IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 15, 2021, accepted December 30, 2021, date of publication January 7, 2022, date of current version January 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141371

Plant Disease Identification Using a Novel Convolutional Neural Network

SK MAHMUDUL HASSAN AND ARNAB KUMAR MAJI, (Member, IEEE)
Department of Information Technology, School of Technology, North-Eastern Hill University (NEHU), Shillong 793022, India
Corresponding author: Arnab Kumar Maji (arnab.maji@gmail.com)

ABSTRACT The timely identification of plant diseases prevents the negative impact on crops. Convolutional neural network, particularly deep learning is used widely in machine vision and pattern recognition task. Researchers proposed different deep learning models in the identification of diseases in plants. However, the deep learning models require a large number of parameters, and hence the required training time is more and also difficult to implement on small devices. In this paper, we have proposed a novel deep learning model based on the inception layer and residual connection. Depthwise separable convolution is used to reduce the number of parameters. The proposed model has been trained and tested on three different plant diseases datasets. The performance accuracy obtained on plantvillage dataset is 99.39%, on the rice disease dataset is 99.66%, and on the cassava dataset is 76.59%. With fewer number of parameters, the proposed model achieves higher accuracy in comparison with the state-of-art deep learning models.

INDEX TERMS Plant disease, machine learning, deep learning, depthwise convolution, pointwise convolution.

I. INTRODUCTION

Diseases in crops caused mainly by bacteria and fungi negatively impact the production and quality of crops [1]. Timely and early identification of disease symptoms is a major challenge in protecting crops. Visual identification of diseases in a large farm by experts and agronomists is the primary approach in developing countries which is time-consuming and costly. Automatic identification of diseases using smart devices is a promising approach to identifying and reducing overall costs [2]. In recent times, deep learning particularly convolution neural network (CNN) gained much attention in agricultural field such as plant detection [3], fruit detection [4], disease identification [5]–[7], weeds detection [8], pest recognition [9] etc. The reason behind the CNN-based model's popularity is the automatic extraction of appropriate features from the data set. Several popular deep learning-based models such as AlexNet [10], GoogleNet [11], VGGNet [12], ResNet [13], DenseNet [14] have been developed for identification of plant diseases.

Real-time applications and the identification of diseases using deep learning architectures are gaining attention in the current scenario. The number of parameters used and the computation cost in the deep learning models depend on the depth and the number of filters used in the model. The deep learning models usually require a large number of parameters. So the computational cost of the deep learning models is also high. Therefore there will be difficulties to implement on small devices having small resources [15]. In recent times, researchers have implemented deep learning architectures using high-powered devices having GPUs and servers. Uses of sophisticated devices having GPU's are not feasible in the agricultural field due to its high cost. Therefore there is a dearth need for applications having fewer parameters, less power consumption, and computation [16].

In prospect of the above consideration, we have designed a novel lightweight deep learning model to identify the diseases in the plant. This paper proposes a novel CNN architecture based on Inception and ResNet with fewer parameters for determining plant diseases. The Inception architecture extracts better features using multiple convolutions of different filter sizes. To tackle the vanishing gradient problem, we have used residual connections. Instead of standard convolution, we have used depth-wise separable convolution, which reduces the parameter size and the computational complexity without affecting performances. The model has trained on three different datasets, and the performances are

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

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

evaluated. The main contribution of the paper is summarized as follows:

- A new CNN architecture is proposed using Inception and Residual connection, which extracts better features and produces higher performance results.
- In this paper, the standard convolution is replaced with depth-wise separable convolution, which reduces the parameter number by a large margin without affecting performances.
- The proposed architecture uses fewer parameters than other deep learning architecture and is also faster than the standard deep learning models.
- To check the robustness, the model's performance is evaluated on three different plant disease data sets. We have considered three different conditioned images. In the PlantVillage dataset, the images were captured on uniform background and laboratory setup conditions. The images in the rice disease dataset were captured in real-time field conditions, and in the cassava plant dataset, the images are field conditioned, and images contain multiple leaves. We have compared the performance of the proposed model with other state-of-art deep learning models on three different datasets. The result shows that our proposed model outperforms other deep learning models.

The rest of the paper is organized as follows: Section 2 provides the existing literature on the identification of plant diseases using deep learning models. Materials and methods are discussed in Section 3. Section 4 presents the experimental results and discussions. Finally, the paper concludes with Section 5.

II. RELATED WORK

This section presents a discussion and review of recent work on identifying plant diseases based on deep learning models. Mohanty *et al.* [5] used AlexNet and GoogleNet to identify 26 diseases of 14 different plant species. To train the models they have used both transfer learning and learning from the scratch method. They achieved a maximum accuracy of 99.34% using GoogleNet. Five different pre-trained deep learning models, such as VGG, AlexNet, AlexNetOWTBn, Overfeat, and GoogleNet used by Ferentinos [7] to identify 58 plant leaf diseases. Geetharamani and Pandian [17] used nine-layer deep CNN to identify the diseases in plants and achieved an accuracy rate of 96.46%. Inspired from AlexNet and GoogLeNet architecture, Liu *et al.* [18] designed a model by replacing the fully connected layer of AlexNet with the inception layer to identify four different apple leaf diseases and recorded an accuracy rate of 97.62%.

Four different pre-trained deep learning architecture namely VGG16, VGG19, ResNet, and InceptionV3 used by Ahmad *et al.* [19] to identify different tomato leaf diseases. They fine-tuned the network parameter to get the optimal result. They achieved the highest performance accuracy using InceptionV3 is 99.60% and 93.70% on laboratory and field images respectively. Rangarajan and Purushothaman [20] used a pre-trained VGG16 model to identify different eggplant diseases. VGG16 was used for the extraction of features. For classification, they used multiclass SVM. To check the robustness of the model performances, they have used three different color model images, namely RGB, YCbCr and HSV. Using RGB images, they recorded maximum accuracy of 99.4%.

Too *et al.* [21] used several pre-trained deep learning models and fine-tuned the model parameter in the identification and classification of diseases in the plant. They have achieved maximum testing accuracy of 99.75% using DenseNet architecture. Deep feature-based and SVM classifier used by Sethy *et al.* [1] to identify rice leaf diseases. They have used 11 deep CNN models to extract the features and SVM for classification purposes. Deep features from the ResNet50 model with SVM perform better with an f1-score of 98.38%. Six different pre-trained deep learning architecture used by Rangarajan Aravind and Raja [22] to identify 10 different diseases of four varieties of plants. Among the architectures, VGG16 gives the highest performance accuracy of 90% on the test dataset.

