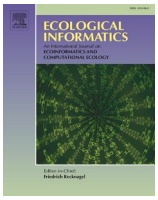
Ecological Informatics 77 (2023) 102245

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



Vision transformer meets convolutional neural network for plant disease classification

Poornima Singh Thakur *, Shubhangi Chaturvedi, Pritee Khanna, Tanuja Sheorey, Aparajita Ojha

PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482001, India

ARTICLE INFO

Keywords:
Plant disease detection
Convolutional neural network
Vision transformer
Deep learning
Grad-CAM
LIME

ABSTRACT

Plant diseases are the primary cause of crop losses globally, which have an impact on the world economy. To deal with these issues, new agriculture solutions are evolving that combine the Internet of Things and machine learning for early disease detection and control. Most of these systems use machine learning and computer vision for real-time disease identification and diagnosis. With advancements in deep learning techniques, various methods have emerged that employ convolutional neural networks for plant disease detection and identification. More recently, vision transformers have attracted the attention of researchers due to their strikingly better performance in classification problems in different vision-based applications. Accordingly, researchers have begun to explore vision transformers for plant pathology applications as well. In the present work, a hybrid model is proposed that combines the strength of a vision transformer with the inherent feature extraction capability of convolutional neural networks for disease identification using plant leaves. It can efficiently identify a large number of plant diseases for several crops. The proposed model has a lightweight structure with only 0.85 million trainable parameters, which makes it suitable for IoT-based agriculture systems. The performance of the proposed model is compared against nine state-of-the-art techniques on five publicly available datasets. The model is shown to outperform all nine methods even under challenging background conditions. On 'PlantVillage' dataset it achieves 98.86% accuracy and 98.9% precision, while on 'Embrapa', it shows 89.24% accuracy and 91.17% precision. On small datasets too its performance is better than other competing methods. Explainability of the proposed model is evaluated using gradient-weighted class activation maps and local interpretable model-agnostic explanations.

1. Introduction

The human population is likely to surpass ten billion in the next thirty years, and the food demand is expected to rise significantly in the coming decades (DESA, 2019). For sustainable growth of the agriculture industry, new technologies have emerged in recent years using the Internet of Things (IoT). A major challenge in agriculture is dealing with huge crop loss due to pests and diseases (FAO, 2021). Knowing well that chemical solutions have their own impact on the ecology, farmers use them in the absence of any large-scale viable solutions. Green agriculture and organic farming are rising, but these initiatives are only sporadic. To deal with plant pathogens and their adverse effect on the crop yield, early detection and containment of diseases are essential. In recent years, several smart agriculture solutions have been developed for early disease detection and control. These solutions incorporate *inter–alia*, vision-based machine learning (ML) techniques to identify diseases and suggest their treatment. Based on plant images, ML techniques detect and identify diseases in real-time. This helps farmers in controlling diseases. Researchers have proposed numerous solutions with different types of ML techniques. Some of the most extensively used ML techniques are support vector machine (SVM) (Chouhan et al., 2021; Hamdani et al., 2021; Hou et al., 2021; Kumar et al., 2020; Sun et al., 2019; Zhang and Wang, 2016), artificial neural networks (ANN) (Hamdani et al., 2021; Ramesh and Vydeki, 2020), Naive Bayes (Abdu et al., 2020; Hamdani et al., 2021; Johannes et al., 2017), K-means clustering (Johannes et al., 2017; Ramesh and Vydeki, 2020), and simple linear iterative clustering (Hou et al., 2021; Johannes et al., 2017; Sun et al., 2019).

* Corresponding author.
E-mail addresses: poornima@iiitdmj.ac.in (P.S. Thakur), shubhangi@iiitdmj.ac.in (S. Chaturvedi), pkhanna@iiitdmj.ac.in (P. Khanna), tanush@iiitdmj.ac.in (T. Sheorey), aojha@iiitdmj.ac.in (A. Ojha).

https://doi.org/10.1016/j.ecoinf.2023.102245
Received 8 December 2022; Received in revised form 30 July 2023; Accepted 30 July 2023
Available online 1 August 2023

In recent years, deep learning (DL) approaches have been utilized for plant disease identification due to the availability of large datasets and powerful computing resources. The automatic feature learning capability of convolutional neural network (CNN) architectures has produced promising results in plant disease identification. Initial works focused on comparing the standard CNN architectures to find out a suitable CNN model for plant disease detection. The pioneering work in this direction were done by Mohanty et al. (2016) and Barbedo (2018). They used well known networks such as AlexNet, GoogleNet, VGG16, and ResNet and applied transfer learning approach in their studies. In another such study, Atila et al. (2021) compared the performance of some of the state-of-the-art architectures and demonstrated the efficacy of EfficientNet B4 and B5 over other networks. Sutaji and Yldz (2022) further experimented with an ensemble of MobileNet v2 and Xception models to build the plant disease detection model using the PlantVillage dataset.

Apart from standard CNN architectures custom CNN architectures were also introduced for plant disease detection tasks (Huang et al., 2020; Yadav et al., 2021). Gokulnath et al. (2021) developed a CNN model with just three convolutional layers for disease detection and identification in tomato and potato species. Keceli et al. (2022) developed a multi-task learning model where a custom CNN model and a pretrained AlexNet model were concatenated for the prediction of plant species and diseases. They used the images of tomato, potato, pepper, and maize species from the PlantVillage dataset to build their model. They also used rice and maize datasets for the performance validation of their model.

There has been a rising interest in deploying attention based CNN models, and these models have exhibited excellent performance in plant disease detection (Karthik et al., 2020; Chen et al., 2021d, 2020c, 2021c, b; Zhao et al., 2022; Thakur et al., 2022a). Pandey and Jain (2022) have recently proposed a CNN model based on attention-dense learning blocks. Li et al. (2023) have suggested a combination of multi-dilated convolution and block attention modules with the DenseNet architecture for the classification of diseases. They have also applied an auxiliary classifier generative adversarial network (ACGAN) for the data augmentation task.

The concept of vision transformer (ViT) (Vaswani et al., 2017) has opened new vistas in computer vision (Dosovitskiy et al., 2020), (Vaswani et al., 2017). ViT has demonstrated exceptional classification performance on benchmark datasets like ImageNet, CIFAR-10, CIFAR-100, Oxford-IIIT Pets, Oxford Flowers-102, and VTAB. Inspired by its performance, researchers have also explored ViT for building plant disease detection models (see for example, (Borhani et al., 2022; Li et al., 2022; Thai et al., 2021; Thakur et al., 2022c)). Although the inherent characteristics of ViT in extracting long-term feature dependencies make it very powerful, it has certain limitations in terms of capturing local features in images (Neyshabur, 2020). Further, ViT models do not converge well on small datasets as compared to CNN models. CNNs have strong inductive biases such as locality and equivariance that make them apt for image feature learning. Therefore, an amalgamation of CNN with the self-attention modules of ViT may help in simultaneously extracting local and global features. Further, ViT combined with CNN may help in enhancing the explainability of plant disease identification models. In recent works, researchers have experimented with hybrids of CNN and ViT for plant disease detection (see for example, (Borhani et al., 2022; Li et al., 2022; Thakur et al., 2022c)). However, most of the models perform well only when they are trained and tested on large datasets. Further, the models are either computationally heavy with high memory demand or do not perform well across different datasets. This limits their applicability for IoT based smart agriculture solutions.

Existing studies indicate that the plant disease detection remains a challenge due to a large number of disease species and a variety of different crops. High similarity in disease symptoms and evolving disease patterns add to the complexity of the problem. While DL models with large number of parameters are data-hungry, lightweight CNN models do not generalize well. Unavailability of in-field data for training DL models is another challenge that hinders the generalizability of these models (cf. (Chouhan et al., 2020) Thakur et al. (2022b). Apart from this, the current plant disease detection methods have not been investigated for the interpretability of their results.

In an effort to provide a solution for a large number of crop varieties and disease species, a plant disease detection and classification model is proposed in this paper. The proposed model is lightweight and provides predictions that are better explainable than the existing methods. Highlights of the present work are as follows.

- A hybrid plant disease detection model with convolutional neural network and vision transformer is proposed with only 0.85 million trainable parameters.
- The proposed model outperforms nine recent deep learning models on five public datasets.
- The model shows improved explainability using the gradient-weighted class activation maps and local interpretable model-agnostic explanations.
- The model is lightweight and is suitable for smart agriculture applications.

Details of the remaining sections are as follows. Related works on crop disease identification are discussed in Section 2. The proposed model is presented in Section 3 with materials and methods used in model building. Results of experiments and research findings are discussed in Section 4. Section 5 provides concluding remarks and the future scope of the work.

2. Related works

This section presents some recent works on plant disease detection that have used CNN and ViT architecture and are related to the proposed work. The section is organized into three parts: the first part contains an overview of some of the recent CNN models. The second part focuses on attention mechanisms based CNNs, while the third part discusses the works utilizing ViT for plant disease detection.

2.1. Convolutional neural network based methods

The initial works on plant disease detection using CNNs were focused on building datasets for the research community and using the transfer learning approaches. In a pioneering study, Mohanty et al. (2016) collected a large-scale dataset 'PlantVillage' with 38 disease classes and compared the performance of GoogleNet and AlexNet models. He reported that GoogleNet model attained an impressive accuracy of 99.35% when used with transfer learning approach. In another study, Barbedo (2018) adopted a transfer learning approach with the GoogleNet as the base model to examine the effect of dataset size and sample variations in identifying plant diseases. He collected 1383 images from 56 different disease classes and removed their background, and experimented with the CNN classifiers. His study concluded that the limitations of the datasets in the area were the main bottleneck in the practical deployment of CNN models. Too et al. (2019) also analyzed the performance of DenseNet121, Inception v4, ResNet50, ResNet101, ResNet152, and VGG16 on the same dataset using a fine tuning technique. In their experiments, DenseNet121 model outperformed all the other models with an accuracy of 99.75%.

Researchers also experimented with custom CNN models using convolution blocks of well known CNN architectures like VGG16, and Inception Nets. Chen et al. (2020a) combined the first two blocks of the pretrained VGG19 and two Inception v3 blocks to create a custom CNN model for plant disease identification. They also collected images to create their own maize dataset. The model was evaluated on three different datasets including PlantVillage dataset and another Maize dataset. Thakur et al. (2022d) also developed a CNN model combining

the initial two convolution layers of the pretrained VGG16 and two Inception v7 blocks. They evaluated the performance of their model on five publicly available datasets and reported 99.16%, 93.66%, 94.24%, 91.36%, and 96.67% accuracy scores on the PlantVillage, Embrapa, Apple, Maize and Rice datasets, respectively. These explorations on the use of in-field datasets were beneficial in the development of methods for plant disease detection that could be used in the real-world applications.

Table 1 lists some of the important plant disease detection techniques using CNN architectures.

2.2. Methods using convolutional neural networks with attention mechanism

Due to widespread applications of attention-based CNN, these have also been explored in plant disease detection and classification. The attention mechanism works by associating a high weightage to pixel locations with relevant information, that helps in fetching salient features for each type of disease. The techniques listed in Table 2 use different types of attention-based CNN architectures for building plant disease detection models. Karthik et al. (2020) have deployed a residual CNN with an attention mechanism to identify tomato leaf diseases. Their model is shown to achieve 98% accuracy on a dataset that contains 95,999 tomato leaf images classified into 10 categories. The dataset is obtained by collecting images from PlantVillage dataset and applying data augmentation. Zeng and Li (2020) have also experimented with a residual CNN equipped with self-attention. On the dataset MK-D2, their model has demonstrated 98% accuracy and on another dataset AES-CD9214 95.33% accuracy is attained. Chen et al. (2021d) have developed a CNN model that uses DenseNet with depth-wise separable convolution alongwith spatial and channel-wise attention modules. Their model works well for the maize data taken from the PlantVillage dataset attaining 98.50% accuracy. The authors have also collected their own maize dataset and have claimed that their model achieves 95.86% accuracy. In another work by Chen et al. (2021c), spatial and channel-wise attentions are applied with MobileNet to develop a rice leaf disease detection model with 98.48% accuracy.

Chen et al. (2021a) have further utilized the squeeze-and-excitation (SE) attention block with MobileNet v2 to achieve 99.33% accuracy on their rice dataset. MobileNet v2 architecture has also been used with depth-wise separable convolution, spatial attention, and channel-wise attention that shows exceptional performance with 99.71% accuracy on a small number of samples from PlantVillage dataset (Chen et al., 2021b). Chen et al. (2021b) have continued to develop another attention-based plant disease detection model for the paddy, corn, and cucumber plants, their model achieves an average accuracy of 99.13%. Zhao et al. (2022) has developed an attention based CNN model with Inception and residual blocks that demonstrates 99.55% average accuracy on potato, corn, and tomato plant diseases. These studies conclude that embedding attention in the well known architectures like MobileNet, Inception or residual blocks helps in improving the models' performance. But these models have been tested only on a small number of plant diseases in contrast to previous models that were trained and tested on a large variety of crops.

