IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 November 2022, accepted 26 December 2022, date of publication 28 December 2022,
date of current version 23 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3232917

RESEARCH ARTICLE

Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network

VIBHOR KUMAR VISHNOI[1], (Member, IEEE), KRISHAN KUMAR[1], BRAJESH KUMAR[2], (Senior Member, IEEE), SHASHANK MOHAN[3], AND ARFAT AHMAD KHAN[4]

[1]Department of Computer Science, Gurukula Kangri (Deemed to be University), Haridwar, Uttarakhand 249404, India
[2]Department of Computer Science and IT, MJP Rohilkhand University, Bareilly, Uttar Pradesh 243006, India
[3]Department of Biosystems and Agricultural Engineering, Michigan State University, East Lansing, MI 48824, USA
[4]Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen 40002, Thailand

Corresponding author: Arfat Ahmad Khan (arfatkhan@kku.ac.th)

This work was supported by the Department of Computer Science, College of Computing, Khon Kaen University, Thailand.

ABSTRACT Plant diseases are a severe cause of crop losses in the agriculture globally. Detection of diseases in plants is difficult and challenging due to the lack of expert knowledge. Deep learning-based models provide promising ways to identify plant diseases using leaf images. However, need of larger training sets, computational complexity, and overfitting, etc. are the major issues with these techniques that still need to be addressed. In this work, a convolutional neural network (CNN) is developed that consists of smaller number of layers leading to lower computational burden. Some augmentation techniques such as shift, shear, scaling, zoom, and flipping are applied to generate additional samples increasing the training set without actually capturing more images. The CNN model is trained for apple crop using a publicly available dataset PlantVillage to identify Scab, Black rot, and Cedar rust diseases in apple leaves. The rigorous experimental results revealed that the proposed model is well fit to identify apple leaf diseases and achieves 98% classification accuracy. It is also evident from the results that it needs lesser amount of storage and takes smaller execution time than several existing deep CNN models. Although, there exist several CNN models for crop disease detection with comparable accuracy, but the proposed model needs lower storage and computational resources. Therefore, it is highly suitable for deploying in handheld devices.

INDEX TERMS Apple diseases, classification, convolutional neural network, deep learning, disease detection, image processing, machine learning.

I. INTRODUCTION

Agribusiness is one of the essential sources of the subsistence of people for many decades. It contributes in the nourishment of about fifty percent of the global population [1]. Agriculture has an immense impact on people globally, either directly or indirectly. Agriculture production needs to rise by 50-60% to ensure the food security in coming years, especially for countries having rapid growth of population [2]. Horticulture contributes around 30% to the GDP of the Indian agriculture industry [3]. *Apple* is one of the most widely consumed fruit globally and among the four most produced fruits after *banana, grape*, and *orange* [4]. The production of *apple* crop has increased in the last decade, but it is not in proportion to the growth of the cultivation area. In India, the cultivation area of *apple* crop is raised by 20%, although the production only increased by 1-2% [5].

Globally, pests and diseases affect the overall production of apple crop. In India, fungal diseases are one of the major causes affecting the quality of apple fruits in Himachal Pradesh, which is the second highest producer state of apple fruits [4]. Infections in plants are categorized into two categories- biotic and abiotic [6]. The pathogens such as virus, fungi, and bacteria are the infectious agents responsible for biotic diseases. The biotic diseases are highly transmissible and dangerous comparable to the abiotic diseases that are

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

TABLE 1. Description of apple diseases.

Disease	Disease Type	Responsible Pathogen	Symptoms
Scab	Fungal	Venturia inaequalis	Light green spots on leaf, then grows like velvety or olive-colored
Black rot	Fungal	Diplodia seriata	circular purplish spots (frog-eye) on leaves
Cedar rust	Fungal	G. juniperi-virginianae	Small pale yellow spots on leaves

caused by physiological factors such as mineral deficiency, sunburn, other environmental factors. *Scab, Cedar rust, leaf blotch, Powdery mildew, blight, Mosaic*, and *Black rot* are some common biotic diseases visible in apple leaves. Diseases such as *Scab, Cedar rust*, and *Black rot* are emphasized to identify. A brief description of these diseases is given here.