Ramacharan *et al.* [23] used transfer learning on InceptionV3 to identify three diseases and two pest damages on cassava plants. The dataset used consists of leaves with multiple numbers of leaves in a single image. The accuracy obtained in the dataset having single leaf images is higher than the images with multiple leaves. Later on, Ramacharan *et al.* [24] used mobile devices to identify six different diseases in the cassava plant. They have used MobileNet deep learning architecture to train the network. They have used both image and video files of diseased leaf images to evaluate the performance. They achieved an accuracy of 80.6% and 70.4% on the image file and video files respectively. Oyewola *et al.* [25] shown that deep learning with residual connection outperforms plain convolution neural network in the identification of different diseases in cassava plants.

Deep residual neural network with 50 layers along with batch normalization and ReLU activation after each convolution used by Picon *et al.* [26] to identify 3 different wheat diseases. They recorded an accuracy rate of 96% on real-time field images captured in Germany. Durmus *et al.* [27] used SqueezeNet architecture to identify different tomato leaf diseases. The structure of SqueezeNet is similar to AlexNet architecture, whose size is 227.6MB, while the size of SequenzeNet is 2.9MB. Modified Cifar10 quick CNN model used by Hu *et al.* [28] and replaced the standard convolution with depthwise separable convolution to identify four different tea leaf diseases. They achieved an improved accuracy rate of 92.5%. Atila *et al.* [29] used EfficientNet architecture to identify diseases in plants and shown that EfficientNet outperforms the standard CNN model such as AlexNet, VGG, etc., with an accuracy rate of 99.97% using EfficientNetB4 on PlantVillage dataset. The EfficientNet model takes less time to train the network as the network uses fewer parameters compared to other deep learning models.

Pre-trained VGG model with inception layer termed as INC-VGGN used by Chen *et al.* [6] to identify the different corn and rice diseases. They replaced the fully connected layers of VGGNet and added two inception layers. The average testing accuracy obtained in rice diseases is 92% and 80.38% in corn diseases. Shallow CNN from the pre-trained VGG16 model used by Li *et al.* [30] to identify different diseases in corn, apple, and grape. They have used only the first four-layer of VGG16 architecture a global pooling layer. Shallow CNN was used to extract the features and PCA to reduce the dimensions. They obtained an f1-score of 94% using SVM and RF respectively. Self-attention CNN was used by Zeng and Li [31] to identify different crop diseases. Attention network is effective in extracting the image features from the critical region. Table 1 summarizes the related works along with the deep learning models and performance results.

III. MATERIALS AND METHODS

A. CONVOLUTION NEURAL NETWORK

Convolutional Neural Network (CNN) is a type of Neural network that is very effective in several computer vision tasks such as pattern recognition and classification. CNN has an advantage that it can learn and extracts the features automatically from the training images whereas in traditional approach there is a need of manual extraction of feature from the images. CNN consists of different layers: Convolutional layer, pooling layer, and fully connected layer. The convolutional layer is the primary and most crucial component of CNN which extracts the feature from the input image. Convolution layers consist of a small array of numbers called kernels applied over the input which produces an output called feature maps. Different convolutional kernels are used to extract different types of features. Several convolutional layers are there, which depend on the size of the input images.

After the convolutional layer, pooling is performed which is responsible for reducing the dimension of the convolutional feature map. The pooling layer performs down sampling operations by reducing the dimension of the feature map, which ultimately helps in reducing the required computational complexity to process the data. Different types of pooling operations are there, such as max-pooling, min-pooling, average-pooling. The output feature maps of the convolution or pooling layer are transformed into a one-dimensional vector in which every input is connected to every output by weight. This layer is also called a dense layer. There can be one or more fully connected layers, and the final fully connected layer has the same number output as the number of classes.

B. RESIDUAL NETWORK

The convolutional neural network may achieve high performances in classification tasks. With the rise in network depth, the performance accuracy gets saturated and degrades rapidly.

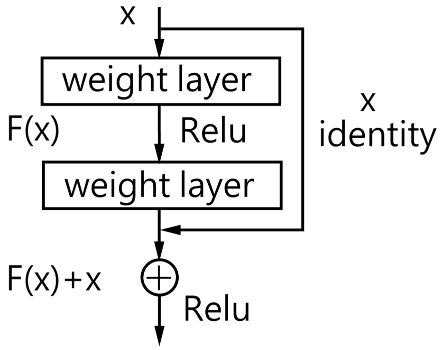


FIGURE 1. Basic block diagram of residual network.

To address this issue in 2015, He *et al.* [13] introduced a deep residual learning network known as ResNet. In deep learning, using the residual connection in the network, we can train a large network and parallely solve the vanishing gradient problem, which usually occurs due to the increase in network depth. Figure 1 shows the basic block diagram of the ResNet model. ResNet introduced a skip connection known as "identity mapping" which combines previous layer's output with the forthcoming layer. To perform identity mapping, it doesn't generate any parameter. Therefore ResNet has the ability to train a deeper network with lower complexity than other networks such as VGG.

C. DEPTHWISE SEPARABLE CONVOLUTION

Depthwise separable convolution introduced by Chollet [34] in the Xception model. Later on, Howard *et al.* [35] used depthwise separable convolution in MobileNet architecture. Depthwise separable convolution factorizes standard convolution into depthwise convolution and 1 × 1 pointwise convolution. In standard convolution, it takes one step to filter the input images and combine these values. Figure 2 shows the block diagram of depthwise separable convolution. The Depthwise layer performs filtering operation, and the pointwise layer combines the output of the depthwise layer. The computation cost of depthwise separable convolution is computed as -

$$D_K^2 \times M \times D_F^2 + M \times N \times D_K^2 \tag{1}$$

whereas the computation cost in standard convolution is -

$$D_K^2 \times M \times N \times D_F^2 \tag{2}$$

where D_F is the dimension of the input image of square size, D_K is the dimension of the kernel, M is the number of channels and N is the number of kernel/filters.

D. PROPOSED NOVEL CNN APPROACH FOR IDENTIFICATION OF PLANT DISEASES

In this paper, we have proposed a novel light weight CNN based on Inception and Residual connections with fewer parameters than the InceptionV3, ResNet50 as well as other

TABLE 1. Summary of related works on identification of plant diseases.

