appended to the positional information Z_i; i=1 . . . ,4 such as the dim (Z_0) = dim (Z_i); $i = 1 \ldots, 4$ as shown in equation [4].

$$Z = [Z_0||Z_1||Z_2||Z_3||Z_4] \tag{4}$$

The third step is transformer encoding which consists of a one block that is executed N times ($N = 8$) to extract the deep features. This block has two main components i.e., multi-head self-attention and MLP beside that, normalization operations are used to improve the performance of transformer encoder. The process of transformer encoder is shown in equations [5 - 6] and Figure 5.

$$X_i(j) = F_{H=4}(Norm(Z_i)) + Z_i; \quad i = 0, \cdots 4 \text{ and } j = 1, \cdots 8 \tag{5}$$

$$\acute{X_i}(j) = MLP(Norm(X_i(j))) + X_i(j); \quad i = 0, \cdots 4 \text{ and } j = 1, \cdots 8 \tag{6}$$

where $F_{H=4}$ is the function of multi-head self-attention with heads $H = 4$, MLP is multi-layer perceptron neural network that uses two hidden layers with 128 neurons in the first layer and 64 neurons in the second layer. Norm is the normalization operation of vector. $\acute{X_i}(j)$ is the output of transformer's block at the j-th iteration. At j=8 (the last iteration of encoder block), the deep features are obtained.

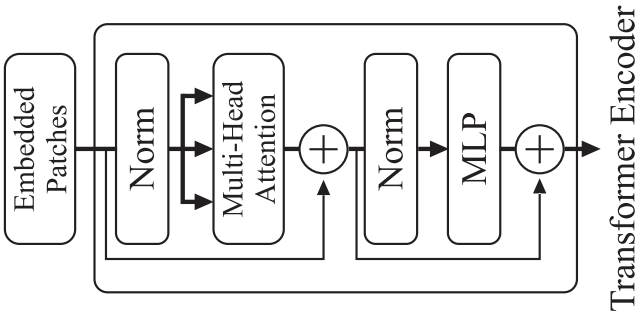


FIGURE 5. Structure of the transformer encoder.

C. CLASSIFICATION

The classifier that has been used in the proposed model is MLP head. Once the transformer encoder generates the deep features in the previous stage, they are fed to the classifier, which determines to which class the image belongs. The deep feature vectors $\acute{X_i}(8)$ are fed to the MLP classifier as input values, where the input layer has 256 neurons equal to the dimension of the deep feature vector $\acute{X_i}(8)$. Two hidden layers with dimensions 1024 and 512 for the first and second layers respectively are used. Lastly, 15 neurons are used as the MLP classifier's output, which is the dataset's number of classes. The process of classification is shown in equation [7].

$$P(j) = MLP_{head}\left(\acute{X_i}(8)\right); \quad j = 1, \cdots 15 \tag{7}$$

VI. EXPERIMENTAL ANALYSIS

All the experiments in this work are carried out on Windows 10 PCs version 21H2, with 64-bit operating system and a processor Intel(R) Xeon(R) 4.01 GHz with 64 GB RAM. CUDA with version 11.2 is used on NVIDIA Geforce GTX 1080 Ti. Python 3.9.12 was used as the programming language and all the experiments were performed on the Spyder IDE including various libraries e.g., Keras, and TensorFlow. Table 5 illustrates the parameters of all models that were experimented in this research. In the data augmentation function, the parameters are randomly initialized in ranges. Dataset Splitted (80% for training and 20% for both evaluating and testing the model). Since the dataset size is not

small, 20% would be sufficient for testing the model (almost 4100 images for testing the model). The Epoch number is chosen (25) to compare the results of all experiments at a specific epoch number to give a fair comparison. Since the dataset is under multi-class classification problems, categorical cross-entropy is used as a loss function that calculates the predicted probability distribution and the true probability distribution. Adam is used as an optimizer since it requires fewer parameters for tuning, which leads to faster computation time. The dimension of the initial feature (features dimension at the last stage of the pre-trained model) is 8*8*C, as explained in section V-B1. So, the patch size of the ViT model is chosen four as half of the initial feature dimension. The rest of the parameters are chosen experimentally.

TABLE 5. Experimental parameters.

Function	Parameter	Value
image augmentation	Rotation by a random angle in degrees	range [-25, 25]
	Random shifting across the width	10%
	Random shifting across the height	10%
	Random shearing	20%
	Horizontal and vertical flipping	TRUE
	Random zooming	20%
Dataset Split	Training	80%
	Validating and Testing	20%
Training parameters	Batch size	32
	Epochs	25
	Optimizer RMSprop	adam
	Initial learning rate	1.00E-03
Classifier (MLP)	The no of neurons in first hidden layer of MLP	1024
	The no of neurons in second hidden layer of MLP	512
ViT parameters	Patch size	4
	Projection dimension	64
	Num heads	4
	Transformer layers	8

VII. RESULTS AND DISCUSSION

This section discusses the results of the proposed model and compares its performance with the pre-trained-based models' performance.