2.3. Vision transformer based methods

With the grand success of ViT models in general classification problems, researchers have also investigated their performance in the identification of plant diseases. Table 3 lists some of the recently introduced ViT based models. Thai et al. (2021) have used the original ViT architecture with fine tuning for cassava disease identification. They have reported that their ViT model achieves an F1 score of 90.3%. They have also deployed their model on an IoT device powered by a Raspberry Pi board. However, the model contains 85.79 million parameters that makes it not so suitable for such applications. Some recent approaches have used a combination of convolution and transformer blocks for developing effective plant disease detection models. Li et al. (2022) have proposed a model with an initial convolution layer followed by 12 transformer blocks for plant disease detection. They have used non-overlapping image patches of size 16×16 to pass through the network and have reported an accuracy of 96.71% on 39 categories from the PlantVillage dataset. Borhani et al. (2022) have also designed a lightweight model using CNN and transformer blocks. Their model has been tested on the Wheat, Rice, and PlantVillage datasets. While on the Wheat dataset, their model is shown to achieve 100%, F1 score, on Rice, and PlantVillage datasets it has exhibited only 91.7%, and 97.83% F1 scores respectively. It may be mentioned that the small number of samples in the wheat dataset could be the reason for overfitting, as it is known that ViT does not perform very well on small datasets (Vaswani et al., 2017).

Thakur et al. (2022c) have developed a CNN + ViT based plant disease detection model that works well on a large number of crops and their diseases. They have trained and tested their model's performance on two public datasets with 98.61% and 87.87% accuracy scores on PlantVillage and Embrapa respectively. Lu et al. (2022) have deployed GhostNet with ViT blocks to detect diseases on vine leaf images from the GLDP12k dataset, and their model has achieved 98.14% accuracy. Li and Li (2022) have focused on developing a lightweight model involving CNN and ViT blocks for apple disease detection. They have introduced a four-stage model, and in each stage, different number of convolution and transformer layers are used. To reduce the number of trainable parameters and floating point operations per second (FLOPS), they have used depth-wise separable convolution. Their model achieves 96.85% accuracy on an Apple dataset with 9.5 million parameters and 0.98 GigaFLOPS (GFLOPS).

A review of the aforementioned studies indicates that the combination of CNN and ViT can be useful in developing plant disease detection models, but more needs to be explored to develop models that can work uniformly well on a large number of crop species and disease types.

Table 1
An overview of related works on plant disease identification using convolutional neural network.

Work	Approach	Dataset	Accuracy	Precision	Recall	F1 score
Mohanty et al. (2016)	GoogleNet and AlexNet	PlantVillage dataset with 54,305 leaf images, 38 categories	–	–	–	–
	GoogleNet		99.35	99.35	99.35	99.34
Barbedo (2018)	GoogleNet	1383 images, 56 categories	87	–	–	–
Too et al. (2019)	ResNet with 50, 101 and 152 layers, VGG16, DenseNet121 and Inception v4	PlantVillage dataset with 54,305 leaf images, 38 categories	–	–	–	–
	DenseNet121		99.75	–	–	–
Chen et al. (2020a)	Inception v3, VGG19	466 images of maize leaves, 4categories	80.38	–	60.76	–
		500 rice leaf images, 5 classes	92	–	80	–
Thakur et al. (2022d)	VGG16, Inception v7	PlantVillage dataset with 54,305 leaf images, 38 categories	99.16	–	–	–
		560 leaf images of rice, 4 categories	96.67	–	–	–

Table 2
An overview of related works on plant disease identification using attention-based convolutional neural network.

Work	Approach	Dataset	Accuracy	Precision	Recall	F1 score
Karthik et al. (2020)	Attention mechanism with residual CNN	PlantVillage dataset with 95,999 tomato leaf images, 10 categories	98	–	–	–
Zeng and Li (2020)	CNN with self-Attention mechanism	AES-CD9214 dataset with 9214 leaf images in 6 categories	95.33	–	–	–
		MK-D2 dataset with 988 leaf images	98	–	–	–
Chen et al. (2021d)	Mobile-DANet, DenseNet, transition layer, depthwise separable convolution, attention module	PlantVillage dataset with 3852 leaf images of maize, 4 categories	98.5	97	97	97
		133 leaf images of maize, 8 categories	95.86	83.45	83.45	83.45
Chen et al. (2021c)	MobileNet-V2, attention mechanism	PlantVillage dataset with 1045 leaf images, 10 categories	99.67	98.37	98.37	99.81
		1107 leaf images of rice, 12 categories	98.48	90.56	90.56	90.56
Chen et al. (2021a)	SE attention with MobileNet	PlantVillage dataset with 1645 images in 11 categories	99.78	–	98.83	–
		Rice dataset with 444 images, 25 categories	99.33	–	87.87	–
Chen et al. (2021b)	MobileNet v2 with Depthwise separable convolution, spatial attention and channelwise attention module	PlantVillage dataset with 1045 leaf images, 10 categories	99.71	–	98.56	98.56
		405 leaf images in 20 categories	99.13	–	91.37	91.37
Zhao et al. (2022)	Residual and Inception CNN with channelwise attention module	PlantVillage dataset with 38,466 images of corn, tomato, and potato, 17 categories	99.55	–	–	–

Table 3
An overview of related works on plant disease identification using vision transformer.

Work	Approach	Dataset	Accuracy	Precision	Recall	F1 score
Thai et al. (2021)	ViT	21,397 images of cassava leaf in five classes	–	–	–	90.3
Lu et al. (2022)	GhostNet, ViT	GLDP12k dataset with 12,615 vine leaf images in 11 categories	98.14	–	–	–
Li and Li (2022)	Depthwise separable CNN, ViT	15,834 apple images in 5 categories	–	95.21	95.19	95.19
Borhani et al. (2022)	CNN, ViT	3679 images of wheat rust in 3 categories	–	100	100	100
		120 rice leaf images, 3 categories	–	92	91.7	91.7
		PlantVillage dataset with 54,306 leaf images, 38 categories	–	97.88	97.83	97.83
Li et al. (2022)	CNN, ViT	PlantVillage dataset with 61,486 images in 39 categories	96.71%	–	–	–
Thakur et al. (2022c)	CNN, ViT	PlantVillage dataset with 54,305 images in 38 categories	98.61	98.24	98.33	98.28
		Embrapa dataset with 46,376 images in 93 categories	87.87	84.7	80.75	82.52

Since the plant diseases are often quite similar in terms of texture and colors, it is important to analyse the models' interpretation of different plant diseases. To the best of our knowledge, despite the high accuracy achieved by several recent models, their explainability has not been analyzed. With advances in explainable artificial intelligence, it is crucial to develop methods that not only perform well on a variety of plant diseases but also aid scientists in analyzing the reasons for high accuracy and possible reasons for their failures.

In the present paper, a ViT-enabled CNN model is proposed that exploits the locality of CNNs with the global attention of ViT blocks. The proposed model significantly improves the disease classification performance for a number of plant diseases and outperforms nine recently introduced DL models on five publicly available datasets. Furthermore, prediction results are found to have improved explainability as compared to those of existing models. Details of the model are presented in the next section.

3. Materials and methods

This section discusses the proposed model for detection and identification of plant diseases. The model consists of two initial blocks of the pretrained VGG16 network, followed by an Inception v7 block and a stack of four transformer encoder modules. For the sake of completeness, we begin with a brief introduction to the ViT.

3.1. Vision transformer

With the huge success of transformer networks in natural language processing problems (Vaswani et al., 2017), Dosovitskiy et al. (2020) developed the ViT model based on the architecture of the original transformer. The ViT is composed of self-attention blocks and multilayer perceptron (MLP) networks with a linear projection and positional embedding mechanism. The organization of a typical ViT is presented in Fig. 1. To feed an image to the ViT, it is first split into fixed-size non-overlapping patches. These patches are flattened and then transformed into lower-dimensional representations. Each flattened patch is subjected to a learnable linear transformation to generate the corresponding linear projection and positional embedding. For details of these operations, reference may be made to Dosovitskiy et al. (2020). The output vector obtained from the linear projection and embedding is passed onto a stack of N number of transformer blocks. A transformer block consists of multi-head self-attention (MHA) and MLP. Each one has a normalization layer before it and a residual connection after it. MHA is a self-attention mechanism that is applied on each patch separately. In MHA, the input vector X is transformed into three separate vectors $Q = XW_Q$, $K = XW_K$, and $V = XW_V$; where W_Q, W_K, and W_V are the weight matrices. The score matrix is generated after performing the dot-product between Q and K. Then, the output vector is subjected to softmax activation as given in Eq. 1.

$$SA(Q,K,V) = SoftMax\left(\frac{QK^T}{\sqrt{d}}\right)*V \tag{1}$$

It may be noted that the dot products are normalized by the square root of the dimension d of the vector K. This is done to avoid the issue of dot products becoming excessively large during training as mentioned in the original work on transformer networks (Vaswani et al., 2017). Self-attention matrices are finally combined and sent to a linear layer, which processes the input and passes onto the regression head. The self-attention enhances the semantic similarity of features at different image locations for classification. The number of MHA in a transformer encoder is a hyperparameter that can be tuned based on the application data. Output of the MHA block is given by Eq. 2. After the MHA block,

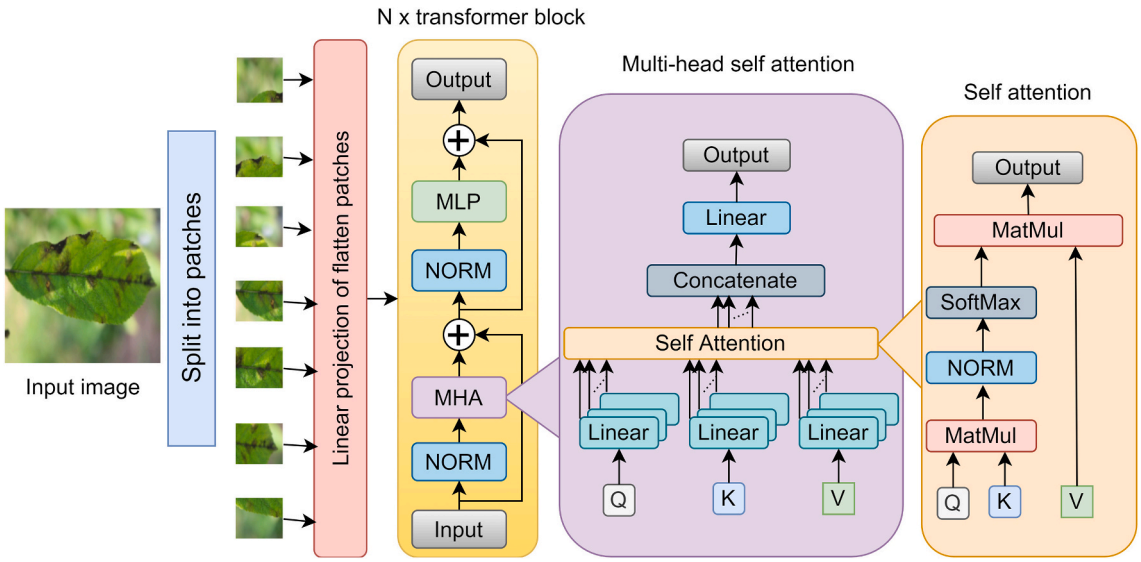


Fig. 1. ViT block with multi-head self-attention units.

the data is passed through the MLP with a GELU activation function. The GELU activation is obtained as the multiplication of the input by its Bernoulli distribution. Each transformer block has a skip connection between the input and the output of MHA as well as that of MLP (Fig. 1). The output of the transformer block is given in Eq. 3.

$$MHA_{out} = MHA(NORM(X_{in})) + X_{in} \tag{2}$$

where X_{in} is the input to transformer block, *NORM* is the normalization layer, *MHA* is multi-head self-attention, and MHA_{out} is the output of multi-head self attention layer.

$$TF_{out} = MLP(NORM(MHA_{out})) + MHA_{out} \tag{3}$$

where *MLP* is the multi layer perceptron block, and TF_{out} is the output of the transformer block.

3.2. Hybrid CNN-ViT model for plant disease identification

The main objective of this work is to create a hybrid model for plant disease identification that combines the capabilities of ViT (Dosovitskiy et al., 2020) and CNN for plant disease detection and identification. Convolution (Conv) blocks are used to efficiently extract local-level features, while the transformer blocks are appended for global feature extraction. The pipeline of the proposed model is presented in Fig. 2. The main components of the proposed model are Conv blocks of VGG16 and Inception v7, and ViT components- MHA, MLP with linear projections.