Author	Deep learning model	Dataset	Results
Mohanthy et al.(2016) [5]	Alexnet, GoogleNet	PlantVillage	99.35%(Training acc.)
Sladojevic et al.(2016) [32]	Fine tune CNN	Own	96.3%(Precision)
Fuentes et al.(2017) [33]	Faster R-CNN	Own	83%(Testing acc)
Ferentinos et al.(2018) [7]	Multiple CNN	PlantVillage	99.53%(Training acc.)
Geetharamani et al. (2019) [17]	Nine Layer CNN	PlantVillage	96.46%(Testing acc.) 97.87%(Validation acc.)
Too et al. (2019) [21]	Multiple CNN	PlantVillage	99.75%(Testing acc.)
Sethy et al.(2020) [1]	Multiple CNN with SVM	Rice disease	98.38%(f1-score)
Zeng et al.(2020) [31]	Self Attention CNN	MK-D2 AES-CD9214	95.33%(Testing acc.) 98%(Testing acc.)
J Chen et al.(2020) [6]	INC-VGGN	PlantVillage	97.57%(Training acc.) 91.83%(Validation acc.)
Yan Li et al.(2020) [30]	Shallow CNN	PlantVillage	94%(Precision) 94%(Recall) 94%(f1-score)
Oyewola et al.(2021) [25]	Deep residual CNN	Cassava	96.75%(Testing acc.)

deep learning models. The Inception in deep learning architecture was introduced by Szegedy *et al.* [11] in 2015 and named as GoogleNet (Inception-v1). Later on, additional factorization is added in the inception network and referred it as Inception-v3. Inception architecture is different from typical convolutional architecture and performs multiple convolutions and pooling operations simultaneously to extract the better feature. It concatenates the outputs of all the convolutions of different filter sizes. Inception v3 consist of numbers of Inception A, reduction A, Inception B and Reduction B blocks as shown in Figure 3(a) and Figure 4(a). The inception A block consist of a convolution layer with kernel size 1×1, 1×1 convolution followed by 3×3 convolutions, 1×1 convolution followed by 5×5 convolutions, 3×3 max-pooling layer followed by 1×1 convolution. Inception B block consists of convolution layer with kernel size 1×1, 1×1 convolution followed by 7×7 convolutions, 1×1 convolution followed by two 7×7 convolutions, 3×3 max-pooling layer followed by 1×1 convolution. 1×1 convolutions were used to reduce the computation of the model. The reduction A block uses a 3×3 max-pooling layer, 3×3 convolution layer, and 1×1 convolution layer followed by a 3×3 convolution layer. The reduction B block of inceptionv3 uses a 3×3 max-pooling layer, 3×3 convolution layer followed by 1×1 convolution layer, and 7×7 convolution layer followed by 3×3 convolution layer. Figure 3(a) and Figure 4(a) represent the block diagram of inception-A and inception-B block of inception-V3 architecture.

In our work, we have proposed a novel CNN model which uses the inception architecture with normal convolution as well as depthwise separable convolutions. In the proposed model, 3×3 convolution of Inception A block is replaced by 3×3 depthwise separable convolution (3×3 depthwise convolution and 1×1 pointwise convolution), and 5×5 convolution is replaced with two 3×3 depthwise separable convolutions as shown in Figure 3(b). The 7×7 convolution used in Inception B is replaced with 7×7 depthwise separable convolution (7×7 depthwise convolution and 1×1 pointwise convolution) as shown in Figure 4(b). The 3×3 convolution layer of reduction A replaced with a 1×1 convolution layer followed by 3×3 depthwise separable convolution. In reduction B block, we have replaced the 3×3 and 7×7 convolution with a 1×1 convolution layer and a 3×3 depthwise separable convolution layer. Table 2 and Table 3 shows the parameter comparison between the original inception-A block and the modified inception-A block with depthwise separable convolution. From Table 2 and Table 3 it is seen that the parameter used in the modified inception-A block is much less as compared to the original inception-A blocks.

Figure 5 shows the proposed CNN architecture used to identify the diseases in plants. The proposed implemented model consists of a convolution layer, batch normalization and activation layer, inception blocks with depthwise separable convolution layer, pooling layer, and fully connected layer. In this architecture, we have replaced the standard convolution with depthwise separable convolution. We have used one standard convolution, three depthwise separable convolutions, two max-pooling, and one global average pooling operation, three modified inception A block with residual connection followed by modified reduction A block, three modified inception B block with residual connection followed by modified reduction B block. After each convolution layer, we have used Batch Normalization and activation. The activation function used is ReLu. Batch Normalization and activation function improves the performance and speeds up the process. After global average pooling, we have used dropout, which reduces the chances of overfitting the model. The required parameter in our proposed model is 428,100 while the number of parameter used in standard inceptionv V3 model is 23,851,784 which is much higher than the proposed model. It is observed that the proposed model uses 70% less parameter as compared to the inception V3 architecture.

IV. RESULTS AND DISCUSSION

A. DATASET

In this paper, we consider three different plant disease datasets to evaluate the performances of the model. We have used the Rice plant dataset [36], cassava plant dataset [25],

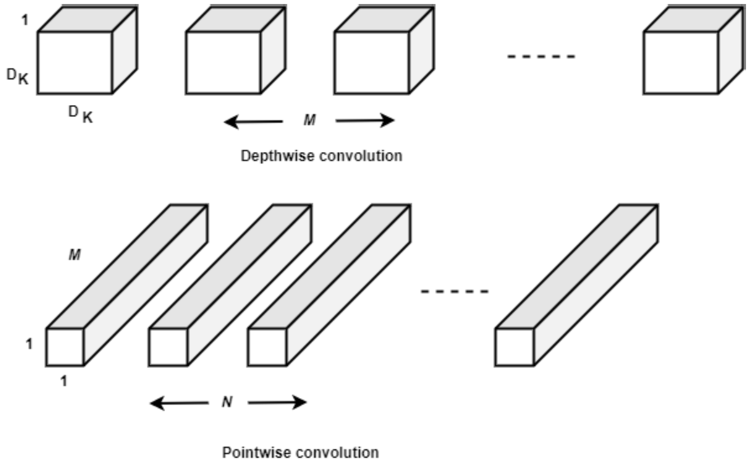


FIGURE 2. Depthwise separable convolution.

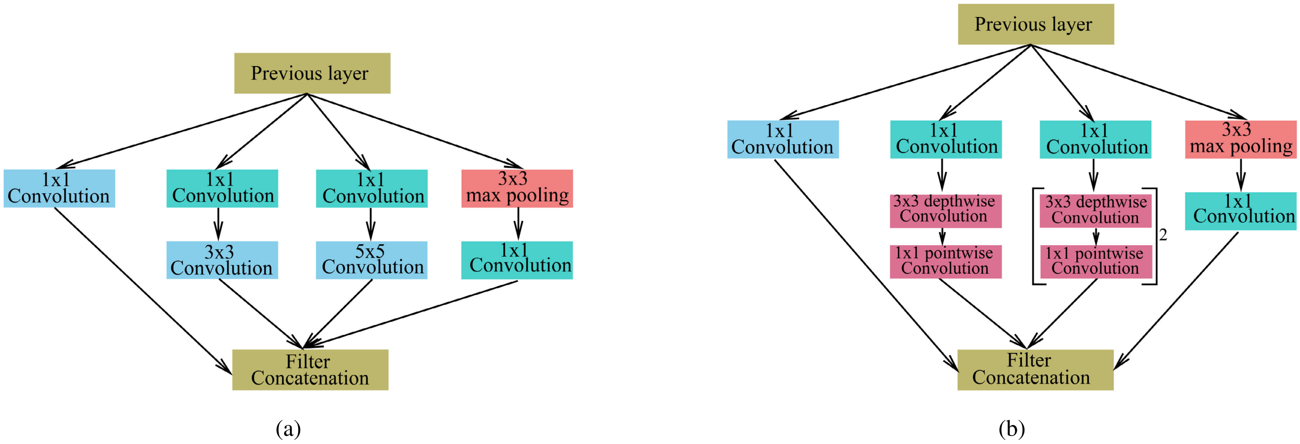


FIGURE 3. Inception architecture of (a) Original inception-A block (b) Modified inception-A block.