A. RESULTS OF OUR PROPOSED MODEL

The proposed model based on a pre-trained model followed by ViT is evaluated in terms of accuracy, loss, precision, F1-score, and recall. Five different pre-trained-based models are experimented to measure the performance of the proposed model. Firstly, the weights of the CNN model are initialized by the frozen weights of a transfer learning-based model that was trained on the ImageNet dataset. Hence, no training is required in the initial features extraction phase, this manner overcomes the issue of consuming a lot of processing time for training a complex model from scratch on the PlantVillage dataset, which contains more than 20 thousand images. The aim of using pre-trained-based models in our approach is to (1) Extract the initial features from the input image and reduce the number of trainable parameters. (2) Reduce the dimension of the input image to accelerate the performance of ViT in the next phase and drive Vit to focus on the most significant features. Then, ViT model is trained to extract the deep features and classify the plant diseases based on the training set of the PlantVillage dataset. After that, the overall model is validated based on the testing set of the PlantVillage dataset. Figure 6 (a, c, e, g, and i) show the classification accuracy of training and validation set for five experiments, i.e., ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively in 25 epochs. The outcome of training accuracy is improved after each epoch till it reaches 90.405%, 92.5%, 99.60%, 98.8%, and 99.2% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively. The outcome of validation accuracy is also improved after each epoch till it reaches 89.2%, 90.49%, 98.43%, 98.81%, and 98.48% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively as shown in Table 6. Figure 6(b, d, f, h, and j) show the training and validation loss outcome in 25 epochs for five experiments, i.e., ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively. The outcome of training loss is decreased after each epoch till it reaches 0.3448, 0.2288, 0.0814, 0.1, and 0.0900 for ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively. The outcome of validation loss is decreased after each epoch till it reaches 0.3735, 0.2974, 0.1300, 0.0947, and 0.1 for ResNet50, AlexNet, VGG16, VGG19, and Inception V3, each of which is followed by ViT-based model respectively as shown in Table 6. Moreover, according to the results in the five experiments, (VGG-16 followed by ViT) model outperforms in the training phase where the training accuracy is equal to (99.60%), and training loss is equal to (0.0814). Whereas, (VGG-19 followed by ViT) model outperforms in the validation phase where the validation accuracy is equal to (98.81%), and validation loss is equal to (0.0947).

Furthermore, Figure 8 shows the confusion matrices of the proposed model that experimented using ResNet50, AlexNet, VGG16, VGG19, and Inception V3 each of which is followed by ViT-based model respectively. Finally, some statistical manners are used to analyze confusion matrices such as (precision, recall, and F1-score) for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 each of which is followed by ViT-based models respectively as shown in Table 8.

B. RESULTS OF TRANSFER LEARNING-BASED MODELS

In this work, five different pre-Trained based models are also experimented to classify the plant diseases on the PlantVillage dataset, which means the extracted features that used in its model are the initial features only as mentioned previously. The performance of five different transfer learning-based models i.e, ResNet50, AlexNet, VGG16, VGG19, and Inception V3 are shown in Table 6. The classification accuracy of the training set is increased after each epoch till it reaches 80.43%, 83.9%, 90.44%, 88.70%, and 92.78% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 respectively, and the classification accuracy of validation set is also improved after each epoch till it reaches to 79.84%, 81.91%, 89.31%, 86.55%, and 90.77% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3

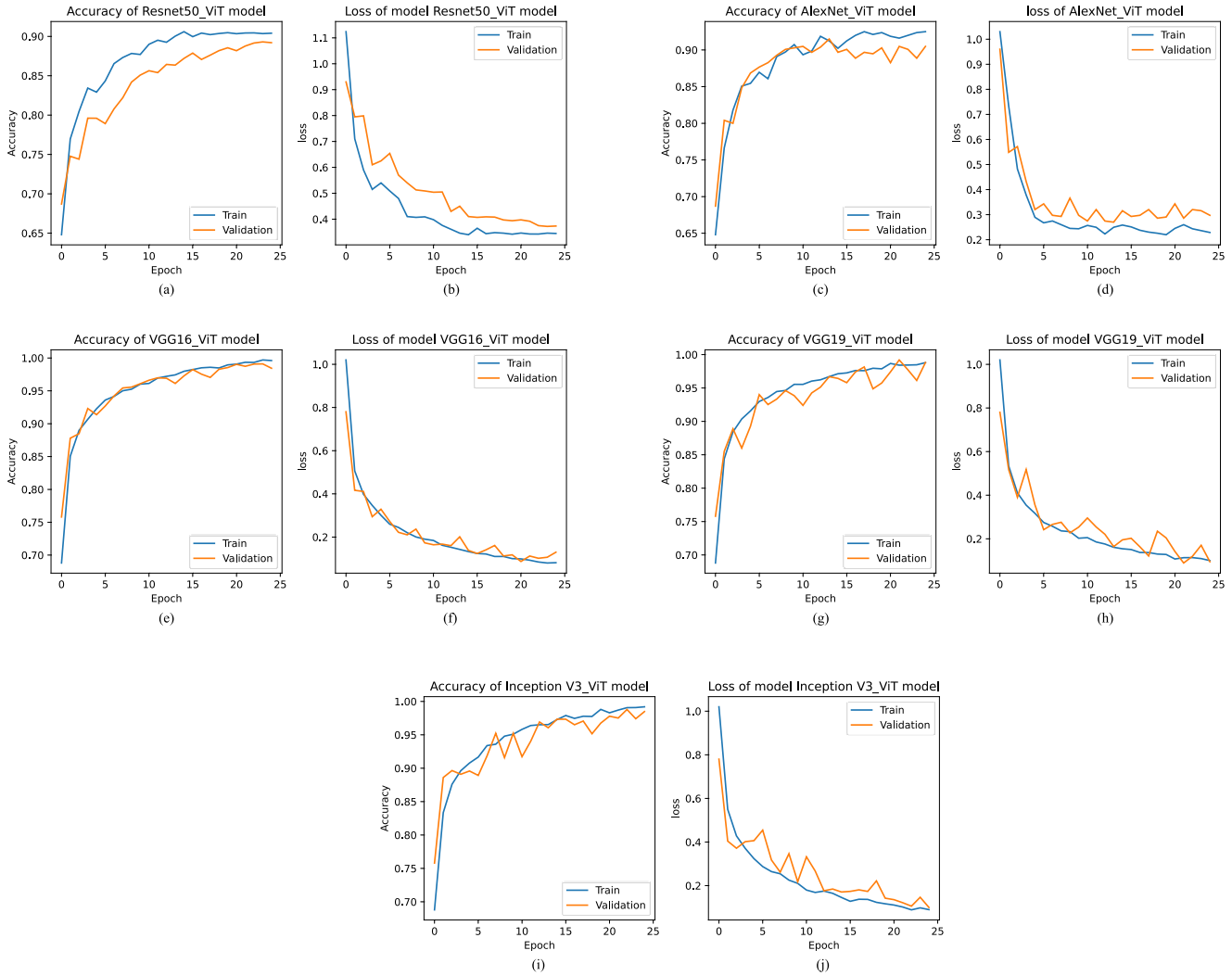


FIGURE 6. The performance of TLMViT models. (a, c, e, g, and i) Accuracy and (b, d, f, h, and j) Loss of ResNet50_ViT, AlexNet_ViT, VGG16_ViT, VGG19_ViT, and Inception V3_ViT models respectively.