An input of size 224 × 224 × 3 is required by the proposed model, as

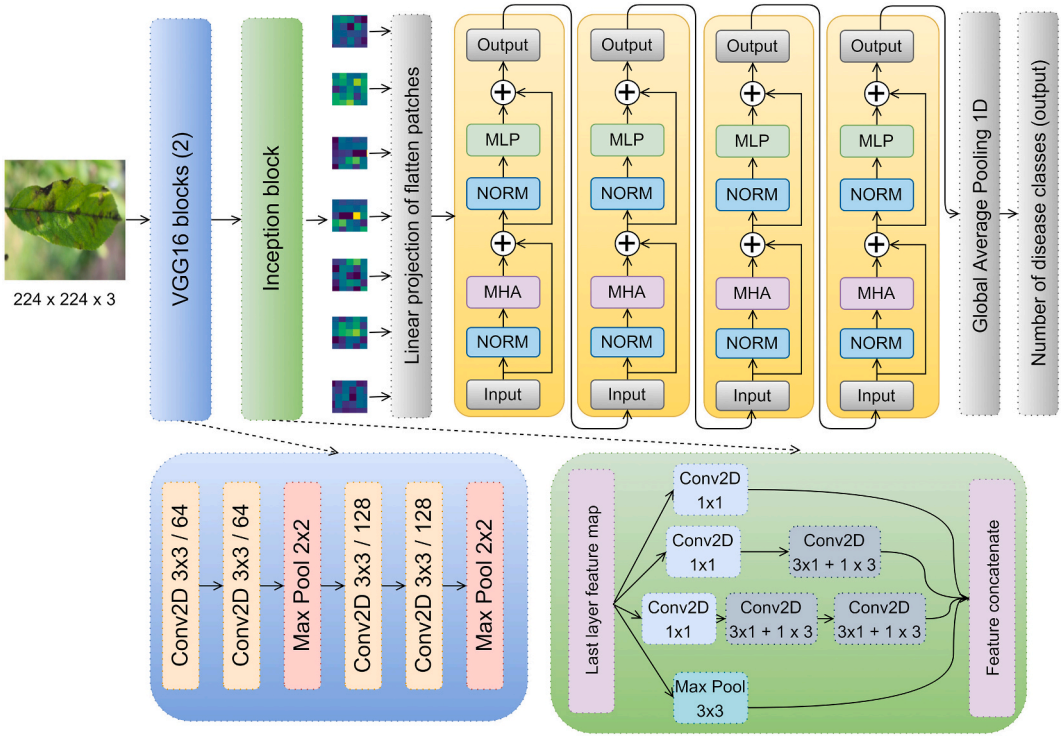


Fig. 2. Block diagram of the proposed plant disease classification model.

shown in Fig. 2. The proposed model comprises two Conv blocks of the pre-trained VGG16 network on the Imagenet dataset. Each block consists of two Conv layers followed by a max pooling layer. The output of second Conv block is $56 \times 56 \times 128$. This output is fed to a multi-level feature extraction block similar to Inception v7 Conv blocks, as shown in Fig. 2. The multi-level feature extraction block is added to enhance the local feature learnability of the model. The Inception block generates an output of size $56 \times 56 \times 512$ after concatenating the feature maps generated by different Conv layers.

The feature map is then converted into patches, each of size 5×5. The flattened patches are then passed through linear projection and generate feature vector of size 121×16. These vectors are fed to a stack of four transformer blocks for feature extraction. The output of the transformer block is then transformed into a 1-dimensional vector by adding the global average pooling layer. Finally, a fully connected layer with softmax activation is created, with the number of neurons matching the number of classes in the dataset. For a dataset with 4 class labels, the model contains $850,500$ total trainable parameters. The model is trained, validated, and tested on a variety of datasets. The experimental results on all the datasets are presented in Section 4. The proposed model is enriched by pre-trained VGG16 that helps with better parameter initialization and an Inception v7 block that generates a rich pool of multi-scale features. The transformer blocks with MHA provide an efficient mechanism for image patch processing and help extract saliency in the patches. The fusion of CNN and the transformer network makes a powerful feature extractor with a combination of global and local information and improves the model's explainability. Five open datasets are used to test the model, and the results are shown in the next section.

The proposed plant disease detection model was developed using five publicly available datasets. Comprehensive experiments were carried out to finalize the model's architecture and evaluate its performance with respect to existing models. Some of the recent lightweight CNN models that have used different attention mechanisms and ViT models were considered for performance comparison. More specifically, the model's performance was compared with five CNN models (Chen et al., 2021d, 2021b, 2021c; Karthik et al., 2020; Zhao et al., 2022) and four recently introduced ViT based models (Borhani et al., 2022; Li et al., 2022; Thai et al., 2021; Thakur et al., 2022c). Recall from Section 2, that the model given in (Karthik et al., 2020) has been developed using a residual attention-based CNN, while the models developed in (Chen et al., 2021d, 2021b, 2021c) have combined the attention mechanism with the base architectures MobileNet and DenseNet. Another model used for the comparative study is taken from (Zhao et al., 2022) where residual and inception blocks combined with spatial and channel attention are used to build the plant disease detection model. The fine-tuned baseline ViT model developed by Thai et al. (2021) is chosen because it is the first model developed by fine-tuning the original (baseline) ViT. The other three ViT models given in (Borhani et al., 2022), (Li et al., 2022) and (Thakur et al., 2022c) have also used different combinations of CNN and ViT, similar to the proposed model.

Initially, the images in each dataset were resized to $224 \times 224 \times 3$, the proposed model was trained using training subsets from the datasets. For training the model, categorical cross-entropy loss with the Adam optimizer was used. The cross-entropy loss is defined in Eq. 4. The batch size was 16 with the learning rate as 0.0001. The performance of the model was assessed after each epoch with the help of validation dataset. The model was evaluated on the test dataset after achieving the necessary level of classification accuracy on the training and validation subsets.

$$Loss = -\frac{1}{n}\sum_{i=1}^{n} y_i \ log\widehat{y}_i \tag{4}$$

Here the loss for all the n samples in a batch was calculated using y_i as the actual label and $\widehat{y}_i$ as the predicted value of the i-th sample. Using y_i as the actual label and $\widehat{y}_i$ as the predicted value of the i-th sample, the loss for all the n samples in a batch was determined.

In the following sections, details of the datasets, evaluation metrics, experimental setup, and ablation study are presented.

3.3. Datasets

Five openly accessible datasets from various geographic origins and a variety of crops are used in the studies. The datasets are chosen from a variety of groups, ranging from small, balanced datasets with 400–500 images to huge, unbalanced datasets with more than 40 K images. These datasets were collected from different contexts with the intention of training the proposed approach for a variety of crops and their illnesses and assessing its performance in various test scenarios. Out of the five publicly available datasets, the PlantVillage dataset (Hughes et al., 2015) contains data in 38 categories with 54,305 images, and the Embrapa dataset (Barbedo et al., 2018) contains 46,376 images in 93 classes. These datasets contain multiple species with a variety of diseases affecting the plant's leaves. On the other hand, the Apple (Thapa et al., 2020), Maize (Chen et al., 2020a), and Rice (Chen et al., 2020b) datasets are single-species datasets selected for the experiments. The details of each dataset, including the number of classes, major diseases covered, and the number of images, are presented in Table 4. Fig. 3 displays examples of images from each of the datasets.

3.4. Evaluation metrics

Each model for comparison is evaluated using standard classification measures. These include accuracy, precision, recall, F1 score, area under the receiver operating curve, and Cohen's kappa score.

Accuracy: A common performance indicator for the image classification task is accuracy. It explains how the actual class value and the predicted class value relate to one another. The algorithm performs better the greater the accuracy value it can get. In Eq. 5, accuracy is described as follows.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5}$$

Here TP, TN, FP, FN have the usual meanings of True positive, true negative, false positive and false negative respectively.

Precision: It is the ratio of the number of TP lables and all the positive predicted labels. It falls somewhere between 0 and 1.

$$Precision = \frac{TP}{(TP + FP)} \tag{6}$$

Recall: It is specified as the proportion of all actual positive labels to predicted positive labels. It is calculated as shown in Eq. 7.

$$Recall = \frac{TP}{(TP + FN)} \tag{7}$$

F1 Score: It represents the harmonic mean between the recall and the precision. The formula for F1 score is depicted in Eq. 8.

$$F1score = 2^{*}\frac{(Precision^{*}Recall)}{(Precision + Recall)} \tag{8}$$

Area Under the Curve (AUC): AUC is the area covered by the receiver characteristic operator (ROC) curve. The ROC is calculated using the plot of true positive rate (TPR) (refer Eq. 9) and false positive rate (FPR) (refer Eq. 10).

$$TPR = \frac{TP}{(TP + FN)} \tag{9}$$

$$FPR = \frac{FP}{(FP + TN)} \tag{10}$$

Table 4
Datasets used in the experiments.

Dataset	# of classes	# of images	Training images	Validation images	Test images
Apple (Thapa et al., 2020)	4 (Healthy, rust, scab, multiple diseases)	1821	1165	291	365
Embrapa (Barbedo et al., 2018)	93 (Cotton, Grapevine, Rice, Coffee, Cashew, Sugarcane, Wheat etc. in leaf spot, bacterial canker, rust, mildew, scald, blight, mosaic virus etc.)	46,376	29,681	7420	9275
Maize (Chen et al., 2020a)	4 (Eyespot, Goss's Bacterial wilt, Gray Leaf Spot, Phaeosphaeria Spot)	481	280	120	81
PlantVillage (Hughes et al., 2015)	38 (Apple, cherry, corn etc. in scab, black rot, rust, healthy, powdery mildew, gray leaf spot, rust etc.)	54,305	34,755	8688	10,862
Rice (Chen et al., 2020a)	5 (Bacterial leaf streak, leaf scald, leaf smut, stackburn, white tip)	560	350	150	60

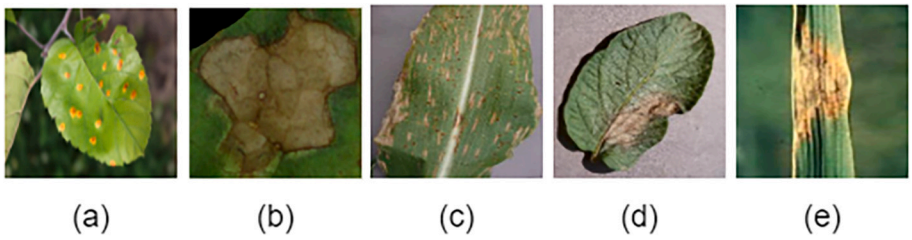


Fig. 3. Sample images from the datasets: (a) rust (Apple), (b) cotton soreshin (Embrapa) (c) northern leaf blight (Maize) (d) potato late blight (PlantVillage) (e) leaf scald (Rice).

Cohen's Kappa Score: Cohen's kappa coefficient or score is a probability based measure where the outcome is the level of agreement between parties for the classification problem. The formula of kappa score calculation is shown below in Eq. 11.

$$Kappa = \frac{p_o - p_e}{(1 - p_e)} \tag{11}$$

Where p_o is the relative agreement probability and p_e is the hypothetical agreement probability between the parties.

In addition, confusion matrices and t-distributed stochastic neighbor embedding (t-SNE) plots are used for evaluating the performance of the model. The confusion matrix indicates the credibility of the model. The t-SNE plots demonstrated how good the model is in terms of distinctive feature extraction for different categories in a dataset (Van der Maaten and Hinton, 2008). Further, the model's explainability is evaluated using two standard methods, namely Grad-CAMs and LIME.

Table 5
The proposed model's performance with different patch sizes.

Patch size	Loss	Accuracy	Precision	Recall	F1 score	AUC	kappa score
Apple							
1	0.31	92.47	92.47	92.47	92.47	**97.61**	0.89
3	0.58	81.72	81.52	80.65	81.08	95.21	0.73
5	**0.3**	**93.55**	**93.55**	**93.55**	**93.55**	97.01	**0.91**
7	0.44	89.25	89.73	89.25	89.49	96.91	0.84
9	0.43	88.17	88.11	87.63	87.87	96.83	0.83
Embrapa							
1	0.53	87.25	89.59	85.85	87.68	98.52	0.87
3	0.54	86.61	88.64	85.11	86.84	98.44	0.86
5	**0.46**	**89.24**	**91.17**	**88.27**	**89.7**	**98.73**	**0.89**
7	0.52	87.63	89.49	86.68	88.06	98.25	0.87
9	0.68	84.73	86.81	83.5	85.12	97.72	0.84
Maize							
1	0.42	86.42	88.61	86.42	87.5	96.74	0.82
3	0.44	90.12	91.03	87.65	89.3	95.48	0.87
5	**0.35**	**92.59**	**92.5**	**91.36**	**91.93**	**96.8**	**0.9**
7	0.44	88.89	92.31	88.89	90.57	95.43	0.85
9	0.42	87.65	88.75	87.65	88.2	96.76	0.84
PlantVillage							
1	0.11	97.06	97.38	96.86	97.12	99.76	0.97
3	0.05	98.66	98.71	98.62	98.66	99.87	**0.99**
5	**0.04**	**98.86**	**98.9**	**98.81**	**98.85**	**99.92**	**0.99**
7	0.09	97.3	97.58	97.08	97.33	99.82	0.97
9	0.08	97.85	98.01	97.82	97.91	99.78	0.98
Rice							
1	0.25	93.33	94.92	**93.33**	94.12	99.41	0.91
3	0.27	91.67	94.74	90	92.31	99.38	0.89
5	0.24	**95**	**98.25**	**93.33**	**95.73**	99.39	**0.94**
7	**0.21**	93.33	94.83	91.67	93.22	**99.73**	0.91
9	**0.21**	95	94.92	**93.33**	94.12	99.51	**0.94**

3.5. Experimental setup

All the experiments were performed on an Nvidia DGX A100 160 GB station with four GPU A100 cards with 40 GB of memory in each. It runs Ubuntu 18.04 LTS and has an AMD 7742 CPU clocked at 2.25 to 3.4 GHz and 512 GB of RAM. The proposed model and other selected models for comparison were implemented using the Keras framework with NVIDIA CUDA v11.5 and the cuDNN v8.3 library.