and Plant village dataset [5] which is the large and available plant disease dataset. From the Plant village dataset specifically, we have used the corn, potato, and tomato plant diseases. It is seen that the images in the PlantVillage dataset are captured in a uniform background, and the intensity is also relatively uniform. The rice plant dataset consists of 5932 images which are divided into four categories, namely 1584 bacterial blight, 1440 blast, 1600 brown spot, and 1308 tungro images. The cassava plant dataset consists of 5,656 images includes five different categories as 316 healthy cassava leaf images and four sets of diseased leaves as 466 cassava bacteria blight, 1,443 cassava brown streak, 773 cassava green mite, and 2,658 cassava mosaic diseased images. The images in the cassava dataset are captured in the complex background, and multiple leaves are also present in a single image. The datasets are divided randomly into training and testing set in the ratio of 80:20. The dimensions of the images are resized into 256 × 256 pixels. Figure 6 shows the sample images of different datasets. Table 4, Table 5, Table 6 shows the dataset details along with the disease common name, disease scientific name, Type of disease along with the number of images in each class.

B. EXPERIMENTAL RESULT

To evaluate the performance of the model, we consider different performance statistics such as number of parameter, accuracy, precision, recall, f1-score and expressed as follows-

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{6}$$

where TP = true positive, TN = true negative, FP = false positive and FN = false negative.

Table 7 shows the performances of the implemented model on different datasets. From Table 7, it is seen that after 50

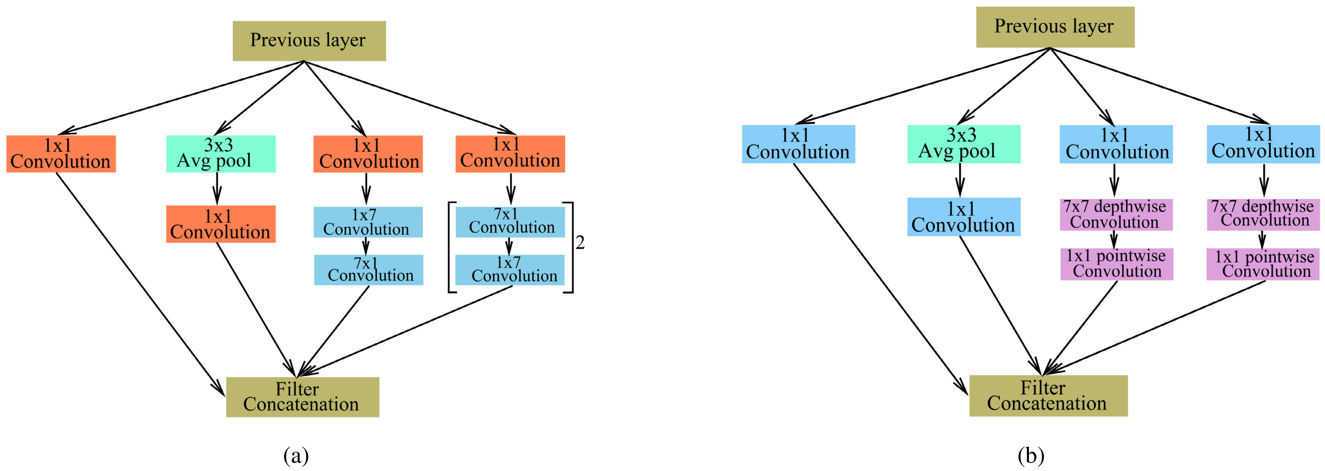


FIGURE 4. Inception architecture of (a) Original inception-B block (b) Modified inception-B block.

TABLE 2. Parameter required in original inception-A block.

Layer	Input	Filter	Stride	Parameter
Input Image	256 × 256	-	-	-
Conv 1	Input image	1 × 1/32	2,2	128
Conv 2	Conv 1	1 × 1/64	-	2112
Conv 3	Conv 1	1 × 1/48	-	1584
Conv 4	Conv 3	3 × 3/32	1,1	13,856
Conv 5	Conv 1	1 × 1/48		1584
Conv 6	Conv 5	5 × 5/32	1,1	38,432
Avg. Pool	Conv 1	3 × 3	1,1	-
Conv 7	Avg. pool	1 × 1/32		1056
Concatenation	Conv 2 Conv 4 Conv 6 Conv 7	-	-	-
Total				58,752

TABLE 3. Parameters required in modified inception-A block with depth-wise separable convolution.

Layer	Input	Filter	Stride	Parameter
Input Image	256 × 256	-	-	-
Conv 1	Input image	1 × 1/32	2,2	128
Conv 2	Conv 1	1 × 1/64	-	2112
Conv 3	Conv 1	1 × 1/48	-	1584
Depthwise separable conv1	Conv 3	3 × 3/32	1,1	2000
Conv 4	Conv 1	1 × 1/48	-	1584
Depthwise separable conv2	Conv 4	3 × 3/32	1,1	2000
Depthwise separable conv3	Depthwise separable conv2	3 × 3/32	1,1	1344
Avg. Pool	Conv 1	3 × 3	1,1	-
Conv 7	Avg. pool	1 × 1/32		1056
Concatenation	Conv 2 Conv 4 Conv 6 Conv 7	-	-	-
Total				11,808

epochs of training, the model achieved the highest training accuracy and loss of 99.81% and 0.0015, validation accuracy and loss of 99.39% and 0.0549 respectively in PlantVillage dataset (potato, corn, tomato). In the rice dataset, the model achieves the highest training accuracy of 99.94%, training loss of 0.0030, validation accuracy of 99.66%, and validation loss of 0.0041. To evaluate the robustness of our proposed model, we have considered the Cassava plant disease dataset in which the images were captured in real-time and in field conditions. Moreover, the images have complex backgrounds