TABLE 6. Performance comparison for training and validation set on PlantVillage dataset in terms of accuracy and loss between transfer learning-based models and TLMViT.

Model	Transfer Learning Based Models				TLMViT			
	Training	Validation	Training Loss	Validation Loss	Training	Validation	Training Loss	Validation Loss
ResNet50	80.43%	79.84%	0.5492	0.6	90.405%	89.2%	0.3448	0.3735
AlexNet	83.9%	81.91%	0.47	0.55	92.5%	90.49%	0.2288	0.2974
VGG16	90.44%	89.31%	0.2605	0.3188	99.6%	98.43%	0.0814	0.13
VGG19	88.7%	86.55%	0.3277	0.3792	98.8%	98.81%	0.1	0.0947
Inception V3	92.78%	90.77%	0.2064	0.2967	99.2%	98.48%	0.09	0.1

TABLE 7. Performance comparison for training and validation set on Wheat Rust dataset in terms of accuracy and loss between transfer learning-based models and TLMViT.

Model	Transfer Learning Based Models				TLMViT			
	Training	Validation	Training Loss	Validation Loss	Training	Validation	Training Loss	Validation Loss
ResNet50	83.68%	81.88%	0.43	0.51	93.26%	91.74%	0.194	0.271
AlexNet	86.27%	84.12%	0.38	0.463	95.48%	93.39%	0.161	0.197
VGG16	92.84%	91.73%	0.211	0.261	99.93%	99.44%	0.0643	0.082
VGG19	91.17%	89.55%	0.27	0.35	98.67%	99.86%	0.0794	0.0715
Inception V3	95.43%	93.26%	0.163	0.194	99.81%	99.69%	0.0734	0.079

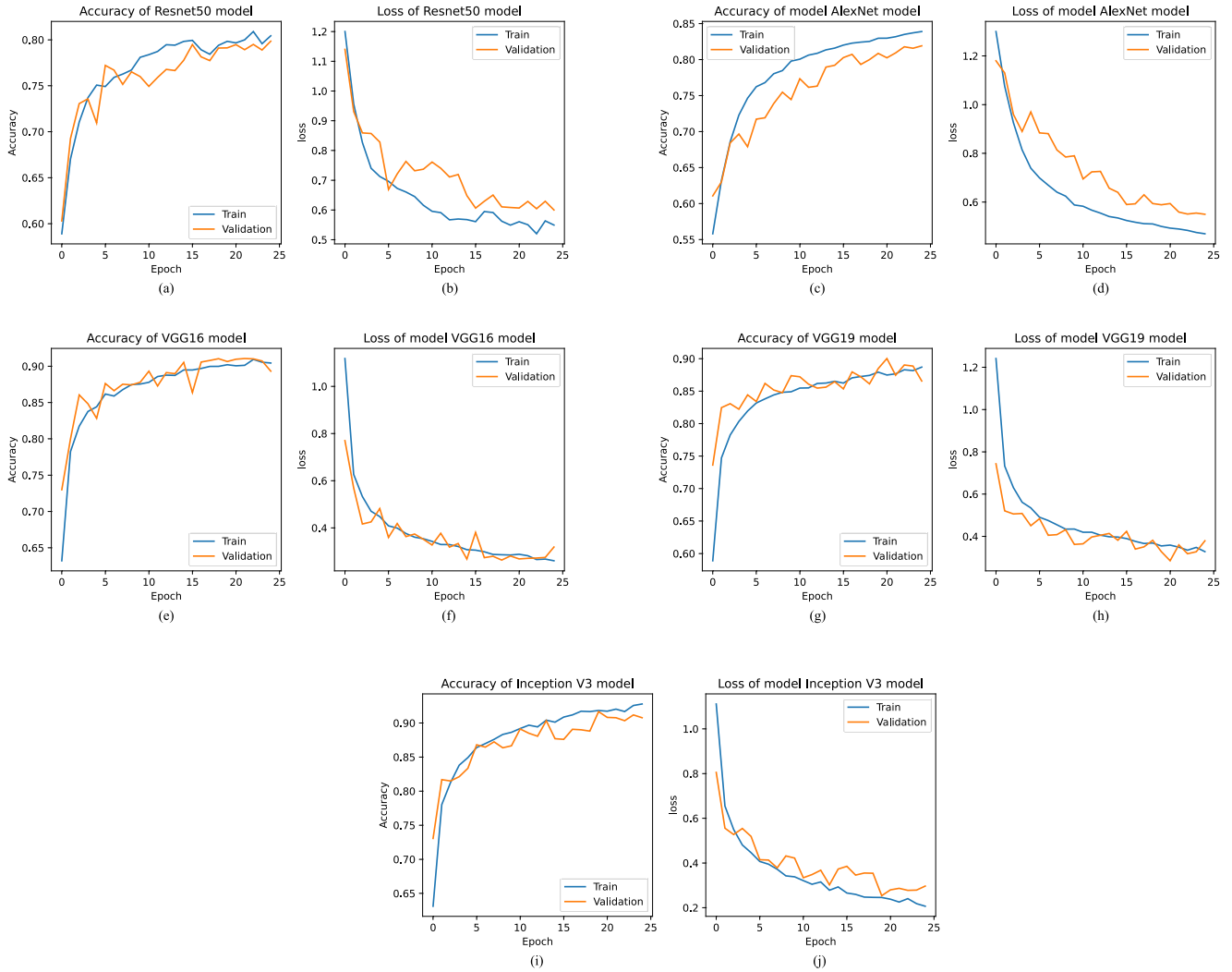


FIGURE 7. The performance of Transfer Learning-based models. (a, c, e, g, and i) Accuracy and (b, d, f, h, and j) loss of ResNet50, AlexNet, VGG16, VGG19, and Inception V3 models respectively.