3.6. Model building and ablation study

For building the model, several hyperparameters were taken into consideration. This included tuning the learning rate, optimizer, patch size for the transformer and number of MHAs. An ablation study was also performed to finalize the model architecture. First the learning rate of 0.0001 was determined through empirical experiments. Simultaneously, the most suitable optimizer was selected. Then the number of MHA layers was also fine-tuned by repeated experiments on the PlantVillage dataset. Further, the patch size being a critical hyperparameter in ViTs, it was also chosen by varying the size and checking the performance of the model on the test sets of all the five datasets used in the study. The patch sizes of 1, 3, 5, 7, and 9 were tested in the experiment (Table 5). As per the results, the patch size of 5 generated the best results in terms of accuracy, precision, recall, and F1 score for all the datasets. However, AUCs for the patch sizes of 1 and 7 were better in Apple and Rice datasets.

To select the best optimizer for improved training performance, SGD, RMSProp, Adamax, Adam, and Nadam optimizers were evaluated. According to the results presented in Table 6, the Nadam optimizer performed well on PlantVillage dataset, but it was not able to provide better results on other datasets. The Adam optimizer on the other hand, was able to maintain the consistency in the model's accuracy across all the datasets. Therefore, Adam was chosen for model building.

An ablation study was also performed for different components of the model architecture. In the ablation study, different components of the proposed model were systematically removed to assess their importance. This analysis revealed the impact of each component on the overall performance of the model. A series of experiments were performed using convolution layers, attention models and ViT blocks to arrive at a suitable architecture of the proposed plant disease detection model. Although the use of VGG blocks led to a rise in the number of parameters, it was one of the most crucial components in extracting salient patterns from the data. Likewise, the utilization of Inception blocks aided in multi-scale feature extraction. We also evaluated the performance with only ViT blocks, but it failed to perform well without the support of Conv blocks.

The performance of different versions of the model architectures were tested on all the datasets. Initially, the plain VGG blocks and then only the Inception blocks were evaluated. Then the VGG + Inception blocks, ViT + Inception blocks, and finally the combination of VGG, Inception, and ViT blocks were assessed for plant disease identification. Results of the study are presented in Table 7. On all five datasets, the model with the proposed architecture consisting of VGG, Inception and ViT blocks turned out to be the best.

4. Results and discussion

This section is devoted to the results of experiments for the performance evaluation of the proposed CNN enabled ViT model developed for plant disease detection. Comparative performance of the proposed model with other state-of-the-art CNN and ViT based models is also presented in this section. Further, the model is evaluated for the explainability of its predictions using two standard methods—gradient-weighted class activation maps (Grad-CAMs) (Selvaraju et al., 2017) and local interpretable model agnostic explanation (LIME) (Ribeiro et al., 2016). A detailed discussion on the findings of the experiments is presented towards the end of the section.

4.1. Results

The training and validation epoch-wise graph of accuracy and loss is presented in Fig. 4 for all five datasets. The graphs converge perfectly for

Table 6
Comparison of the model's performance using various optimizers.

Optimizer	Loss	Accuracy	Precision	Recall	F1 score	AUC	kappa score
Apple							
SGD	0.95	60.22	68.25	46.24	55.13	84.36	0.42
RMSProp	0.35	**93.55**	**93.55**	**93.55**	**93.55**	97.34	0.91
Adamax	0.81	68.28	73.86	60.75	66.67	88.66	0.54
Adam	0.3	**93.55**	**93.55**	**93.55**	**93.55**	97.01	0.1
Nadam	**0.28**	93.01	93.51	93.01	93.26	**97.89**	**0.9**
Embrapa							
SGD	0.72	81.31	91.72	72.42	80.94	98.99	0.81
RMSProp	0.71	83.33	83.33	80.59	81.94	97.83	0.83
Adamax	0.56	86	92.28	81.16	86.36	98.99	0.85
Adam	**0.46**	**89.24**	**91.17**	**88.27**	**89.7**	**98.73**	**0.89**
Nadam	0.53	87.04	88.72	85.77	87.22	98.41	0.87
Maize							
SGD	0.75	70.37	73.68	69.14	71.34	90.32	0.61
RMSProp	0.44	87.65	88.31	83.95	86.07	96.04	0.84
Adamax	0.44	90.12	**92.11**	86.42	89.17	96.1	0.87
Adam	**0.35**	**92.59**	92.5	**91.36**	**91.93**	**96.8**	**0.9**
Nadam	0.47	87.65	90.79	85.19	87.9	94.99	0.83
PlantVillage							
SGD	0.09	97.5	98.07	96.79	97.43	99.95	0.97
RMSProp	0.08	97.72	97.94	97.52	97.73	99.80	0.98
Adamax	0.06	98.49	98.58	98.33	98.45	99.91	0.98
Adam	0.04	98.86	98.9	98.81	98.85	99.92	0.99
Nadam	**0.03**	**99.06**	**99.14**	**98.99**	**99.06**	**99.92**	**0.99**
Rice							
SGD	0.98	70	83.78	51.67	63.92	88.8	0.63
RMSProp	0.36	91.67	94.55	86.67	90.44	98.38	0.89
Adamax	0.55	85	94	78.33	85.45	96.6	0.81
Adam	**0.24**	**95**	**98.25**	**93.33**	**95.73**	**99.39**	**0.94**
Nadam	0.66	76.67	80.36	75	77.59	94.31	0.71

Table 7
Results of the ablation study with different combinations of the model's components.

Combination	Loss	Accuracy	Precision	Recall	F1 score	AUC	kappa score
Apple							
VGG only	1.02	59.68	67.52	42.47	52.14	82.66	0.41
Inception only	1.19	39.78	57.14	44.3	49.9	72.04	0.3
ViT only	1.3	46.77	49.36	41.4	45.03	73.58	0.23
VGG + Inception	0.6	79.57	81.76	74.73	78.09	93.82	0.7
VGG + ViT	0.36	89.78	90.22	89.25	89.73	97.54	0.85
Inception + ViT	0.77	74.19	75.69	73.66	74.66	91.18	0.62
VGG + Inception + ViT	**0.3**	**93.55**	**93.55**	**93.55**	**93.55**	**97.01**	**0.91**
Embrapa							
VGG only	0.6	80.97	87.62	75.1	80.88	99.36	0.8
Inception only	1.27	61.73	76.2	46.86	58.03	98	0.6
ViT only	0.85	75.01	81.61	69.2	74.89	98.41	0.74
VGG + Inception	0.66	80.89	85.84	77.17	81.27	98.77	0.8
VGG + ViT	0.54	86.11	89.04	84.09	86.49	98.72	0.86
Inception + ViT	0.66	80.89	85.84	77.17	81.27	98.77	0.8
VGG + Inception + ViT	**0.46**	**89.24**	**91.17**	**88.27**	**89.7**	**98.73**	**0.89**
Maize							
VGG only	0.78	77.78	78.75	77.78	78.26	92.96	0.7
Inception only	0.99	61.73	58.54	29.63	39.34	82.04	0.49
ViT only	1.02	62.96	66.67	49.38	56.74	83.58	0.51
VGG + Inception	0.77	83.95	83.95	83.95	83.95	93.51	0.79
VGG + ViT	0.58	80.25	83.12	79.01	81.01	94.21	0.74
Inception + ViT	0.9	67.9	70.13	66.67	68.35	88.66	0.57
VGG + Inception + ViT	**0.34**	**92.59**	**93.67**	**91.36**	**92.5**	**97.21**	**0.9**
PlantVillage							
VGG only	0.15	95.82	96.7	95.44	96.06	99.85	0.96
Inception only	0.77	77.85	86.84	68.8	76.78	98.69	0.77
ViT only	0.33	90.27	92.09	89.51	90.78	99.3	0.9
VGG + Inception	0.07	98.2	98.3	98.2	98.25	99.88	0.98
VGG + ViT	0.12	96.82	97.07	96.69	96.88	99.71	0.97
Inception + ViT	0.2	94.75	95.37	94.28	94.82	99.52	0.95
VGG + Inception + ViT	**0.04**	**98.86**	**98.90**	**98.81**	**98.85**	**99.92**	**0.99**
Rice							
VGG only	0.65	76.67	85.42	68.33	75.93	94.71	0.7
Inception only	1.34	48.33	77.78	11.67	20.29	75.31	0.35
ViT only	1.15	61.67	75.68	46.67	57.74	84.51	0.51
VGG + Inception	0.38	85	85.45	78.33	81.74	98.19	0.81
VGG + ViT	0.3	95	94.74	90	92.31	99.17	0.94
Inception + ViT	0.92	76.67	79.63	71.67	75.44	89.59	0.71
VGG + Inception + ViT	**0.08**	**98.33**	**98.33**	**98.33**	**98.33**	**99.94**	**0.98**

the Maize, PlantVillage, and Rice datasets, whereas there is more difference in the training and validation data results for Apple and Embrapa datasets. The reason for this difference in the Embrapa dataset could be due to high imbalance and very few samples in some classes. In Apple dataset, samples having multiple diseases get misclassified due to textural similarity between different types of diseases.

The confusion matrices of the results on the test sets of Apple, Maize, and Rice datasets using the proposed model are shown in Fig. 5. As the other two datasets, PlantVillage and Embrapa, have a large number of classes, it is not possible to present the confusion matrices for them. In all three datasets, it can be seen that the proposed model has a reasonably good classification score. In Fig. 5 (a), the classifications of healthy, rust, and scab are very accurate for the Apple dataset. However, multiple diseases are highly misclassified, with only 30% of them correctly classified. Similarly, in the Maize dataset (Fig. 5 (b)), Goss's bacterial wilt and phaeosphaeria spots are classified correctly. However, 10% of the test images in the eyespot class are misclassified as gray leaf spots and phaeosphaeria spots. On the other hand, 20% of the gray leaf images are misclassified as eye spots and phaeosphaeria spots. In the Rice (see Fig. 5 (c)) dataset, the model correctly classifies images of bacterial leaf streak, leaf smut, stackburn, and white tip. Only 6.67% of the leaf scald images are misclassified as bacterial leaf streaks. It can be observed that leaves with more than one disease are highly misclassified in the Apple dataset. On the other hand, diseases having similar patterns, i.e., spots, are misclassified. These are important challenges, not only for the proposed model, but for all existing models.

The proposed model is compared with nine recently introduced plant disease classification models. Five of these are attention based CNN models (Chen et al., 2021d, 2021b, 2021c; Karthik et al., 2020; Zhao et al., 2022) and four models have used ViT in their architecture (Borhani et al., 2022; Li et al., 2022; Thai et al., 2021; Thakur et al., 2022c). Table 8 presents the quantitative performance results of all the nine models under comparison and the proposed model on five different datasets, using the performance metrics mentioned in the table. The proposed model has an accuracy of 93.55%, 89.24%, 92.59%, 98.86%, and 98.33% on Apple, Embrapa, Maize, PlantVillage, and Rice datasets, respectively. Table 8 illustrates that the proposed model outperforms all the CNN and ViT models (Borhani et al., 2022; Chen et al., 2021d, 2021b, 2021c; Karthik et al., 2020; Li et al., 2022; Thai et al., 2021; Thakur et al., 2022c; Zhao et al., 2022) on all the datasets, it is able to attain the top performance across all the measures. Fig. 6 also presents histograms of values of six quantitative measures attained by nine existing models and the proposed model on five datasets.

The results of performance comparison with other models indicate that the suitably chosen combination of transformer blocks and VGG + Inception blocks has resulted in better feature extraction for each category of disease. The ablation study performed in this work has worked well in selecting the best combination of different units. These findings are further validated by t-SNE plots as detailed below.