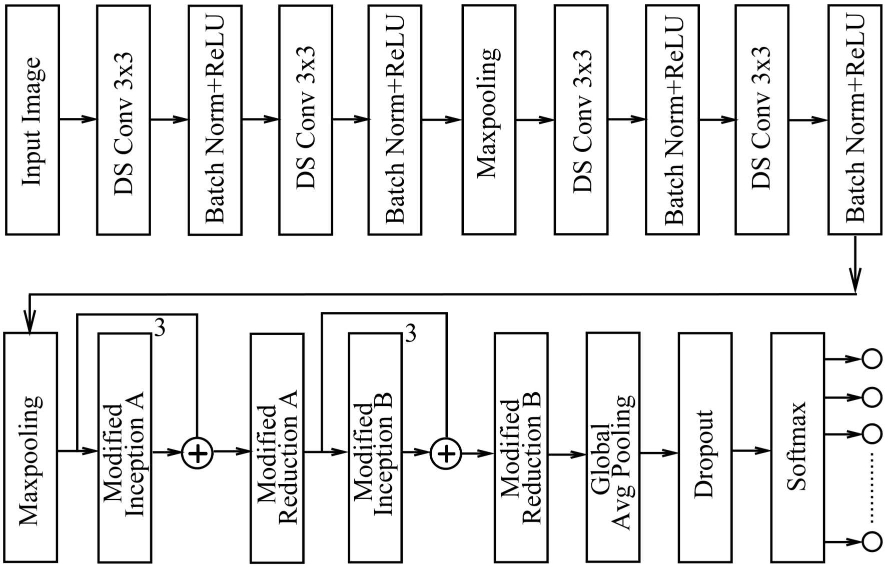


FIGURE 5. Proposed CNN approach in identification of plant diseases.

TABLE 4. Data description of PlantVillage dataset (corn, potato, tomato).

Class	Plant Name	Disease Name	Disease Scientific Name	Type of Disease	No of Image
C1	Corn	-	-	-	1162
C2	Corn	Grey leaf spot	Cercospora zeae-maydis	Fungal	513
C3	Corn	Common rust	Puccinia sorghi	Fungal	1192
C4	Corn	Northern leaf blight	Setosphaeria turcica	Foliar	985
C5	Potato	-	-	-	182
C6	Potato	Late Blight	Phytophthora infestans	Fungal	1000
C7	Potato	Early Blight	Alternaria solani	Fungal	1000
C8	Tomato	-	-	-	1591
C9	Tomato	Bacterial spot	Xanthomonas campestris pv. Vesicatoria	Bacterial	2127
C10	Tomato	Early blight	Alternaria solani	Fungal	1000
C11	Tomato	Late blight	Phytophthora infestans	Fungal	1909
C12	Tomato	Septoria leaf spot	Septoria lycopersici	Fungal	1771
C13	Tomato	Spider mites	Tetranychus urticae	Pest	1676
C14	Tomato	Tomato mosaic virus	Tomato mosaic virus (ToMV)	Viral	373
C15	Tomato	Leaf Mold	Fulvia fulva	Fungal	952
C16	Tomato	Target spot	Corynespora cassiicola	Fungal	1404
C17	Tomato	YLCV	Begomovirus (Fam. Geminiviridae)	Viral	5357

TABLE 5. Data description of rice disease data.

Class	Plant Name	Disease Name	Disease Scientific Name	Type of Disease	No of Image
C1	Rice	Bacterial blight	Xanthomonas oryzae pv. oryzae	Bacterial	1584
C2	Rice	Blast	Magnaporthe grisea	Fungal	1440
C3	Rice	Brown Spot	Cochliobolus miyabeanus	Fungal	1600
C4	Rice	Tungro	tungro bacilliform	Viral	1308

and having multiple leaves present on a single image. Compared with the PlantVillage and Rice plant datasets, the Cassava plant dataset gives less training and validation accuracy. The obtained training and validation accuracies are 98.17% and 76.59%, respectively. The cassava dataset's performance accuracy drops compared to the plantvillage and rice dataset due to data imbalance, and the images in the dataset contain images with complex backgrounds.

After splitting the dataset into 80%-20% training and validation set we train the model upto 50 epoch and evaluate the performances on training and validation set. Figure 7-9 shows the accuracy and loss for training and validation of the proposed model on PlantVillage, rice and cassava dataset respectively. From Figure 7-9 it is seen that the proposed model gives satisfactory performances with fewer number of parameters.

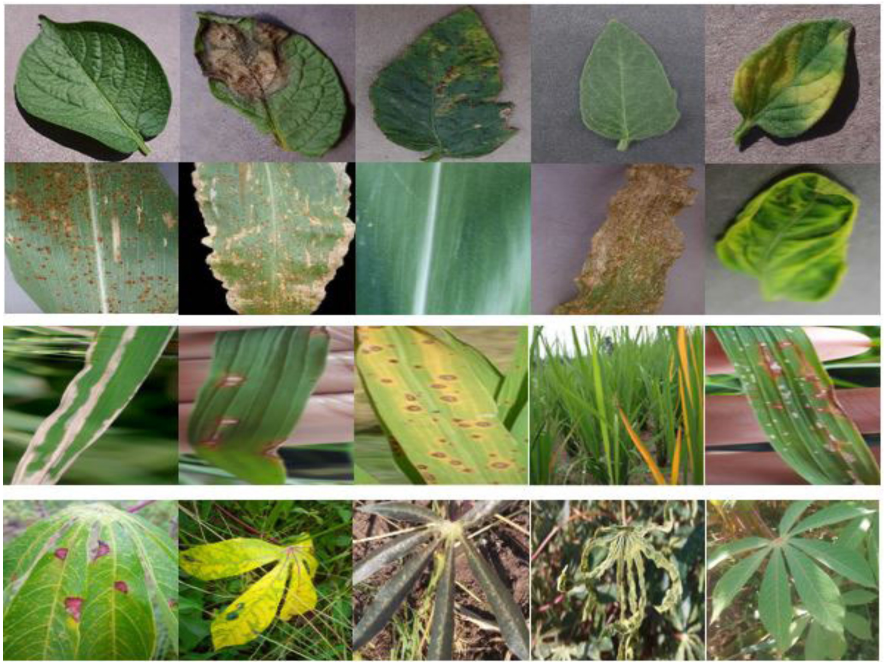


FIGURE 6. Sample images of Plantvillage dataset, rice and cassava plant dataset.

TABLE 6. Data description of cassava dataset.

Class	Plant Name	Disease Name	Disease Scientific Name	Type of Disease	No of Image
C1	Cassava	-	-	-	316
C2	Cassava	Bacterial blight	Xanthomonas axonopodis pv. manihotis	Bacterial	466
C3	Cassava	Brown streak	Cassava brown streak viruses	Bacterial	1443
C4	Cassava	Green mite	Mononychellus tanajoa	Pest	773
C5	Cassava	Mosaic disease	Cassava mosaic disease	Viral	2658

TABLE 7. Summary of deep learning based implemented methods.