1) *Black rot* is a fungal infection caused by *Diplodia seriata* fungus. It develops small sneaks on the upper surface of the leaf during leaf unfolding. Later, frog-eye spots with reddish or purplish edges appear on the infected leaves. As the lesion ages, it becomes chlorotic and higher severity leads to defoliation weakening the tree. Black rot affects both fruits and leaves of the tree.
2) *Scab* is a serious infection transmitted through *Venturia inaequalis* fungus. Disease symptoms appear as pale-yellow or olive-green spots on the upper surface and as velvety or dark lesions on the lower surface of the leaf. As the infection spreads, it causes leaves to curl up or drop. The severe infection leads to continuous defoliation and damage to the tree. Scab also affects both fruits and leaves of the tree.
3) *Cedar rust* is also a fungal infection caused by *Gymnosporangium juniperi-virginianae* fungus. Initially, the reddish or pale yellow circular lesions appear on leaves' upper surface and gradually enlarge into bright orange-yellow spots. Severe infection results in pale yellow or orange spots on fruits, and an unseasonable fall of leaves.

It is essential to increase production capacity by timely detecting and quickly diagnosing plant diseases. So that the food security standards can be maintained globally. Fluorescence in-situ hybridization (FISH), polymer chain reaction (PCR), immunofluoresence (IF), flow cytometry are the common molecular or serological method-based traditional techniques used for disease detection/identification [6]. These techniques can be used by the experts or phytopathologist only inside the laboratories. Besides, the optical observation of diseases in leaves is a complex way that is error prone also. The common advantages and limitations of these techniques are listed in Table 2. The visual characteristics of the infected parts such as leaves, roots, etc. are usually affected significantly. Some specific visible patterns are generated by responsible pathogen for most of the infections. Advancements in image processing, artificial intelligence, and computational resources like graphical processing unit (GPU) can reform the process of detection and prevention of plant diseases [7].

Machine Learning (ML) based approaches such as k-nearest neighbor (KNN), artificial neural networks (ANN), and support vector machine (SVM) are some most commonly used techniques in plant disease detection/ identification. Recently, deep learning techniques emerged to provide a better hierarchical feature learning approach. It represents well both low-level and high-level features with some added complexity [8]. The convolutional neural network (CNN) provides a solution to many complex problems such as natural language processing (NLP), object detection, image classification, speech recognition. In recent years, CNNs have proved their suitability for detecting the plant diseases. However, processing a large dataset through CNN models leads to the requirement of high computational resources. Therefore, it is a challenging job to deploy such models in handheld devices such as smart phones to aid the farmers. The motivation behind this work is to develop a light-weight deep CNN model leading to low computational burden in diagnosing the diseases efficiently. In this work, a novel deep CNN model consisting of three convolutional layers (Conv-3 DCNN) is developed to diagnose three diseases in *apple* plants using leaf images. It makes use of data augmentation to enhance the performance of the model. The hyperparameters are tuned using random search technique that helps to select the best suitable hyperparameters. The proposed model is compared with the recent transfer learning approaches such as VGG-19 [9], ResNet-152 [10], DenseNet-201 [11], MobileNetV2 [12], ResNet-50 [10], VGG-16 [9], InceptionV3 [13], Xception [14], and MobileNet [15]. The major contribution of this paper is the development of a novel light-weight deep CNN model for detecting apple leaf diseases with less computational burden.

The organization of the remaining paper is as follows: Section II presents the related work. The materials and methodologies are introduced in Section III which consists of the theoretical background, dataset used, and the outlines of proposed method. The experimental setup, performance matrices, results, and the related discussions are covered in Section IV. At last, it is concluded with remarks on future research scope in the Section V.

II. RELATED WORK

Image processing and ML have the potential to aid farmers or orchardists in detecting plant diseases in computationally intelligent ways instead of manual checkups.The robust foundation of ML for decision-making based on systematically integrating expert knowledge is very promising in many

TABLE 2. Traditional techniques to detect/ identify plant diseases.

Technique	Advantages	Limitations
PCR	Ease of use, portable, and a mature technique for detection	Based on the effectiveness of extracted DNA, PCR buffer, and polymerase activities
ELISA	Detection can be performed based on change in visual colors, and a low cost technique	Less accurate for bacteria
FISH	Molecular detection technique and depends on DNA prob, plant gene	Photobleaching causes false negative results, accuracy highly depends on nucleotide probes, and specificity disturbed by autofluorescence materials
FCM	Fast detection, simultaneous evaluation of several parameters	Detects bacterial infections only, complex and High Cost
IF	distribution of target molecule can be visualized, Highly sensitive	Photobleaching causes false negative results