The t-SNE method, which displays feature similarity and dissimilarity for samples of the same and different classes in a dataset, is employed to illustrate the efficacy of the proposed method in feature learning(Van der Maaten and Hinton, 2008). The learned features by the proposed model and other competing models for each of the datasets are projected onto the 2D-plane using the t-SNE algorithm, an unsupervised nonlinear dimensionality reduction technique (Van der Maaten and

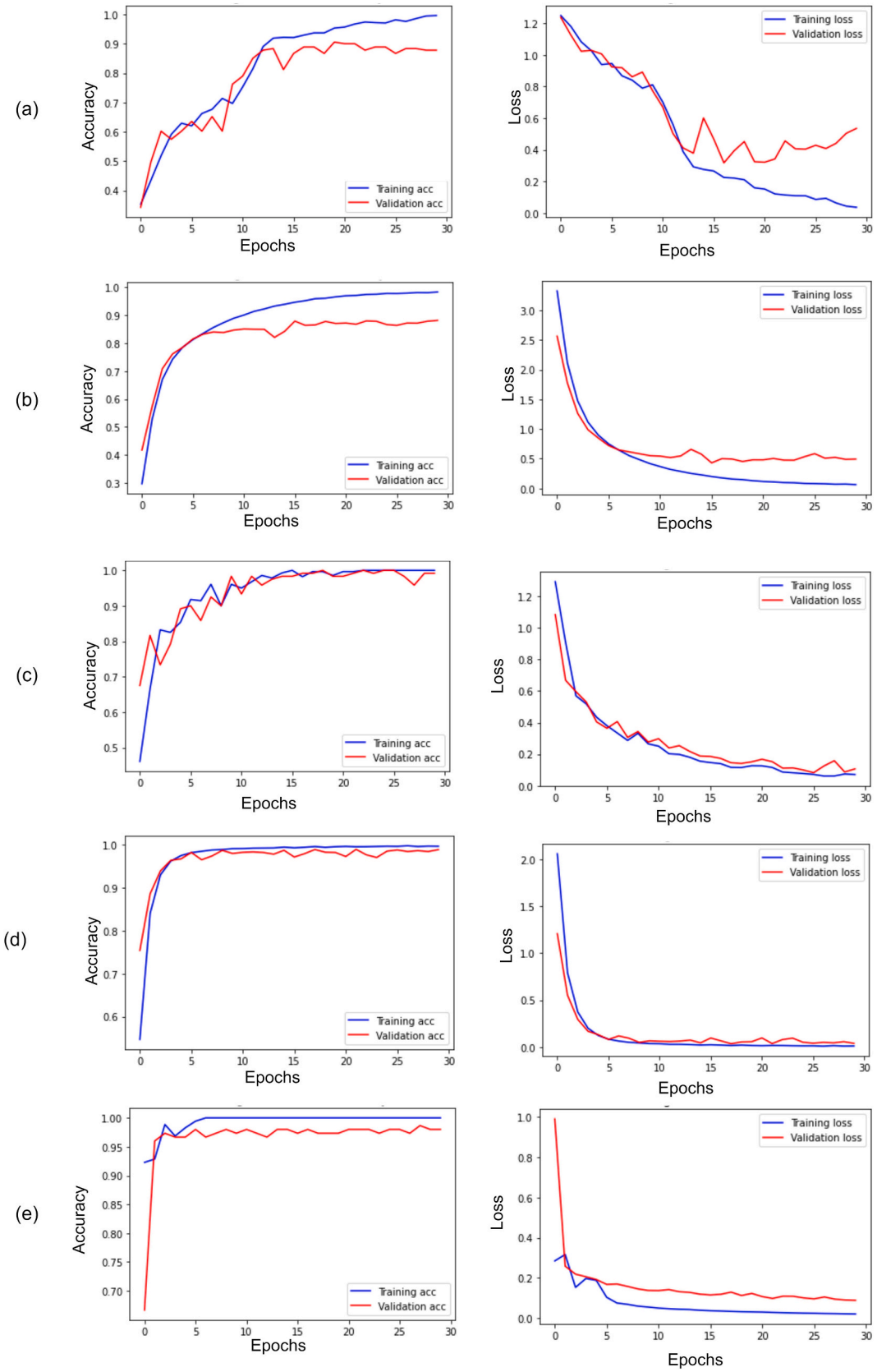


Fig. 4. Accuracy and loss graphs for (a) Apple, (b) Embrapa, (c) Maize, (d) PlantVillage, and (e) Rice datasets.

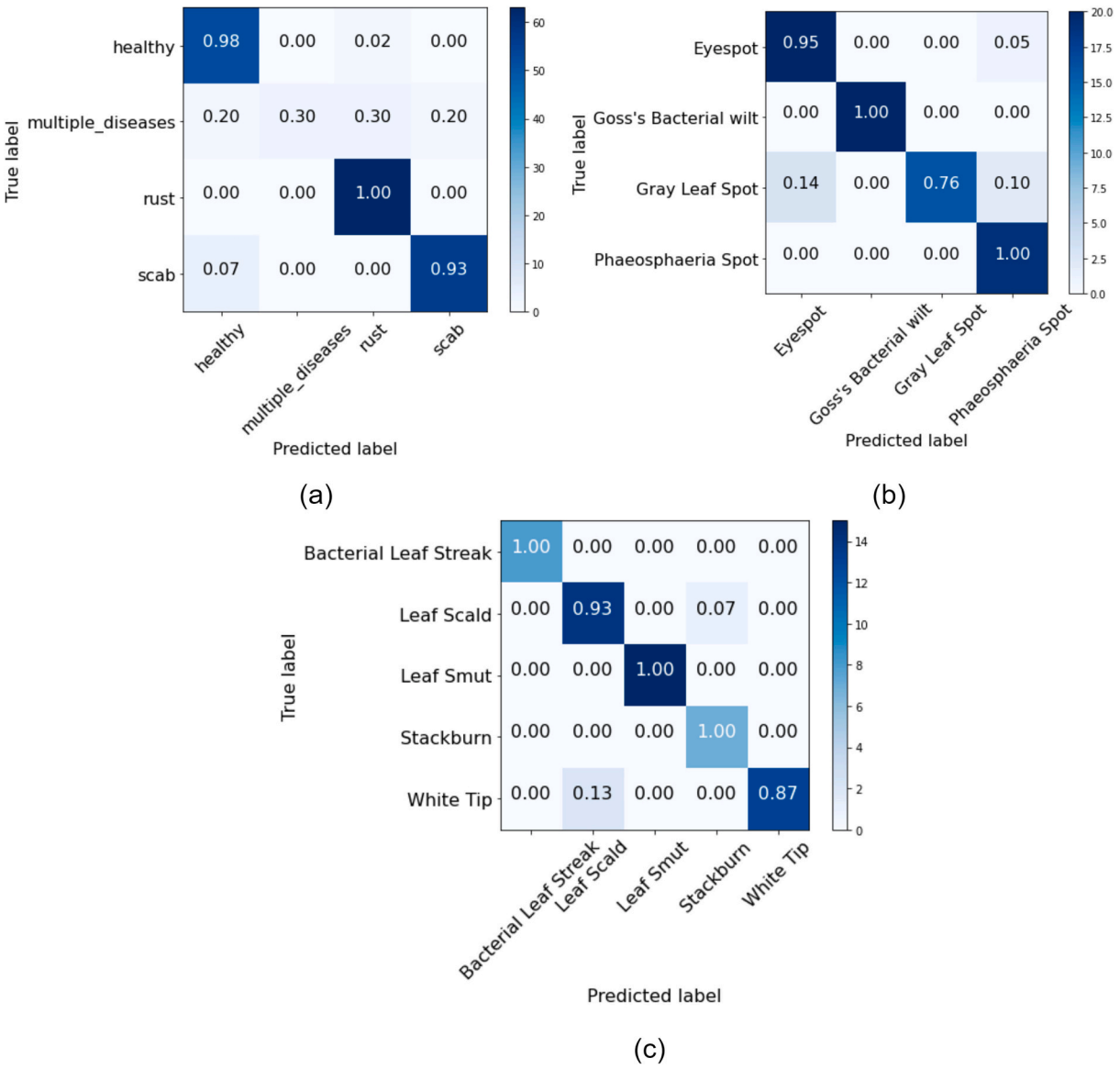


Fig. 5. Normalized confusion matrices for (a) Apple, (b) Maize, and (c) Rice datasets.

Hinton, 2008). The visualization results using 2-D vector scatter plots obtained with the t-SNE method for all the five datasets are shown in Fig. 7. The output of the last convolutional/reshape layer is used to visualize the feature maps. In the Apple dataset (refer Fig. 7 (1)), red color denotes the healthy class, lemon green denotes multiple diseases in a single image, the blue color denotes the rust disease, and the green color denotes the scab disease. Fig. 7 (1)(j) shows that the features produced by the proposed model are easily distinguishable as the clusters of the features are the best among all the methods for three disease classes. Thus, it can be concluded that the proposed model is quite efficient at understanding the salient features of different classes in a dataset. However, categorization becomes challenging when there are multiple diseases in a sample. In this case, the model can misinterpret the disease class, as can be observed in 7, Column 1 for Apple dataset. In this case, the feature produced by the proposed model make three clusters only whereas there are four distinct disease categories in the dataset. It may be mentioned that the dataset contains samples having multiple disease labels.

Grad-CAM (Selvaraju et al., 2017) results for the proposed model and other competing models are also shown in Fig. 8 using a sample of each of the five datasets. It is worth noting that the model developed by Karthik et al. (2020) is not able to identify the correct disease region for any of the datasets, while other models identify the portion from the entire leaf with greater precision in the Apple, Maize, and Rice datasets (Chen et al., 2021d, 2021c,b). Further, the model by Zhao et al. (2022) is able to recognize the disease portion in the Apple and Maize datasets reasonably well. But the results are not good for the Embrapa, PlantVillage, and Rice datasets. One may notice that the model by Thai et al. (2021) is hardly able to correctly identify the illness portion in any of the datasets. Again, the models by Borhani et al. (2022) and Li et al. (2022) miss the disease regions. The model by Thakur et al. (2022c) also fails to recognize the disease in Apple and Maize datasets, but it performs better on other datasets. In contrast, the proposed model shows better capability in locating the disease regions. It has been observed that the models using pretrained networks perform better in locating the disease regions, but some of these models have concentrated on a larger portion around the actual disease. This is probably due to the CNN's nature of focusing on the regions around the points of saliency to extract features. The previous models using a combination of CNN and ViT appear to be less effective in terms of CAM as compared to the proposed model. The reason could be the multi-scale feature extraction property of the Inception block combined with the initial pretrained VGG layers that

Table 8
Performance comparison of the proposed model with state-of-the-art models.

Approach	Loss	Accuracy	Precision	Recall	F1 score	AUC	kappa score
Apple							
Karthik et al. (2020)	1.56	62.37	62.84	61.83	62.35	83.53	0.45
Chen et al. (2021d)	1.96	55.91	56.35	54.84	55.58	78.32	0.36
Chen et al. (2021b)	0.79	83.33	83.7	82.8	83.25	94.1	0.76
Chen et al. (2021c)	0.63	83.33	85.47	82.26	83.83	95.59	0.76
Zhao et al. (2022)	0.58	88.71	89.62	88.17	88.89	95.81	0.84
Thai et al. (2021)	0.55	83.33	92.31	77.42	84.21	93.76	0.77
Borhani et al. (2022)	4.02	45.16	45.41	45.16	45.28	69.08	0.21
Li et al. (2022)	1.34	41.94	44.59	35.48	39.52	71.68	0.16
Thakur et al. (2022c)	1.03	85.42	74.86	71.64	73.21	83.24	0.81
Proposed	**0.3**	**93.55**	**93.55**	**93.55**	**93.55**	**97.01**	**0.91**
Embrapa							
Karthik et al. (2020)	0.77	80.29	83.07	78.38	80.6	97.8	0.82
Chen et al. (2021d)	0.6	80.95	85.08	76.02	80.3	98.02	0.78
Chen et al. (2021b)	1.11	73.63	80.12	67.86	73.48	98.02	0.73
Chen et al. (2021c)	1.12	74.88	82.93	66.06	73.18	98.2	0.75
Zhao et al. (2022)	0.36	88.48	89.13	87.44	88.28	98.23	0.88
Thai et al. (2021)	1.12	76.77	95.34	60.71	74.18	97.56	0.76
Borhani et al. (2022)	1.03	79.44	80.63	78.7	79.65	96.43	0.79
Li et al. (2022)	0.9	73.17	79.23	67.59	72.95	98.38	0.72
Thakur et al. (2022c)	0.48	87.87	84.7	80.75	82.68	89.98	0.88
Proposed	**0.46**	**89.24**	**91.17**	**88.27**	**89.7**	**98.73**	**0.89**
Maize							
Karthik et al. (2020)	1.17	54.32	59.72	53.09	56.21	83.19	0.39
Chen et al. (2021d)	1.26	49.38	50.85	37.04	42.86	76.16	0.33
Chen et al. (2021b)	0.6	87.65	87.65	87.65	87.65	96.03	0.84
Chen et al. (2021c)	0.5	88.89	91.03	87.65	89.31	96.78	0.85
Zhao et al. (2022)	1.4	77.78	78.75	77.78	78.26	94.55	0.7
Thai et al. (2021)	0.88	91.36	95	23.46	37.63	96.76	0.88
Borhani et al. (2022)	2.34	70.37	70.37	70.37	70.37	82.05	0.6
Li et al. (2022)	1.38	50.62	51.47	43.21	46.98	72.47	0.34
Thakur et al. (2022c)	0.94	84.09	83.9	83.93	83.91	89.32	0.79
Proposed	**0.34**	**92.59**	**93.67**	**91.36**	**92.5**	**97.21**	**0.9**
PlantVillage							
Karthik et al. (2020)	0.16	95.83	96.2	95.6	95.89	99.7	0.96
Chen et al. (2021d)	0.4	87.94	89.59	86.71	88.07	99.14	0.85
Chen et al. (2021b)	0.17	96.61	97.09	96.11	75.03	99.66	0.96
Chen et al. (2021c)	0.17	96.68	97.49	95.83	96.64	99.26	0.97
Zhao et al. (2022)	0.12	97.28	97.49	97.06	97.27	99.78	0.97
Thai et al. (2021)	0.28	96.46	99.33	92.44	95.76	99.65	0.96
Borhani et al. (2022)	0.53	90.13	90.59	89.89	90.24	98.25	0.9
Li et al. (2022)	0.46	86.51	88.85	85.08	86.92	98.9	0.86
Thakur et al. (2022c)	0.07	98.61	98.24	98.33	98.28	99.01	0.98
Proposed	**0.04**	**98.86**	**98.9**	**98.81**	**98.85**	**99.92**	**0.99**
Rice							
Karthik et al. (2020)	0.99	71.67	79.55	58.33	69.31	88.26	0.64
Chen et al. (2021d)	0.75	76.67	83.02	73.33	77.87	92.93	0.71
Chen et al. (2021b)	0.3	96.67	96.67	96.67	96.67	98.97	0.95
Chen et al. (2021c)	0.25	95	96.61	96.61	96.61	99.72	0.94
Zhao et al. (2022)	0.41	91.67	91.67	91.67	91.67	98.55	0.89
Thai et al. (2021)	1.02	71.67	99	91.2	94.94	96.18	0.65
Borhani et al. (2022)	0.75	80	85.71	80	82.76	94.94	0.75
Li et al. (2022)	1.16	53.33	54.72	48.33	51.33	84.45	0.41
Thakur et al. (2022c)	0.57	92.19	92.08	93	92.54	95.52	0.9
Proposed	**0.08**	**98.33**	**98.33**	**98.33**	**98.33**	**99.94**	**0.98**