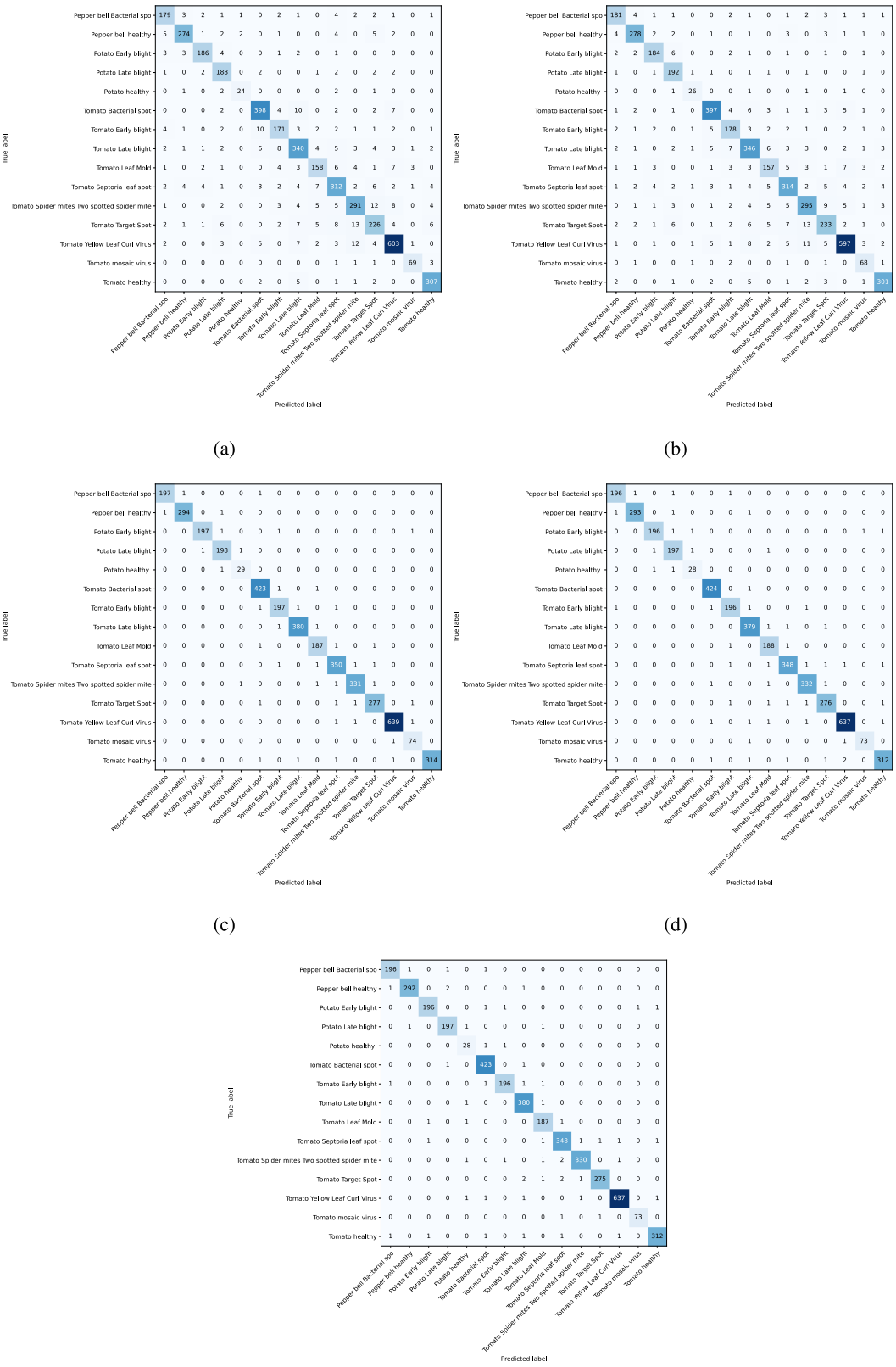
TABLE 8. Results Comparison in terms of F1-score, precision, and recall between transfer learning-based models and TLMViT using PlantVillage dataset.

Model	Transfer Learning Based Model			TLMViT model		
	F1-score	Precision	recall	F1-score	Precision	recall
ResNet50	79.56%	79.42%	79.71%	89.56%	89.38%	89.75%
AlexNet	81.78%	81.64%	81.94%	90.31%	90.11%	90.53%
Inception V3	90.26%	90.39%	90.15%	98.32%	98.36%	98.29%
VGG16	89.24%	89.32%	89.18%	98.23%	98.15%	98.32%
VGG19	86.4%	86.37%	86.44%	98.73%	98.72%	98.76%

TABLE 9. Results Comparison in terms of F1-score, precision, and recall between transfer learning-based models and TLMViT using Wheat Rust dataset.