Dataset	30 epochs				50 epochs				Training time (sec/epoch)	Testing accuracy (%)
	Training acc	Training loss	Val acc	Val loss	Training acc	Training loss	Val acc	Val loss		
PlantVillage	0.9981	0.0273	0.9891	0.0864	0.9973	0.0015	0.9939	0.0549	883	0.9939
Rice	0.9970	0.0085	0.9823	0.0452	0.9994	0.0030	1.00	0.0041	227	0.9966
Cassava	0.9545	0.1201	0.6793	0.6732	0.9817	0.0508	0.7659	0.4465	221	0.7659

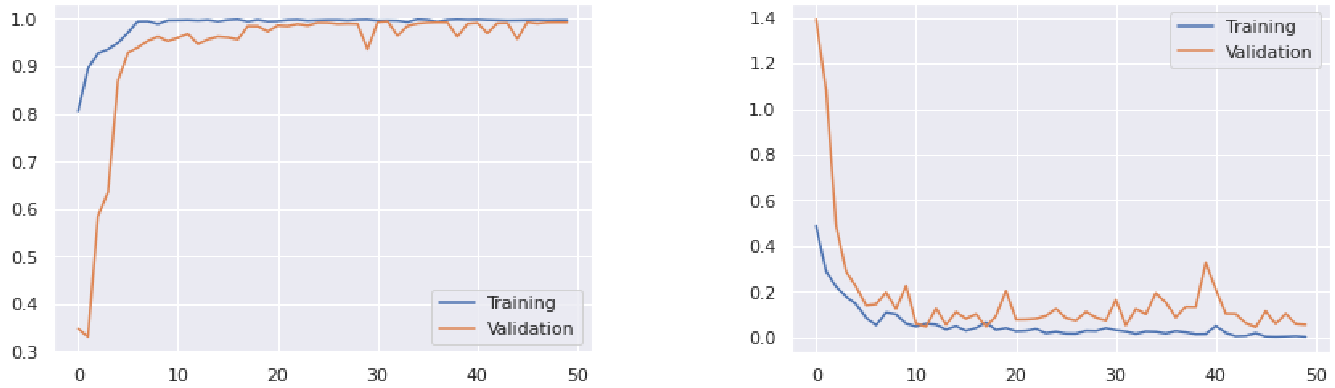
We have also calculated the confusion matrix to evaluate the performance of the model. Table 8 shows the performances of the implemented model in terms of testing accuracy, precision, recall, and f1-score on three plant datasets. From Table 8, it is seen that the Rice plant dataset gives the highest testing accuracy than PlantVillage and Cassava dataset. Figure 10 gives the performance metrics of the implemented model on different datasets. From Figure 10, it is seen that the average prediction accuracy, precision, recall, and F1-score is more than 99% on the rice and plantvillage dataset.

C. COMPARISON OF PERFORMANCES AND ROBUSTNESS OF THE MODEL

To validate the stability of our proposed CNN model, we use k-fold cross-validation (5 fold) to evaluate the performances on the disease datasets. The disease datasets are divided randomly into five equal parts. Each part of the dataset is considered as the test set, and the remaining part of the dataset is considered as the training set. Based on the splitting of the dataset, we get the performance of the model on a different set of training and testing images. Table 9 shows the performances of the proposed model on each fold. From Table 9, it is seen that there is not much performance deviation in each dataset. In the PlantVillage dataset, the accuracy ranges from 99.29% to 99.37%, in the Rice dataset, it ranges from 99.33% to 99.66% and in the Cassava dataset, the accuracy ranges from 76.42% to 76.58%. The result shows that our proposed CNN method has good stability in the identification of diseases and is consistent in each data split part.

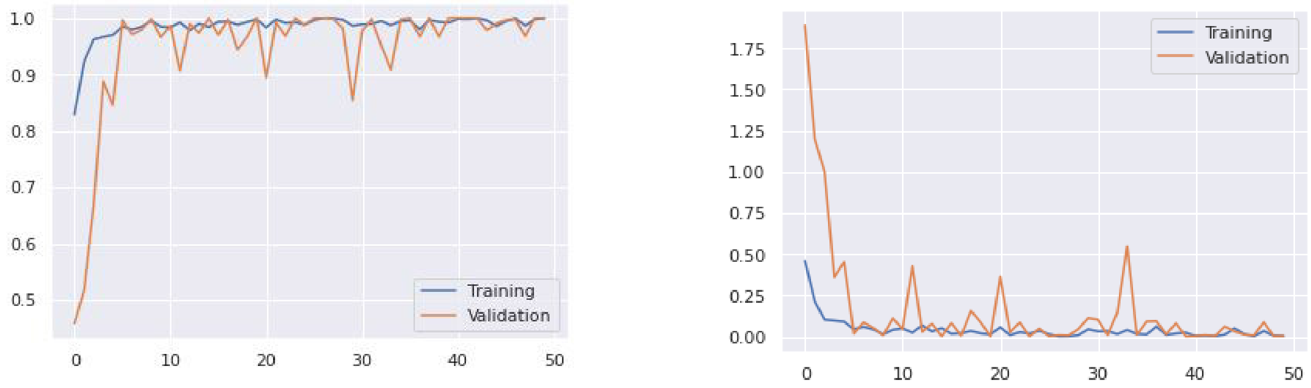
D. PERFORMANCE COMPARISON WITH PRE-TRAINED NETWORK

We have evaluated the performances of several pre-trained deep learning models such as VGG16, VGG19, InceptionV3,



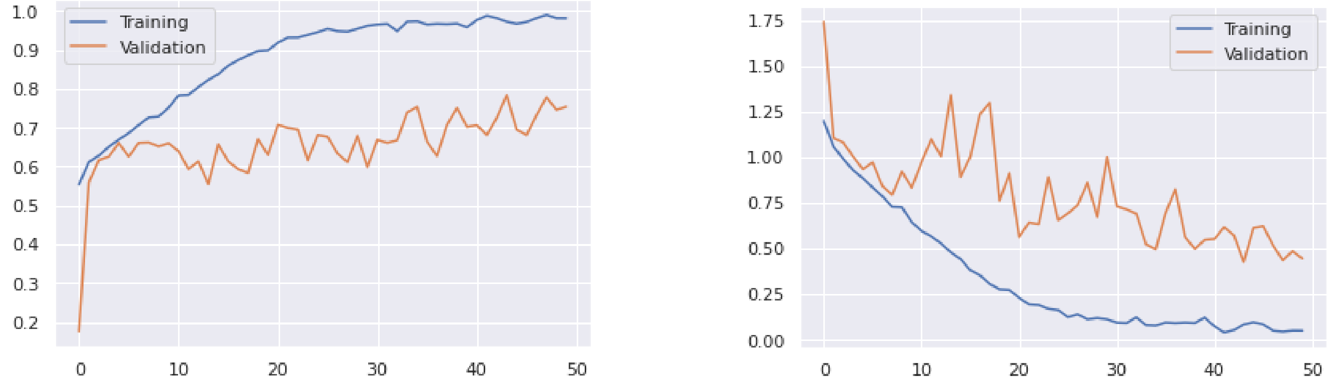
(a) (b)

FIGURE 7. (a) Training & validation accuracy (b) Training & validation loss on PlantVillage dataset.