makes a powerful CNN structure for pattern recognition. These results have been verified for a large number of samples from the test datasets and for a majority of images, the models show the similar behaviour. For brevity, only one sample is chosen from each dataset to show representative results.

LIME is another interpretability method that uses model-agnostic features for analyzing the results of a classifier (Ribeiro et al., 2016). It is based on a local linear approximation of the model. The LIME results for the proposed model and other comparative models are presented in Fig. 9. The first row (a) shows the input image from each dataset. It may be noted that the model by Karthik et al. (2020) is again not able to identify the disease region correctly for the sample from the Embrapa and PlantVillage dataset. The reason could be its low convergence on all the datasets. Similarly, the model by Chen et al. (2021b) is not able to identify the disease correctly. The model by Chen et al. (2021d) generates comparatively better results. Furthermore, the model by Chen et al. (2021c) generates good results for Apple and Rice datasets, but not for other samples. These models have already been trained on the ImageNet dataset, which helps them in faster convergence and better understanding of the salient portions in images. However, the LIME results for Zhao et al. (2022) indicate that the results are not very explainable on any of the five datasets. Also, the model given by Thai et al. (2021) is not able to generate good results in any of the datasets. Again, the model introduced by Borhani et al. (2022) is partially focusing on the disease portion in all the datasets. The model by Li et al. (2022) is generating better results on the Embrapa and Maize datasets, while the model by Thakur et al. (2022c) is able to generate satisfactory results on the Apple and Maize datasets only. It is indeed remarkable that the proposed model can more accurately capture the disease portion with better precision in all the five datasets. These samples are just

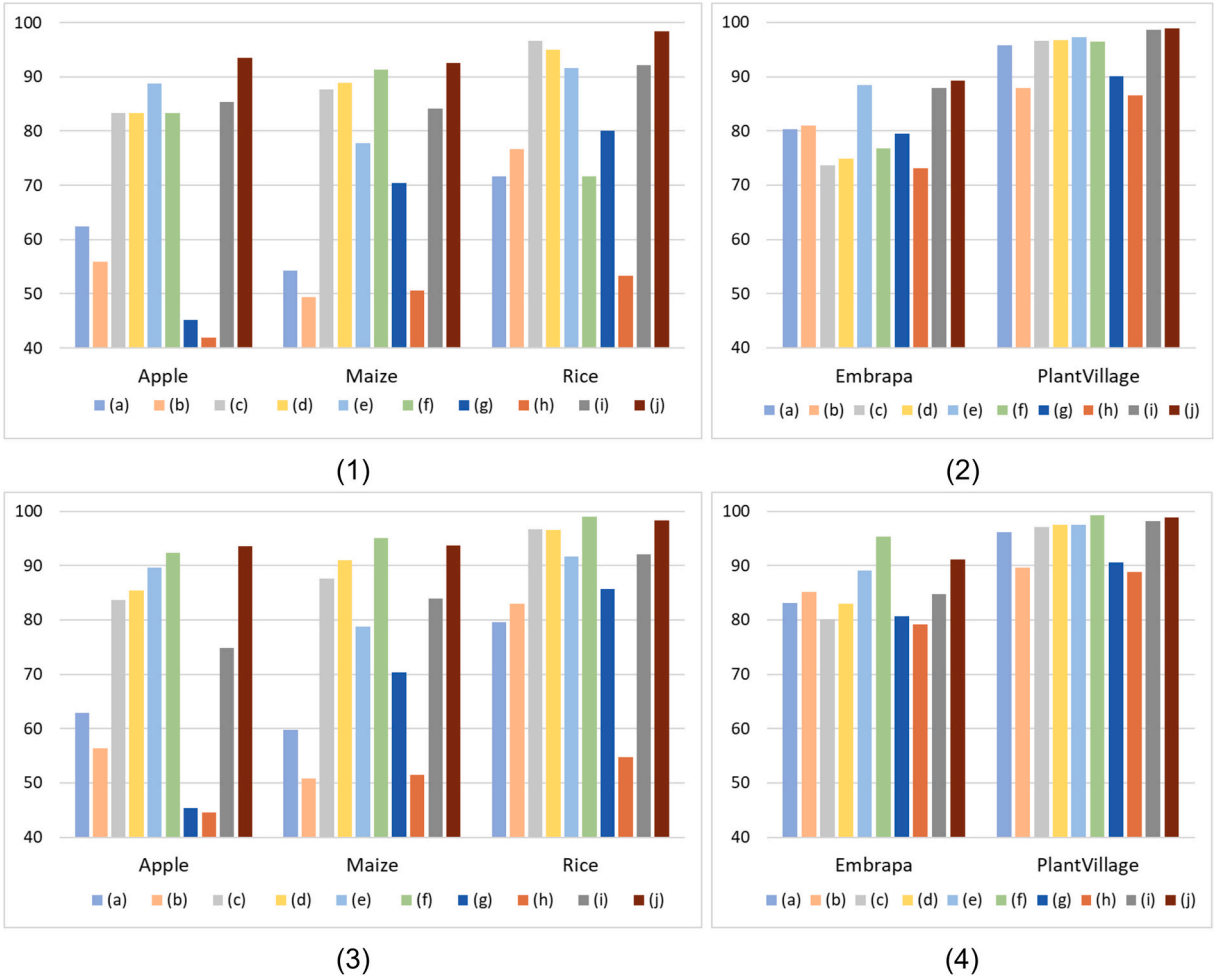


Fig. 6. Histograms of values for Accuracy on (1) small datasets (2) large datasets, Precision on (3) small datasets, and (4) large datasets attained by existing models and the proposed model. (a) Karthik et al. (2020) (b) Chen et al. (2021d) (c) Chen et al. (2021b) (d) Chen et al. (2021c) (e) Zhao et al. (2022) (f) Thai et al. (2021), (g) Borhani et al. (2022), (h) Li et al. (2022), (i) Thakur et al. (2022c) and (j) Proposed model.

representative of how the methods are interpreting different types of diseases. The results have been verified for a large number of samples from the test datasets.

4.2. Discussion

The proposed model is designed to aid the developments in artificial intelligence based crop monitoring and disease control solutions. For deploying these solutions in IoT devices like drones or smart phones, one needs efficient lightweight models. When the number of parameters are reduced to make a model suitable for smart agriculture solutions, it can not capture the large variance in various disease types. Due to similarity in disease patterns that can even be challenging for a naive human eyes, one can not expect that a lightweight model will be able to capture those details for prediction. Nonetheless, efforts are made by researchers to produce networks with lightweight but efficient architectures. The proposed model shows its efficacy for such applications, as it can work well on a variety of crops and a large set of plant disease species with a small set of trainable parameters.

Table 9 presents an account of the memory requirement, GFLPOS count, average test time, pros, and cons of the proposed method as well as other competing methods. It may be noted that the proposed model has relatively less number of trainable and total parameters, except for the models in (Borhani et al., 2022; Karthik et al., 2020; Thakur et al., 2022c). However, performance of these models deteriorates on various small datasets. For example, performance of the model by Karthik et al. (2020) is not at par with the proposed model on small datasets like Apple, Embrapa, Maize, and Rice. Similar is the case for the model by Borhani et al. (2022) and Thakur et al. (2022c). These models are not able to capture the disease patterns due to significantly low number of parameters. The experimental results indicate that a balance between the model size and performance needs to be maintained when selecting a lightweight architecture, especially when the training datasets are small in size.

A subtle point of discussion is the effectiveness of a model in terms of the explainability of its predictions. It may be observed from the results of Grad-CAM and LIME (cf. Figs. 8 and 9) that not every combination of ViT and CNN can provide explainable results. Interestingly, the inception blocks in conjunction with ViT blocks in the proposed model have probably helped in improving the explainability of its results. The reason is that the inception blocks capture multi-scale features simultaneously, and that can help the network learn and interpret the patterns better.

A model's suitability for IoT devices does not only depend upon its low memory footprint, it also heavily depends on its computational needs typically measured in FLOPS. One downside of the proposed model is its relatively higher FLOPS count due to the operations involved in the Inception block. While the inception blocks help in better interpretability of the model, they demand high computation. Observing the GRAD-CAM and LIME results and the quantitative measure of the proposed model, it can be concluded that the combination of initial VGG16 blocks, the inception blocks and four ViT blocks has worked well in identifying disease features. The approach by Chen et al. (2021b) has the

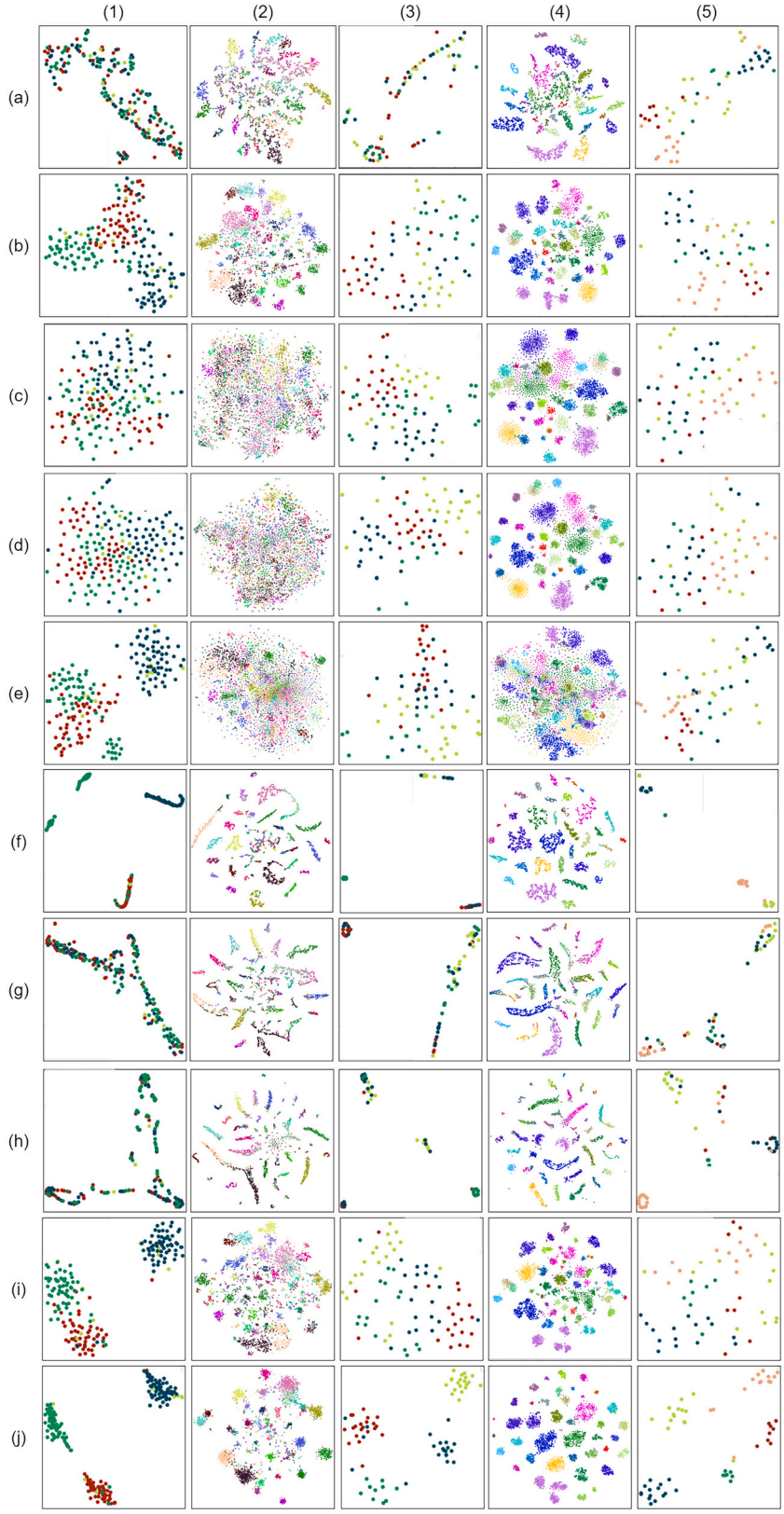


Fig. 7. t-SNE plots for (1) Apple, (2) Embrapa, (3) Maize, (4) PlantVillage, and (5) Rice datasets. (a) Karthik et al. (2020) (b) Chen et al. (2021d) (c) Chen et al. (2021b) (d) Chen et al. (2021c) (e) Zhao et al. (2022) (f) Thai et al. (2021), (g) Borhani et al. (2022), (h) Li et al. (2022), (i) Thakur et al. (2022c) and (j) Proposed model.