Model	Transfer Learning Based Model			TLMViT model		
	F1-score	Precision	recall	F1-score	Precision	recall
ResNet50	82.99%	82.79%	83.21%	93.13%	92.99%	93.27%
AlexNet	83.23%	83.31%	83.16%	93.57%	93.48%	93.67%
Inception V3	93.36%	93.41%	93.21%	99.92%	99.93%	99.91%
VGG16	91.81%	91.87%	91.75%	99.58%	99.65%	99.51%
VGG19	88.42%	88.49%	88.36%	99.71%	99.78%	99.65%

respectively. Figure 7(a, c, e, g, and i) show the classification accuracy of training and validation set in 25 epochs for ResNet50, AlexNet, Inception V3, VGG16, and VGG19 respectively. Whereas Figure 7(b, d, f, h, and j) show classification loss of training and validation set in 25 epochs for ResNet50, AlexNet, Inception V3, VGG16, and VGG19 based models respectively. The value of training loss is decreased after each epoch till reaches 0.5492, 0.47, 0.2605, 0.3277, and 0.2064 for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 respectively. The value of validation loss is decreased after each epoch till reaches 0.6, 0.55, 0.3188, 0.3792, and 0.2967 for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 respectively as shown in Table 6. Moreover, the values of precision, F1-score, and recall for ResNet50, AlexNet, Inception V3, VGG16, and VGG19 are shown in Table 8. Moreover, a different comparison is done for the pre-trained based models with each other, on the PlantVillage dataset. According to the results, Inception V3 achieves the



(a) (b)

(c) (d)

(e)

FIGURE 8. The confusion matrix of the proposed model. (a) ResNet50 with ViT, (b) AlexNet with ViT, (c) VGG16 with ViT, (d) VGG19 with ViT, (e) Inception V3 with ViT.

TABLE 10. A comparison between the proposed model vs. other works.

Author(s), Ref. No.	Dataset	Algorithm (classification)	Accuracy	Precision	Recall	F1-Score
Tabbakh, A., Barpanda, S.S. [33]	PlantVillage	Machine learning model using LGBM model	94.39%	94.75%	94.76%	94.72%
Madana Mohana, R., et al. [61]	PlantVillage	CNN	96.77%	96.46%	96.25%	96.35%
Sachdeva, G., Singh, P. and Kaur, P. [62]	PlantVillage	Deep Convolutional neural network with Bayesian learning	98.9%	98.2%	97.9%	98.04%
Jasim, M.A. and Al-Tuwaijari, J.M. [63]	PlantVillage	CNN	98.029%	98.27%	98.01%	98.14%
Shijie J., et al. [64]	PlantVillage	VGG16	88.00%	-	-	-
Karthik, R., et al. [12]	PlantVillage	Attention and the residual network	95.83%	96.20%	95.60%	95.89%
Chen, J., et al. [65]	PlantVillage	DenseNet along with attention approach	97.94%	89.59%	86.71%	88.07%
Chen, J., et al. [66]	PlantVillage	pre-trained MobileNet-V2 and attention mechanism	96.68%	97.49%	95.83%	96.64%
Thakur, P.S., et al. [67]	PlantVillage	PlantViT	98.61%	98.24%	98.33%	98.28%
Suri, D., et al. [68]	Wheat	MobileNet	95.45%	94.56%	94.87%	94.72%
Lin, Z., et al. [69]	Wheat	Matrix based CNN	90.1%	90.15%	88.62%	85.5%
Kukreja, V. and Kumar, D. [70]	Wheat	DCNN	97.16%	-	-	-
Haider, W., et al. [71]	Wheat	Sequential CNN	97.20%	96.53%	96.43%	96.46%
Our proposed model	PlantVillage	TLMViT (VGG19 followed by ViT	98.81	98.72	98.76	98.73
Our proposed model	Wheat	TLMViT (VGG19 followed by ViT	99.86	99.78	99.65	99.71

best performance among all, obtaining the highest training accuracy equal to 92.78%, and the lowest training loss equal to 0.2064 values, as well as the highest validation accuracy equal to 98.48% and lowest validation loss equal to 0.1.

Furthermore, our proposed model is compared with the pre-trained based models. According to the results, the performance of the proposed model outperforms the same model without using ViT, due to extracting the deep features from the initial features by ViT. The initial features are extracted from the previous phase by the Transfer Learning-based model. In other words, ViT in the proposed model's topology is considered the second level of extracting features of leaf images. By analyzing the comparison above, the proposed model enhances performance compared with the Transfer learning-based model in terms of accuracy and loss for plant disease classification. Where enhancements percentages are (1.11%, 1.10%, 1.08%, 1.10%, and 1.14% validation accuracy) and (1.60%, 1.89%, 2.9%, 2.45%, and 4.00% validation loss) for ResNet50, AlexNet, Inception V3, VGG16, and VGG19 respectively. On average, the TLMViT model has 1.106% higher validation accuracy and 2.568% lower validation loss than pre-trained-based models.

Table 11 compares the epochs number and execution time of a model taken to reach a specific accuracy for both the pre-trained-based models and the proposed model on the PlantVillage dataset. The specific accuracy is chosen as per the highest accuracy of the pre-trained-based model. For example, ResNet50 took 15 epochs to achieve ∼80% accuracy in 56.90 minutes, whereas the proposed model using ResNet50 with ViT reached almost the same accuracy in 2 epochs in 7.79 minutes.

Similarly, the proposed approach is experimented on the Wheat Rust dataset. Table 7 presents the performance comparison for the training and validation set of the Wheat Rust dataset in terms of accuracy and loss between transfer learning-based models and TLMViT. Where the outcomes of training accuracy are equal to 93.26%, 95.48%, 99,93%, 99.67%, and 99.81% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 each of which is followed by ViT-based model respectively. Furthermore, the outcomes of validation accuracy are equal to 91.74%, 93.39%, 99.44%, 99.86%, and 99.69% for ResNet50, AlexNet, VGG16, VGG19, and Inception V3 each of which is followed by ViT-based model respectively. Moreover, Table 9 shows the results comparison between transfer learning-based models and TLMViT on the Wheat Rust dataset in terms of F1-score, precision, and recall. Table 12 compares the epochs number and execution time of a model taken to reach a specific accuracy for both the pre-trained-based models and the proposed model on the Wheat Rust dataset. The specific accuracy is chosen as per the highest accuracy of the pre-trained-based model. For example, ResNet50 took 21 epochs to achieve ∼84% accuracy in 13.15 minutes, whereas the proposed model using ResNet50 with ViT reached almost the same accuracy in 2 epochs in 1.39 minutes. According to the results of all models that are experimented on the PlantVillage and Wheat Rust datasets, the proposed model performs accurately on classification problems for both datasets using less number of Epochs and execution time.