(a) (b)

FIGURE 8. (a) Training & validation accuracy (b) Training & validation loss on rice disease dataset.



(a) (b)

FIGURE 9. (a) Training & validation accuracy (b) Training & validation loss on cassava plant dataset.

ResNet50, and DenseNet201 on three different datasets. The performance is compared with our proposed CNN model in terms of performance accuracy and training time. The performance comparison with other pre-trained deep learning models on three different datasets is shown in Table 10. The accuracy defines the correctly identified classes of the images

TABLE 8. Performance metric of the proposed model on testing images.

Dataset	Accuracy	Recall	Precision	F1-score
Rice Dataset	99.66	99.67	99.66	99.67
PlantVillage	99.39	99.19	99.17	99.18
Cassava	76.59	72.63	62.03	66.91

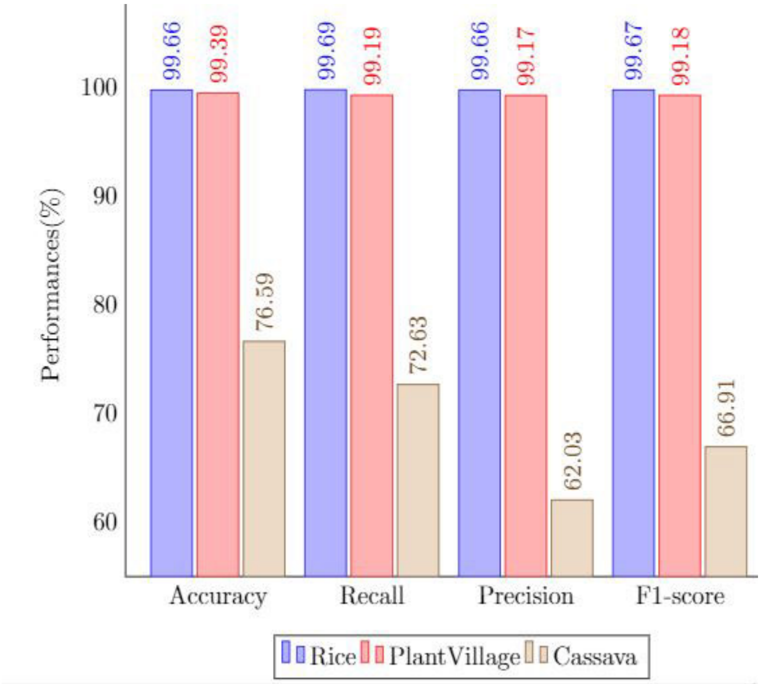


FIGURE 10. Performance metric of proposed model on different dataset.

from the test image set. From the performance results, it is seen that our proposed model gives satisfactory performances on three different datasets (99.39% on PlantVillage, 99.66% on Rice, 76.59% on imbalance cassava) with fewer parameters (428,100). Table 10 shows the time required to train the models per epoch on three different datasets. From Table 10, it is seen that the required training time is much less in comparison with the other pre-trained networks in all three datasets as the number of layer used in our proposed model is less. The performance accuracies of the pre-trained deep learning models are less as the models uses the pre-trained weight where the models are trained on Imagenet Dataset. Due to the uses of pre-trained weight, these models did not achieve the optimal results. Whereas, the proposed model has an advantages of inception layer which has ability to extract better features as well as residual connection that removes vanishing gradient problem. Moreover, the uses of batch normalization and dropout increases the performance of model. Although the proposed model uses depthwise separable convolution which reduces the model parameter, also the number of layers used in this model is less. So the required time to train the model is much less as compared to pre-trained models. DenseNet201 requires more training time than the other deep learning models as the number of layers are more.

E. COMPARISON WITH SOME EXISTING LITERATURE

To verify the performances of our proposed CNN model, we compare the performances of our proposed model with other deep learning models from the literature. Table 11 shows the performance comparison with some deep learning models. From Table 11, it is seen that the proposed CNN model performs better. From Table 11, we can also see that the proposed model requires much fewer parameter than other deep learning models. As the model uses fewer parameters, it requires less time to train the model compared to other deep learning models. From Table 11 it is also seen that the models were implemented on PlantVillage dataset where the images are captured in controlled environment. In this work, we have used three different dataset where rice disease images were captured in field condition and contains background image. The cassava dataset consist of field images along with more than one leaf in single image. Our proposed CNN model has advantages in terms of performance accuracy, parameter size, and training time.

TABLE 9. Result of proposed CNN based on k-fold cross validation.

No of	Accuracy		
Fold	PlantVillage	Rice Dataset	Cassava
1-fold	0.9931	0.9966	0.7649
2-fold	0.9929	0.9954	0.7645
3-fold	0.9937	0.9951	0.7642
4-fold	0.9934	0.9933	0.7658
5-fold	0.9931	0.9959	0.7652
Average	0.9932	0.9952	0.7649

TABLE 10. Performance comparison with pre-trained network.

Models	VGG16	VGG19	InceptionV3	ResNet50	DenseNet201	Proposed
Parameter	138,357,544	143,667,240	23,851,784	25,636,712	20,242,984	428,100
Accuracy (PlantVillage)	0.9565	0.9887	0.9727	0.9798	0.9975	0.9939
Training Time (sec/epoch) (PlantVillage)	2135	2463	2742	5475	9626	883
Accuracy (Rice)	0.7931	0.7888	0.7931	0.7888	0.7991	0.9966
Training Time (sec/epoch) (Rice)	490	608	697	1434	2582	227
Accuracy (Cassava)	0.6355	0.6813	0.7143	0.7439	0.7567	0.7659
Training Time (sec/epoch) (Cassava)	477	598	673	1412	2556	221

TABLE 11. Performance comparison with different deep learning models.