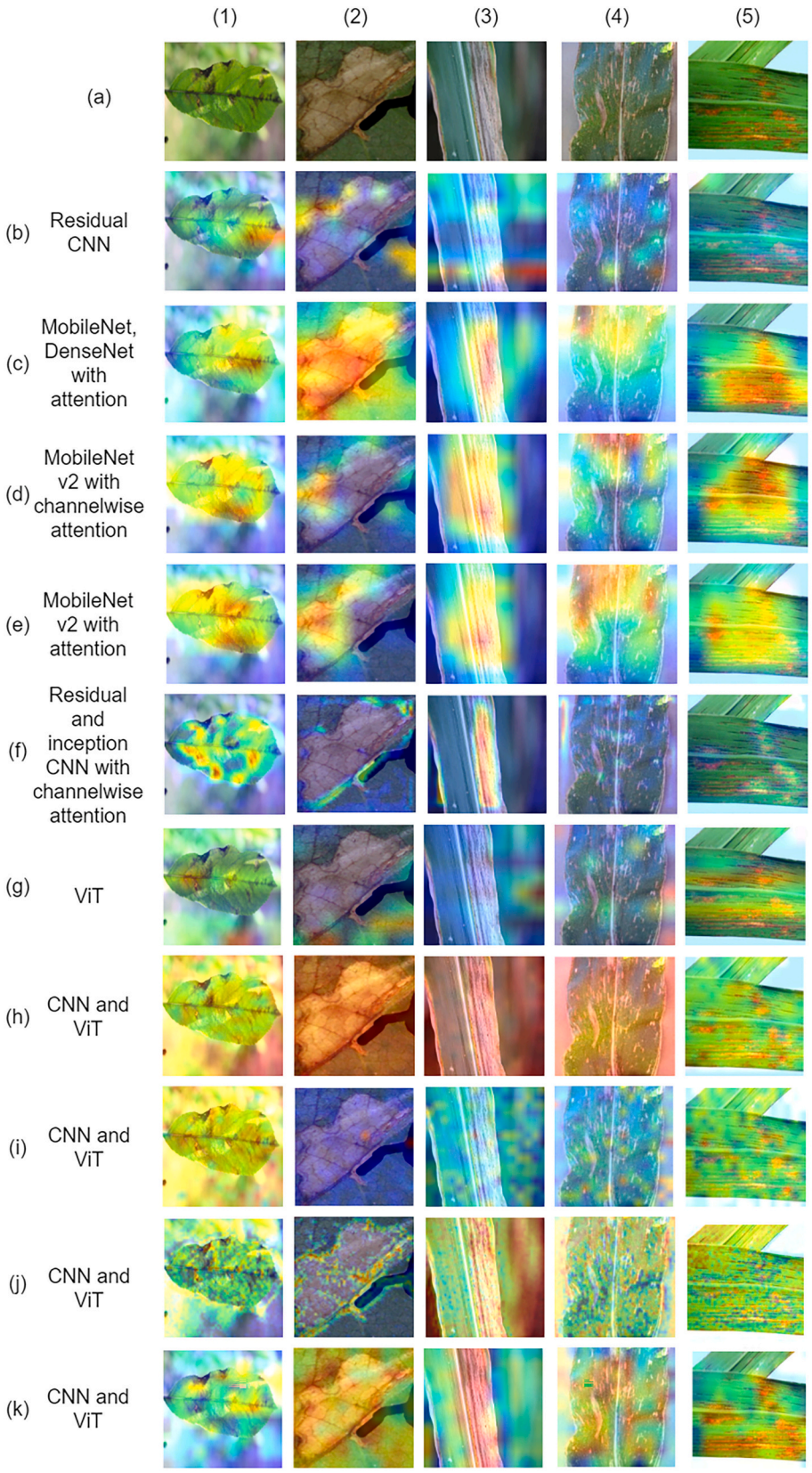


Fig. 8. Grad-CAMs for (1) Apple, (2) Embrapa, (3) Maize, (4) PlantVillage, and (5) Rice datasets. (a) input image (b) Karthik et al. (2020) (c) Chen et al. (2021d) (d) Chen et al. (2021b) (e) Chen et al. (2021c) (f) Zhao et al. (2022) (g) Thai et al. (2021), (h) Borhani et al. (2022), (i) Li et al. (2022), (j) Thakur et al. (2022c) and (k) Proposed model.

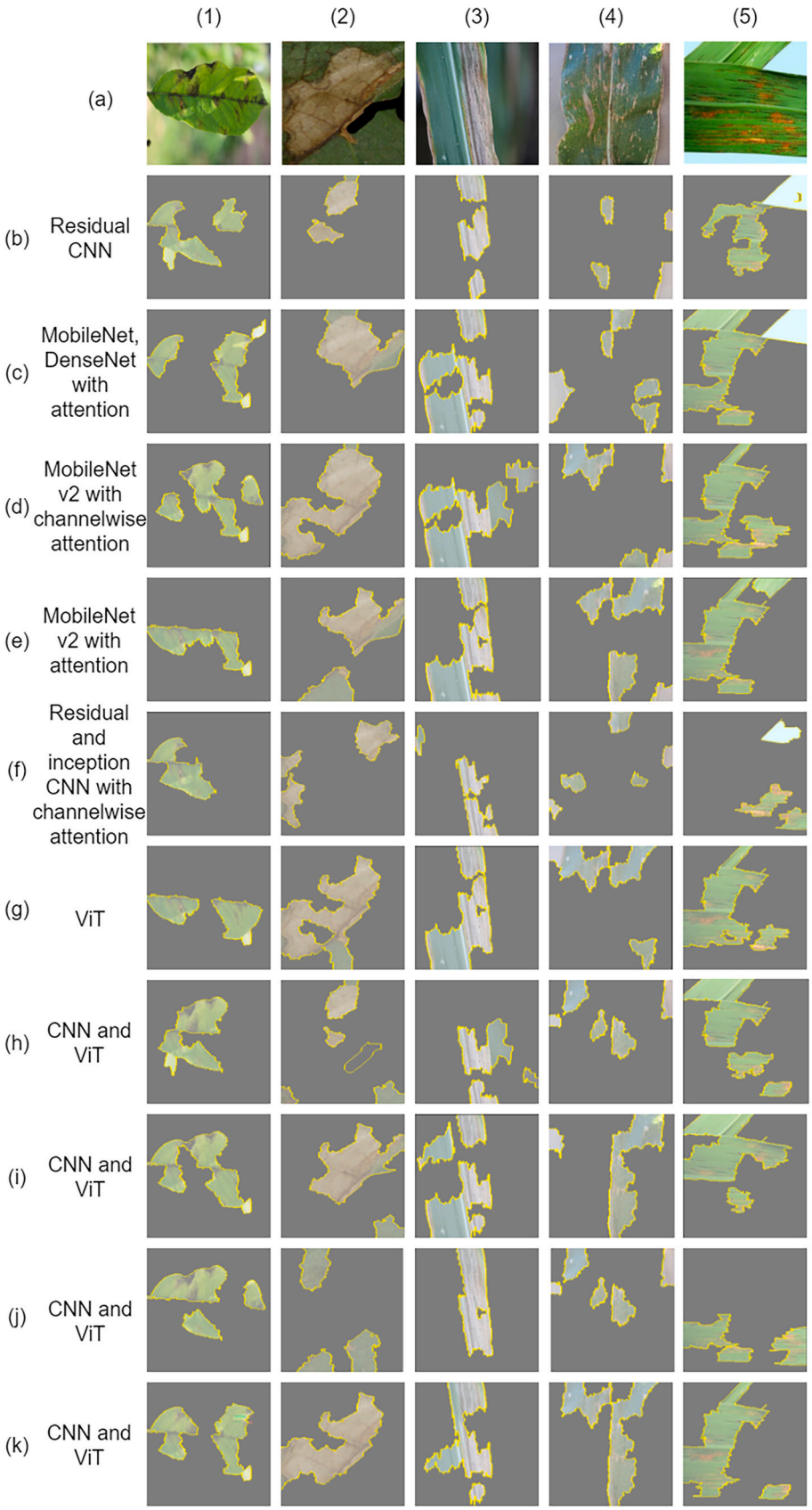


Fig. 9. LIME for (1) Apple, (2) Embrapa, (3) Maize, (4) PlantVillage, and (5) Rice datasets. (a) input image (b) Karthik et al. (2020) (c) Chen et al. (2021d) (d) Chen et al. (2021b) (e) Chen et al. (2021c) (f) Zhao et al. (2022) (g) Thai et al. (2021), (h) Borhani et al. (2022), (i) Li et al. (2022), (j) Thakur et al. (2022c) and (k) Proposed model.

Table 9
Comparison of the trainable parameters, FLOPS, memory footprint, average testing time, pros and cons.

Author	Approach	Total params (M)	Trainable params(M)	GFLOPS	Memory (MB)	Avg test time (ms)	Pros	Cons
Karthik et al. (2020)	Residual CNN	0.72	0.72	3.59	2.8	**7**	Lightweight model	Poor performance
Chen et al. (2021d)	MobileNet, DenseNet with attention	0.82	0.82	3.4	**0.76**	11	Lightweight model	Poor performance
Chen et al. (2021b)	MobileNet v2 with channelwise attention	4.32	2.06	0.78	16.8	16	Comparative performance for Apple, Maize, PlantVillage, and Rice datasets	Comparatively heavy model
Chen et al. (2021c)	MobileNet v2 with attention	4.32	2.06	0.83	16.9	18	Comparative performance for Apple, Maize, PlantVillage, and Rice datasets	Comparatively heavy model
Zhao et al. (2022)	Residual and inception CNN with channelwise attention	6.71	6.71	11.9	25.8	13	Comparative performance for Embrapa and PlantVillge and Rice datasets	Comparatively heavy model
Thai et al. (2021)	ViT	85.79	85.79	35.2	327.5	44	Comparative performance for Maize and PlantVillage datasets	Complex and heavy model
Borhani et al. (2022)	CNN and ViT	**0.4**	**0.4**	**0.22**	1.7	11	Minimum number of trainable parameters and FLOPS	The performance is comparatively poor
Li et al. (2022)	CNN and ViT	0.8	0.8	0.23	0.78	40	Lightweight model	Poor performance
Thakur et al. (2022c)	CNN and ViT	0.49	0.49	4.9	27.7	12	Lightweight model with less number of trainable parameters	High memory footprint
Proposed	CNN and ViT	0.85	0.85	11.8	3.4	18	Achieved best performance	Higher FLOPS count

least FLOPS count. However, that is maintained at the cost of model's classification efficiency, as can be seen from its performance results with respect to all the measures. It may also be observed that the models with lower FLOPS count have recorded significantly low values of AUC on all the five datasets as compared with the proposed model. Therefore the degree of separability across different classes is well recognized and demonstrated by the proposed model. The model given by Thai et al. (2021) has the highest number of trainable parameters and FLOPS count as it uses a fine-tuned version of the original ViT model. Despite this, their model's performance is not at par with the proposed model or some of the other approaches. In small datasets especially, the performance of pure ViT model deteriorates drastically. The reason could be the limitation of ViT in capturing the local features when it is not having enough samples to learn.

The model by Borhani et al. (2022) has the least number of trainable parameters but its performance is not at par with any of the models on all the five datasets. The model by Li et al. (2022) also has comparatively less number of trainable parameters and FLOPS count; however its performance is not consistent on different datasets. Similarly, the previous model by the present authors Thakur et al. (2022c) has less number of trainable parameters but suffers from high FLOPS count. In a way, the proposed model is able to maintain a balance between, the FLOPS, Memory requirement and its classification efficiency.