In addition, Table 10 compares the performance of our model with other models using the main performance metrics (accuracy, precision, recall, and F1-score). In this

TABLE 11. Comparison of Epochs number and execution time for both Transfer Learning-based model and TLMViT model using PlantVillage dataset.

Model	Accuracy approx.	Pre-Trained based model		Proposed model	
		Epoch number	Time (Minutes)	Epoch number	Time (Minutes)
ResNet50	~80%	15	56.90	2	7.79
AlexNet	~84%	24	103.7	3	11.93
VGG16	~91%	22	74.24	3	11.91
VGG19	~89%	24	82.49	2	7.76
Inception V3	~93%	24	84.75	6	19.41

TABLE 12. Comparison of Epochs number and execution time for both Transfer Learning-based model and TLMViT model using Wheat Rust dataset.

Model	Accuracy approx.	Pre-Trained based model		Proposed model	
		Epoch number	Time (Minutes)	Epoch number	Time (Minutes)
ResNet50	~84%	21	13.15	2	1.39
AlexNet	~86%	23	18.91	3	2.46
VGG16	~93%	21	13.62	5	3.31
VGG19	~91%	21	13.96	4	2.86
Inception V3	~95%	22	20.68	7	6.15

comparison, the ImageNet dataset is used to fine-tune the parameters for our proposed model and all comparative works that used the transfer learning approaches. As shown in Table 10, the recent works that are done so far on plant disease classification concentrate on using four different scopes: machine learning models, CNN models, CNN along with attention approach, and CNN along with ViT. The plant disease-based machine learning classifiers need more steps in pre-processing stage to extract the leaf features, and the model performance would be less accurate due to the limitation of machine learning. In [33], a machine learning model using LGBM mode is proposed and trained on the PlantVillage dataset, the performance metrics did not exceed 94%. Though CNN models are effective in classifying plant disease images, these models represent only the connections between neighbouring pixels and are not able to encode the orientation and position of infected parts in the leaf. For this reason, their performance, i.e., accuracy, precision, recall, and F1-score are not very close to 1. In [61], [62], [63], and [64], CNN models are trained on the PlantVillage dataset, where the performance metrics did not reach 99%. Similarly, in [68], [69], [70], and [71] different types of CNNs are trained on the Wheat dataset where the performance metrics did not reach 99%. Combining the attention approach along with the CNN or pre-trained CNN model improves the accuracy because the attention unit is considered an adaptive filter, where its weights are set according to how well two pixels compose. In [65] and [66] DenseNet and pre-trained MobileNet-V2 along with attention mechanism are used respectively. In their proposed work, the performance metrics got improved and reached 98% and 97% approximately. Finally, the pre-trained CNN model along with ViT achieves the best performance in plant disease classification where the metrics are close to 1. In [67], a CNN model followed by ViT is trained on PlantVillage for plant disease detection, and the performance metrics reached 99% approximately. Whereas our proposed model uses pre-trained models followed by ViT is trained on PlantVillage and Wheat datasets. The performance metrics achieved accuracy equal to 98.81% and 99.86% for plantVillage and Wheat datasets respectively, which outperformed the previous models.

VIII. CONCLUSION

This paper proposes a hybrid model (TLMViT) for plant disease classification based on a pre-trained based model followed by a vision transformer. TLMViT has four stages: data acquisition, image augmentation, feature extraction, and classification. Firstly, three crops namely(bell peppers, potatoes, and tomato leaves) including fifteen classes of the PlantVillage dataset and three classes of the Wheat Rust dataset, are used individually to train and evaluate our proposed model. Secondly, the leaf images in the datasets are augmented by many functions such as rotation, shifting, shearing, zooming, and flipping to increase the number of training samples. Thirdly, the features are extracted in two consecutive phases: initial features extraction and deep features extraction. In the first phase, the pre-trained model is used to extract the features of the leaf and call them initial features. The weights of the pre-trained model are fine-tuned using the ImageNet dataset. In the second phase of features extraction, the vision transformer is utilized as deep layers to extract the deep features of leaves based on initial features. Lastly, the MLP head classifier determines to which class the leaf belongs. The proposed model's performance is experimented by five pre-trained models and evaluated in terms of accuracy, loss, precision, F1-score, and recall. The results show that the proposed model performs accurately in plant disease classification, getting the highest validation accuracy for VGG-19, followed by the ViT model. The validation accuracy is equal to 98.81% and 99.86%, and validation loss is equal to 0.0947 and 0.0715 for PlantVillage and wheat datasets respectively. Moreover, the results of TLMViT are compared with transfer Learning based models (without using ViT) to show the efficiency of using ViT for deep feature extraction. The TLMViT model outperforms the Transfer Learning-based model with an enhancement of 1.11% and 1.099% higher in validation accuracy and 2.576% and 2.92% lower in validation loss for PlantVillage and wheat datasets respectively. From all the above, the following findings of the proposed model can be noted: (1) The data augmentation approach provides a simple way to overcome the lack of images that leads to overfitting issues and reduce the influence of an imbalanced dataset. (2) A pre-trained-based model could be utilized to extract the initial leaf features and reduce the dimensionality of the original image, then be used as inputs for deep layers or a deep approach to extract the deep features. (3) ViT model shows the ability to extract the deep features from extracted features of the CNN model (initial features). On the other hand, the pre-trained model often fine-tunes for specific tasks. So, the proposed model could need more training and fine-tuning the weights in the first phase of feature extraction for particular crops. In future work, various crops could be considered for training on TLMViT to study the ability of the proposed model.

ACKNOWLEDGMENT

The authors would like to thank the High-Performance Computing Laboratory, VIT-AP University, for providing the resources and support that enabled this research to be conducted. The availability of advanced computing tools and skilled technical assistance played a crucial role in the successful completion of this article.

REFERENCES

[1] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.