Deep learning model	Dataset used	Parameter required	Training acc	Testing acc	Testing loss	Epoch	Training time (sec/epoch)
VGG [7]	PlantVillage	138 million	-	98.87	0.0542	49	4208
AlexNet [5]	PlantVillage	60 million	99.35	93.88	-	30	-
INC-VGGN [6]	PlantVillage	more than 138 million	97.57	91.83	0.2409	30	-
GoogleNet	PlantVillage	7 million	-	97.3	-	20	-
ResNet50+SVM [1]	Rice	23 million	-	98.38	-	-	69.04
Nine layer CNN [17]	PlantVillage	-	97.87	96.46	-	3000	-
Shallow CNN+ RF [30]	PlantVillage	0.26 million	-	94	-	-	-
Deep Residual CNN [25]	Cassava	-	-	58.39	-	80	-
Proposed CNN	PlantVillage	0.42 million	99.73	99.39	0.0549	50	883
Proposed CNN	Rice	0.42 million	99.94	99.66	0.0041	50	227
Proposed CNN	Cassava	0.42 million	98.17	76.59	0.4465	50	221

V. CONCLUSION

Deep learning is a recent and advanced technique in the field of image pattern recognition and much effective in the identification of diseases in plants. In this paper, we have proposed a novel CNN model based on the inception and residual connection that can effectively classify the diseases in plants. In addition, we have used depthwise separable convolution in inception architecture which reduces the computation cost by reducing the number of parameters by a margin of 70%. Therefore, training the network requires much less time as compared to the standard CNN. The experimental result shows that the proposed model achieves higher performance accuracy. To evaluate the robustness of the model, we have used three different plant datasets. The testing accuracies of the proposed model is 99.39%,99.66% and 76.59% on Plantvillage, Rice, and Cassava dataset, respectively. In [25] author recorded an accuracy rate of 52.87% and 46.26% using plain convolution neural network and deep residual neural network respectively on imbalance dataset. In [24] author achieved an accuracy rate of 80.6% on balance dataset. In comparison with [24], [25], our proposed model achieves a higher accuracy rate of 76.59% on the imbalance cassava dataset. For the rice dataset Sethy *et al.* [1] achieved an accuracy rate of 98.25%, and our proposed model achieved higher accuracy of 99.66% on the same dataset. In the future, we carry forward and try to investigate the performance of the proposed model on the different agricultural fields such as weed detection, pest identification, etc. Another future work includes the identification of plant diseases with different plant disease datasets with a different variety of images, different geographical regions. Using clustering based [37] unsupervised technique in identification of diseases also an important aspects.

REFERENCES

[1] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based Rice leaf disease identification using support vector machine," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105527.

[2] 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.

[3] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognit.*, vol. 71, pp. 1–13, Nov. 2017.

[4] G. Farjon, O. Krikeb, A. B. Hillel, and V. Alchanatis, "Detection and counting of flowers on apple trees for better chemical thinning decisions," *Precis. Agricult.*, vol. 21, pp. 1–19, Aug. 2019.

[5] 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.

[6] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105393.

[7] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018.

[8] V. Hoang Trong, Y. Gwang-hyun, D. Thanh Vu, and K. Jin-young, "Late fusion of multimodal deep neural networks for weeds classification," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105506.

[9] F. Ren, W. Liu, and G. Wu, "Feature reuse residual networks for insect pest recognition," *IEEE Access*, vol. 7, pp. 122758–122768, 2019.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[13] 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.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[15] K. Zhang, K. Cheng, J. Li, and Y. Peng, "A channel pruning algorithm based on depth-wise separable convolution unit," *IEEE Access*, vol. 7, pp. 173294–173309, 2019.

[16] M. Tao, X. Ma, X. Huang, C. Liu, R. Deng, K. Liang, and L. Qi, "Smartphone-based detection of leaf color levels in Rice plants," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105431.

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

[18] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, 2018.

[19] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, "Optimizing pretrained convolutional neural networks for tomato leaf disease detection," *Complexity*, vol. 2020, pp. 1–6, Sep. 2020.

[20] A. K. Rangarajan and R. Purushothaman, "Disease classification in eggplant using pre-trained VGG16 and MSVM," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020.

[21] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019.

[22] K. R. Aravind and P. Raja, "Automated disease classification in (Selected) agricultural crops using transfer learning," *Automatika, Časopis Za Automatiku, Mjerenje, Elektroniku, Računarstvo I Komunikacije*, vol. 61, no. 2, pp. 260–272, 2020.

[23] 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.

[24] 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.

[25] D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševičius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," *PeerJ Comput. Sci.*, vol. 7, p. e352, Mar. 2021.

[26] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Comput. Electron. Agricult.*, vol. 161, pp. 280–290, Jun. 2019.

[27] H. Durmus, E. O. Gunes, and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," in *Proc. 6th Int. Conf. Agro-Geoinform.*, Aug. 2017, pp. 1–5.

[28] G. Hu, X. Yang, Y. Zhang, and M. Wan, "Identification of tea leaf diseases by using an improved deep convolutional neural network," *Sustain. Comput., Informat. Syst.*, vol. 24, Dec. 2019, Art. no. 100353.

[29] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Informat.*, vol. 61, Mar. 2021, Art. no. 101182.

[30] Y. Li, J. Nie, and X. Chao, "Do we really need deep CNN for plant diseases identification?" *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105803.

[31] W. Zeng and M. Li, "Crop leaf disease recognition based on self-attention convolutional neural network," *Comput. Electron. Agricult.*, vol. 172, May 2020, Art. no. 105341.

[32] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–11, May 2016.

[33] F. Alvaro, Y. Sook, K. Sang, and P. Dong, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[36] P. K. Sethy, "Rice leaf disease image samples," *Mendeley Data*, vol. 1, 2020, doi: 10.17632/fwcj7stb8r.1.,

[37] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

SK MAHMUDUL HASSAN received the B.Tech. and M.Tech. degrees in information technology from North Eastern Hill University (NEHU), Shillong, India, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in information technology. His research interests include AI, machine learning, and image processing.

ARNAB KUMAR MAJI (Member, IEEE) received the B.E. degree in information science and engineering from Visvesvaraya Technological University (VTU), in 2003, the M.Tech. degree in information technology from Bengal Engineering and Science University, Shibpur (Currently IIEST, Shibpur), in 2005, and the Ph.D. degree from Assam University, Silchar (a Central University of India), in 2016. He has approximately 17 years of professional experience. He is currently working as an Assistant Professor (Stage-3) with the Department of Information Technology, North Eastern Hill University, Shillong (a Central University of India). He has published more than 32 articles in different reputed international journals and conferences, more than 22 articles as book chapter, and authored two books with several international publishers, such as Elsevier, Springer, IEEE, MDPI, IGI Global, and McMilan International. Four Ph.D. scholars are currently pursuing doctoral degrees under his active supervision. He has also guided successfully 15 M.Tech. thesis and three Ph.D. scholars. His research interests include machine learning, image processing, and natural language processing. He is a reviewer of several reputed international journals and a Guest Editor of one Springer journal.