Considering the average test time of the models, Karthik et al. (2020) uses the lowest average test times, followed by Chen et al. (2021d), Zhao et al. (2022), Borhani et al. (2022), and Thakur et al. (2022c). The proposed model could attain a moderate test time among all the methods. The models with a lower average test time do not exhibit the precision level at par with other models with comparatively heavier architectures. (Please refer 8 and 8). In contrast, the proposed model is comparatively lightweight while maintaining the best performance among all the datasets. The only issue is the higher FLOPS count of the proposed model that results in a comparatively higher average test time. It is worth mentioning that there are not many works that utilize the potential of ViT to build efficient and lightweight plant disease detection model. Methods have emerged that combine the capabilities of CNN with ViT for further improvement in this direction.

Another crucial aspect to consider is the generalizability of the model. To assess this, the proposed model is first pretrained on the PlantVillage dataset. Subsequently, it is evaluated on the Apple dataset, which is an in-field dataset. The results are presented in Table 10. Although the performance of the model is superseded by that of Chen et al. (2021b), it is the second-best performing model on four out of six performance measures and the best performing model in terms of Kappa score. The model by Chen et al. (2021b) achieves higher performance in terms of accuracy, precision, recall, F1 score and AUC. But this model has a significantly high number of trainable parameters as compared to the proposed model.

Another important aspect is the limitations in the model development due to the image capturing environment in various datasets. Most of the datasets are primarily created by image focus on a single leaf, whereas in a real-world scenario, there are multiple leaves with complex backgrounds. This may limit the model's performance in real life scenarios. Nonetheless, the datasets included in the present study provide

Table 10
Comparison of model's generalizability with other state-of-the-art models on the Apple dataset.

Approach	Loss	Accuracy	Precision	Recall	F1 score	AUC	Kappa score
Karthik et al. (2020)	21.31	18.67	19.78	18.33	19.03	61.33	**0.07**
Chen et al. (2021d)	11.93	33.5	34.72	33.5	34.1	65.18	0.14
Chen et al. (2021b)	**1.79**	**69**	73.6	**67.83**	**70.6**	**91.69**	0.2
Chen et al. (2021c)	1.96	54.33	**81.69**	46.83	59.53	90.37	0.3
Zhao et al. (2022)	18.52	29.17	30.51	28.83	29.65	65.25	0.11
Thai et al. (2021)	4.21	66.33	67.23	66.33	66.78	81.16	0.01
Borhani et al. (2022)	8.02	64.33	64.92	64.17	64.54	82.55	0.18
Li et al. (2022)	3.84	44.83	48.49	42.83	45.48	78.9	**0.07**
Thakur et al. (2022c)	5.7	65.83	65.94	65.83	65.88	83.17	0.12
Proposed	4.3	66.67	67	66.67	66.83	83.42	**0.07**

many varieties of crops with multiple diseases as well as a single disease. In the present study, five different types of datasets are used, that definitely makes the proposed model suitable for IoT applications than the existing models. Nonetheless, there are several challenges that need to be addressed in the area of plant disease detection and classification. For example, high similarity of disease patterns, colors and textures, and diseases that do not show initial symptoms of plant leaves pose challenges in developing AI solutions. Climate change has also resulted in emergence of new disease species and their spread behaviour. Dealing with these problems requires new methods and techniques.

5. Conclusion and future scope

In the present work, a ViT-enabled CNN model is proposed for plant disease detection and identification. The model combines the global feature extraction property of the transformer with the inherent locality of the CNN to achieve better classification performance. The experimental results demonstrate that the model's performance is impressive on five publicly available datasets of different sizes with images captured under varying background conditions. The proposed model achieves 93.55%, 89.24%, 92.59%, 98.86%, and 98.33% overall accuracy on Apple, Embrapa, Maize, PlantVillage, and Rice datasets, respectively. It is remarkable to note that the model outperforms nine state-of-the-art models in identifying plant diseases. The model is also evaluated for the interpretability of its prediction results using the Grad-CAM and LIME methods, and the results show that the model is reasonably interpretable. t-SNE plots demonstrate that the model captures distinct features of different disease categories.

The only hindrance to the applicability of the proposed model in IoT based systems is its relatively high computational demand. In the future, it is planned to work on reducing the FLOPS while maintaining the model's efficiency. Another important challenge that none of the existing methods are able to address is the similarity of disease patterns. More efficient models that understand the distinctive features of different diseases will definitely be in much demand. Although, the existing methods of interpreting the model's prediction are used in the present work, more suitable methods to interpret the model's performance will be explored.

Data availability

Dataset is publicly available.

References

Abdu, A.M., Mokji, M.M., Sheikh, U.U., 2020. Automatic vegetable disease identification approach using individual lesion features. Comput. Electron. Agric. 176, 105660.

Atila, Ü., Uçar, M., Akyol, K., Uçar, E., 2021. Plant leaf disease classification using efficientnet deep learning model. Ecol. Inform. 61, 101182.

Barbedo, J.G.A., 2018. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput. Electron. Agric. 153, 46–53.

Barbedo, J.G.A., Koenigkan, L.V., Halfeld-Vieira, B.A., Costa, R.V., Nechet, K.L., Godoy, C.V., Junior, M.L., Patricio, F.R.A., Talamini, V., Chitarra, L.G., et al., 2018. Annotated plant pathology databases for image-based detection and recognition of diseases. IEEE Lat. Am. Trans. 16 (6), 1749–1757.

Borhani, Y., Khoramdel, J., Najafi, E., 2022. A deep learning based approach for automated plant disease classification using vision transformer. Sci. Rep. 12 (1), 1–10.

Chen, J., Chen, J., Zhang, D., Sun, Y., Nanehkaran, Y.A., 2020a. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 173, 105393.

Chen, J., Yin, H., Zhang, D., 2020b. A self-adaptive classification method for plant disease detection using gmdh-logistic model. Sustain. Comput. Inform. Syst. 28, 100415.

Chen, X., Zhou, G., Chen, A., Yi, J., Zhang, W., Hu, Y., 2020c. Identification of tomato leaf diseases based on combination of abck-bwtr and b-arnet. Comput. Electron. Agric. 178, 105730.

Chen, J., Zhang, D., Suzauddola, M., Nanehkaran, Y.A., Sun, Y., 2021a. Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning. IET Image Processing 15 (5), 1115–1127.

Chen, J., Zhang, D., Suzauddola, M., Zeb, A., 2021b. Identifying crop diseases using attention embedded mobilenet-v2 model. Appl. Soft Comput. 113, 107901.

Chen, J., Zhang, D., Zeb, A., Nanehkaran, Y.A., 2021c. Identification of rice plant diseases using lightweight attention networks. Expert Syst. Appl. 169, 114514.

Chen, J., Wang, W., Zhang, D., Zeb, A., Nanehkaran, Y.A., 2021d. Attention embedded lightweight network for maize disease recognition. Plant Pathology 70 (3), 630–642.

Chouhan, S.S., Singh, U.P., Jain, S., 2020. Applications of computer vision in plant pathology: a survey. Arch. Comput. Methods Eng. 27, 611–632.

Chouhan, S.S., Singh, U.P., Sharma, U., Jain, S., 2021. Leaf disease segmentation and classification of jatropha curcas l. and pongamia pinnata l. biofuel plants using computer vision based approaches. Measurement 171, 108796.

DESA, 2019. World population prospects 2019. https://www.un.org/development/desa/publications/world-population-prospects-2019-highlights.html. Online; accessed 30-05-2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv: 2010.11929.

FAO, 2021. New standards to curb the global spread of plant pests and diseases. http://www.fao.org/news/story/en/item/1187738/icode/. Accessed: 2021-09-04.

Gokulnath, B., et al., 2021. Identifying and classifying plant disease using resilient lf-cnn. Ecol. Inform. 63, 101283.

Hamdani, H., Septiarini, A., Sunyoto, A., Suyanto, S., Utaminingrum, F., 2021. Detection of oil palm leaf disease based on color histogram and supervised classifier. Optik 245, 167753.

Hou, C., Zhuang, J., Tang, Y., He, Y., Miao, A., Huang, H., Luo, S., 2021. Recognition of early blight and late blight diseases on potato leaves based on graph cut segmentation. J. Agric. Food Res. 5, 100154.

Huang, S., Zhou, G., He, M., Chen, A., Zhang, W., Hu, Y., 2020. Detection of peach disease image based on asymptotic non-local means and pcnn-ipelm. IEEE Access 8, 136421–136433.

Hughes, D., Salathé, M., et al., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv: 1511.08060.

Johannes, A., Picon, A., Alvarez-Gila, A., Echazarra, J., Rodriguez-Vaamonde, S., Navajas, A.D., Ortiz-Barredo, A., 2017. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput. Electron. Agric. 138, 200–209.

Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., Menaka, R., 2020. Attention embedded residual cnn for disease detection in tomato leaves. Appl. Soft Comput. 86, 105933.

Keceli, A.S., Kaya, A., Catal, C., Tekinerdogan, B., 2022. Deep learning-based multi-task prediction system for plant disease and species detection. Ecol. Inform. 69, 101679.

Kumar, S., Sharma, B., Sharma, V.K., Sharma, H., Bansal, J.C., 2020. Plant leaf disease identification using exponential spider monkey optimization. Sustain. Comput. Inform. Syst. 28, 100283.

Li, X., Li, S., 2022. Transformer help cnn see better: a lightweight hybrid apple disease identification model based on transformers. Agriculture 12 (6), 884.

Li, H., Li, S., Yu, J., Han, Y., Dong, A., 2022. Plant disease and insect pest identification based on vision transformer. In: International Conference on Internet of Things and Machine Learning (IoTML 2021), vol. 12174. SPIE, pp. 194–201.

Li, E., Wang, L., Xie, Q., Gao, R., Su, Z., Li, Y., 2023. A novel deep learning method for maize disease identification based on small sample-size and complex background datasets. Ecol. Inform. 75, 102011.

Lu, X., Yang, R., Zhou, J., Jiao, J., Liu, F., Liu, Y., Su, B., Gu, P., 2022. A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. J. King Saud Univ. Comput. Inform. Sci. 34 (5), 1755–1767.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419.

Neyshabur, B., 2020. Towards learning convolutions from scratch. Adv. Neural Inf. Proces. Syst. 33, 8078–8088.

Pandey, A., Jain, K., 2022. A robust deep attention dense convolutional neural network for plant leaf disease identification and classification from smart phone captured real world images. Ecol. Inform. 70, 101725.

Ramesh, S., Vydeki, D., 2020. Recognition and classification of paddy leaf diseases using optimized deep neural network with jaya algorithm. Inform. Process. Agric. 7 (2), 249–260.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp. 618–626.

Sun, Y., Jiang, Z., Zhang, L., Dong, W., Rao, Y., 2019. Slic_svm based leaf diseases saliency map extraction of tea plant. Comput. Electron. Agric. 157, 102–109.

Sutaji, D., Yldz, O., 2022. Lemoxinet: lite ensemble mobilenetv2 and xception models to predict plant disease. Ecol. Inform. 70, 101698.

Thai, H.-T., Tran-Van, N.-Y., and Le, K.-H. (2021). Artificial cognition for early leaf disease detection using vision transformers. In 2021 International Conference on Advanced Technologies for Communications (ATC),ss pages 33–38. IEEE.

Thakur, P.S., Khanna, P., Sheorey, T., Ojha, A., 2022a. Attention based twin convolutional neural network with inception blocks for plant disease detection using wavelet transform. In: International Conference on Neural Information Processing. Springer, pp. 308–319.

Thakur, P.S., Khanna, P., Sheorey, T., Ojha, A., 2022b. Trends in vision-based machine learning techniques for plant disease identification: a systematic review. Expert Syst. Appl. 118117.
Thakur, P.S., Khanna, P., Sheorey, T., Ojha, A., 2022c. Vision transformer for plant disease detection. In: International Conference on Computer Vision and Image Processing. Springer, pp. 501–511.
Thakur, P.S., Sheorey, T., Ojha, A., 2022d. Vgg-icnn: a lightweight cnn model for crop disease identification. Multimed. Tools Appl. 1–24.
Thapa, R., Snavely, N., Belongie, S., Khan, A., 2020. The plant pathology 2020 challenge dataset to classify foliar disease of apples. arXiv preprint arXiv:2004.11958.
Too, E.C., Yujian, L., Njuki, S., Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. 161, 272–279.
Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. J. Mach. Learn. Res. 9 (11).
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008.
Yadav, S., Sengar, N., Singh, A., Singh, A., Dutta, M.K., 2021. Identification of disease using deep learning and evaluation of bacteriosis in peach leaf. Ecol. Inform. 101247.
Zeng, W., Li, M., 2020. Crop leaf disease recognition based on self-attention convolutional neural network. Comput. Electron. Agric. 172, 105341.
Zhang, S., Wang, Z., 2016. Cucumber disease recognition based on global-local singular value decomposition. Neurocomputing 205, 341–348.
Zhao, Y., Sun, C., Xu, X., Chen, J., 2022. Ric-net: a plant disease classification model based on the fusion of inception and residual structure and embedded attention mechanism. Comput. Electron. Agric. 193, 106644.