[2] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. Ur Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: A review," *Comput. Electron. Agricult.*, vol. 153, pp. 12–32, Oct. 2018, doi: 10.1016/j.compag.2018.07.032.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[4] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosyst. Eng.*, vol. 144, pp. 52–60, Apr. 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[5] A. K. Pradhan, S. Swain, and J. K. Rout, "Role of machine learning and cloud-driven platform in IoT-based smart farming," in *Machine Learning and Internet of Things for Societal Issues*. Berlin, Germany: Springer, 2022, pp. 43–54.

[6] G. C. Index and C. Profiles, "Geneva: International telecommunication union," Tech. Rep., Dec. 2015.

[7] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," *Large-Scale Kernel Mach.*, vol. 34, no. 5, pp. 1–41, 2007, doi: 10.1038/nature14539.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Oct. 2015, doi: 10.1016/j.neunet.2014.09.003.

[9] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 3, pp. 243–257, Mar. 2021.

[10] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricult. Technol.*, vol. 3, Feb. 2023, Art. no. 100083.

[11] U. Mokhtar, "SVM-based detection of tomato leaves diseases," in *Intelligent Systems* (Advances in Intelligent Systems and Computing). Cham, Switzerland: Springer, 2015, pp. 641–652.

[12] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105933.

[13] T.-T. Tran, J.-W. Choi, T.-T. Le, and J.-W. Kim, "A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant," *Appl. Sci.*, vol. 9, no. 8, p. 1601, Apr. 2019.

[14] A. K. Dey, M. Sharma, and M. R. Meshram, "Image processing based leaf rot disease, detection of betel vine (Piper BetleL.)," *Proc. Comput. Sci.*, vol. 85, pp. 748–754, Jan. 2016.

[15] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers Plant Sci.*, vol. 8, p. 1852, Oct. 2017.

[16] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, J. Legg, and D. P. Hughes, "A mobile-based deep learning model for cassava disease diagnosis," *Frontiers Plant Sci.*, vol. 10, p. 272, Mar. 2019.

[17] O. O. Abayomi-Alli, R. Damasevicius, S. Misra, and R. Maskeliunas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," *Exp. Syst.*, vol. 38, no. 7, Nov. 2021, Art. no. e12746.

[18] H.-T. Thai, N.-Y. Tran-Van, and K.-H. Le, "Artificial cognition for early leaf disease detection using vision transformers," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2021, pp. 33–38.

[19] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, "Seasonal crops disease prediction and classification using deep convolutional encoder network," *Circuits, Syst., Signal Process.*, vol. 39, no. 2, pp. 818–836, Feb. 2020.

[20] A. J. Rozaqi and A. Sunyoto, "Identification of disease in potato leaves using convolutional neural network (CNN) algorithm," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Nov. 2020, pp. 72–76.

[21] A. Singh and H. Kaur, "Potato plant leaves disease detection and classification using machine learning methodologies," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1022, 2021, Art. no. 012121.

[22] J. Johnson, G. Sharma, S. Srinivasan, S. K. Masakapalli, S. Sharma, J. Sharma, and V. K. Dua, "Enhanced field-based detection of potato blight in complex backgrounds using deep learning," *Plant Phenomics*, vol. 2021, pp. 1–13, Jan. 2021.

[23] T.-Y. Lee, I.-A. Lin, J.-Y. Yu, J.-M. Yang, and Y.-C. Chang, "High efficiency disease detection for potato leaf with convolutional neural network," *Social Netw. Comput. Sci.*, vol. 2, no. 4, pp. 1–11, Jul. 2021.

[24] A. Alharbi, "AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery," *Sensors*, vol. 21, no. 11, p. 3830, 2021.

[25] B. Srinivas, "Prediction of guava plant diseases using deep learning," in *Proc. 3rd Int. Conf. Commun. Cyber Phys. Eng.* Cham, Switzerland: Springer, 2021, pp. 1495–1505.

[26] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agricult.*, vol. 4, pp. 41–49, Mar. 2017.

[27] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agricult.*, vol. 150, pp. 220–234, Jul. 2018.

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

[29] J. Kong, H. Wang, C. Yang, X. Jin, M. Zuo, and X. Zhang, "A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition," *Agriculture*, vol. 12, no. 4, p. 500, Mar. 2022.

[30] A. Pahurkar and R. Deshmukh, "PDMBM: Design of a high-efficiency plant disease classification method using multiparametric bio inspired modelling," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Apr. 2022, pp. 1607–1615.

[31] A. Bhargava, A. Bansal, and V. Goyal, "Machine learning–based detection and sorting of multiple vegetables and fruits," *Food Anal. Methods*, vol. 15, no. 1, pp. 228–242, Jan. 2022.

[32] A. Bhargava and A. Bansal, "Machine learning based quality evaluation of mono-colored apples," *Multimedia Tools Appl.*, vol. 79, nos. 31–32, pp. 22989–23006, Aug. 2020.

[33] A. Tabbakh and S. S. Barpanda, "Evaluation of machine learning models for plant disease classification using modified GLCM and wavelet based statistical features," *Traitement Du Signal*, vol. 39, no. 6, pp. 1893–1905, Dec. 2022, doi: 10.18280/ts.390602.

[34] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018.

[35] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, Jun. 2019.

[36] R. Chaturvedi, A. Sharma, A. Bhargava, J. Rajpurohit, and P. Gothwal, "Multi-level segmentation of fruits using modified firefly algorithm," *Food Anal. Methods*, vol. 56, pp. 2891–2900, Nov. 2022.

[37] N. E. M. Khalifa, M. H. N. Taha, L. M. Abou El-Maged, and A. E. Hassanien, "Artificial intelligence in potato leaf disease classification: A deep learning approach," in *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*. Berlin, Germany: Springer, 2021, pp. 63–79.

[38] A. Mittal and H. Gupta, "An experimental evaluation in plant disease identification based on activation-reconstruction generative adversarial network," in *Proc. 2nd Int. Conf. Advance Comput. Innov. Technol. Eng. (ICACITE)*, Apr. 2022, pp. 361–366.

[39] Q. Liang, S. Xiang, Y. Hu, G. Coppola, D. Zhang, and W. Sun, "PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network," *Comput. Electron. Agricult.*, vol. 157, pp. 518–529, Feb. 2019.

[40] R. Reedha, E. Dericquebourg, R. Canals, and A. Hafiane, "Transformer neural network for weed and crop classification of high resolution UAV images," *Remote Sens.*, vol. 14, no. 3, p. 592, Jan. 2022.

[41] S. Wu, Y. Sun, and H. Huang, "Multi-granularity feature extraction based on vision transformer for tomato leaf disease recognition," in *Proc. 3rd Int. Academic Exchange Conf. Sci. Technol. Innov. (IAECST)*, Dec. 2021, pp. 387–390.
[42] Y. Borhani, J. Khoramdel, and E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer," *Sci. Rep.*, vol. 12, no. 1, pp. 1–10, Jul. 2022.
[43] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 105–116.
[44] L. Windrim, A. Melkumyan, R. J. Murphy, A. Chlingaryan, and R. Ramakrishnan, "Pretraining for hyperspectral convolutional neural network classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2798–2810, May 2018.
[45] C. D. Gurkaynak and N. Arica, "A case study on transfer learning in convolutional neural networks," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.
[46] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *Proc. 2nd Int. Conf. Image, Vis. Comput. (ICIVC)*, 2017, pp. 783–787.
[47] C. Narvekar and M. Rao, "Flower classification using CNN and transfer learning in CNN—Agriculture perspective," in *Proc. 3rd Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2020, pp. 660–664.
[48] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3464–3473.
[49] F. Long, Z. Qiu, Y. Pan, T. Yao, J. Luo, and T. Mei, "Stand-alone inter-frame attention in video models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3192–3201.
[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[51] Kaggle. *The Dataset is Taken From the Kaggle Opensource Link*. Accessed: Aug. 2022. [Online]. Available: https://www.kaggle.com/datasets/emmarex/plantdisease
[52] D. Vasan, "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture," *Comput. Netw.*, vol. 171, Apr. 2020, Art. no. 107138.
[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.
[54] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, Sep. 2017.
[55] S. C. Bhardwaj, P. Prasad, O. P. Gangwar, H. Khan, and S. Kumar, "Wheat rust research-then and now," *Indian J. Agric. Sci.*, vol. 86, pp. 1231–1244, Oct. 2016.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[57] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, Jun. 2019.
[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[60] C. Feng, H. Zhang, S. Wang, Y. Li, H. Wang, and F. Yan, "Structural damage detection using deep convolutional neural network and transfer learning," *KSCE J. Civil Eng.*, vol. 23, no. 10, pp. 4493–4502, Oct. 2019.
[61] R. M. Mohana, C. K. K. Reddy, and P. R. Anisha, "A study and early identification of leaf diseases in plants using convolutional neural network," in *Proc. 4th Int. Conf. Smart Comput. Informat.* Cham, Switzerland: Springer, 2021, pp. 693–709.
[62] G. Sachdeva, P. Singh, and P. Kaur, "Plant leaf disease classification using deep Convolutional neural network with Bayesian learning," *Mater. Today, Proc.*, vol. 45, pp. 5584–5590, Jan. 2021.
[63] M. A. Jasim and J. M. Al-Tuwaijari, "Plant leaf diseases detection and classification using image processing and deep learning techniques," in *Proc. Int. Conf. Comput. Sci. Softw. Eng. (CSASE)*, Apr. 2020, pp. 259-265.
[64] J. Shijie, H. Siping, and L. Haibo, "Automatic detection of tomato disease and pests based on leaf images," in *Proc. Chin. Autom. Congr.*, 2017, pp. 2510–2537.
[65] J. Chen, W. Wang, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Attention embedded lightweight network for maize disease recognition," *Plant Pathol.*, vol. 70, no. 3, pp. 630–642, Apr. 2021.
[66] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Identification of Rice plant diseases using lightweight attention networks," *Exp. Syst. Appl.*, vol. 169, May 2021, Art. no. 114514.
[67] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Vision transformer for plant disease detection: *PlantViT*," in *Proc. 6th Int. Conf.* Cham, Switzerland: Springer, 2022, pp. 501–511.
[68] D. Suri, S. Saksenaa, U. Sehgal, and R. Garg, "Disease classification in wheat from images using CNN," in *Proc. 13th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2023, pp. 566–571.
[69] Z. Lin, "A unified matrix-based convolutional neural network for fine-grained image classification of wheat leaf diseases," *IEEE Access*, vol. 7, pp. 11570–11590, 2019.
[70] V. Kukreja and D. Kumar, "Automatic classification of wheat rust diseases using deep convolutional neural networks," in *Proc. 9th Int. Conf. Rel., INFOCOM Technol. Optim. (ICRITO)*, Sep. 2021, pp. 1–6.
[71] W. Haider, A.-U. Rehman, N. M. Durrani, and S. U. Rehman, "A generic approach for wheat disease classification and verification using expert opinion for knowledge-based decisions," *IEEE Access*, vol. 9, pp. 31104–31129, 2021.

AMER TABBAKH received the B.Tech. degree in informatics engineering from Aleppo University, Syria, in 2014, and the M.Tech. degree in computer science and engineering from KIIT University, Bhubaneswar, India, in 2020. He is currently a Ph.D. Scholar with the School of Computer Science and Engineering, VIT-AP University, Amaravati, India. His research interests include machine learning, deep learning, and image processing.

SOUBHAGYA SANKAR BARPANDA received the B.Tech. degree in computer science and engineering from the Biju Patnaik University of Technology, Rourkela, and the M.Tech. degree in software engineering and the Ph.D. degree from NIT, Rourkela. He is currently an Associate Professor with the School of Computer Science and Engineering, VIT-AP University, India. His research interests include image processing, software engineering, and biometrics.