areas, including agriculture procedures. SVM is capable of separating non-linear data using hyperplanes. Therefore, it is one of the most widely applied ML approach in plant disease detection. In [16], authors used SVM to investigate *tomato* crop health status. They considered the texture characteristics as a significant feature to describe the tomato leaf and proposed a system that uses gray-level co-occurrence matrix (GLCM) with SVM algorithm. The similar work was done by Deshapande et. al. [17] to identify fungal disease in *maize* crop. They also used GLCM based texture features with SVM classifier and achieved 88% classification accuracy. Authors in [18] and [19], also used texture features of disease lesions in recognizing the diseases in *cucumber* using SVM classification technique. In a study, Chakraborty et al. [20] employed multiclass SVM on texture features to predict disease in *apple* leaves with accuracy to 96%. The feature set was created using GLCM, which provides texture structure of disease lesions. Vishnoi et al. [21] also used texture information to identify diseases in *urdbean* and *apple* crops using SVM. Authors in [22] used texture characteristics with SVM to check health status of *palm* trees. They achieved 95% classification accuracy. In [23], authors performed an investigation on fungal infections in *soybean* crop using SVM and achieved an average accuracy of ∼86% on PlantVillage dataset with texture properties. They emphasized that the use of lesion texture could be helpful in identification of plant diseases. However, color and shape features can also be helpful in identifying lesions with similar texture [24]. Sharif et al. [25] investigated the utility of color and shape features with lesion texture for detecting the diseases in *citrus*. They used a multiclass SVM classifier and achieved 95.8% accuracy. Ahmad et al. [26] presented a framework, which is based on SVM and a combination of color, texture and shape features for disease detection in guava. The selection of optimized features out of a large set of extracted features can lead to more efficient disease detection [25]. Khan et al. [27] used genetic algorithm to select optimized features and used them to classify the diseases in *apple* leaves and achieved a comparatively better accuracy of 97.10%. In addition, authors in [28] showed benefits of ANN to identify the diseases in plants. The parametric nature of ANN and certainty in model size helped to make faster and accurate predictions. Zhang et al. [29] also used ANN to find diseases in *maize* crop using color, texture, and shape features. In [30], authors used a neural network with color and shape properties of infectious leave to investigate disease in *rice* crop.

In recent times, CNN has emerged as a promising technique for plants specific applications. CNNs have capabilities to automatically generate low and high-level features. Geetharamani and Arun Pandian [31] developed a 9-layer deep CNN to detect multiple diseases in various cultures. This model achieved 96% classification accuracy. In order to enhance the accuracy [32] proposed a deep CNN model of 14 layers. The model achieved remarkable accuracy, however it was computationally high expensive having a large number of trainable parameters. A model with a large number of layers takes long time in training and requires higher computing resources. Similarly, authors in [33] developed a deep CNN for plant disease detection, which was better than several baseline classifiers. Some deep CNN models were also developed by Agarwal et al. [34] to find the efficient solution to plant disease detection problem. However, these models are computationally expensive consisting of higher number of layers and take more training time. Kamal et al. [35] used region-based convolutional neural network (RCNN) to detect diseases in *apple* plants. RCNN model is highly computationally expensive but performed considerably better than some other models.

Deep CNN models are successful but computationally expensive at the same time. Fine-tuning and transfer learning allow to reuse the the existing models trained for some other problem. There are several well known models trained for general purpose object detection problems. These pre-trained models can be fine tuned using transfer learning to perform plant disease detection. The pre-trained models such as CaffeNet, AlexNet, VGG, and GoogLeNet were used to address the disease identification problem in various plants [36]. Coulibaly et al. [37] explored a fine tuned VGG-16 model to detect mildew diseases in *millet* crop images. Similarly, transfer learning based CNN model was

used in [38] to classify *tomato* leaf infections. VGG-16 model with two additional Inception modules was used in [39] in order to accurately detect *apple* leaf diseases. The model achieved accuracy of 97.14% but inclusion of two Iinception units made VGG-16 more computational expensive. A work showcased various recent pre-trained models for disease detection in *guava* plants [40]. In [41], authors exploited pre-trained ResNet-50 model through transfer learning approach to identify disease in *tomato* crop with 97% accuracy. The similarity among symptoms of different diseases raises the uncertainty in detection. In order to quantify such uncertainty, researcher in [42] embedded Monte-Carlo dropout (MCD) in VGG-16. However, it is computationally expensive and less accurate. An investigation was made by Shin et al. [43] on different pre-trained models to detect *powdery mildew* infection in *strawberry*. The augmentation techniques were also used by them to improve training data. Authors in [44] explored fine-tuned ResNet-50 and MobileNet models on a self-prepared *apple* leaf dataset and found MobileNet comparatively more stable and computationally better. Recently, fine-tuned InceptionV3 model was explored by Tahir et al. [45] to detect *apple* leaf diseases. They achieved 97% accuracy. In another study, AlexNet model was used to predict *apple scab* disease but the prediction accuracy was not found better than some existing methods [46]. The pre-trained models have shown good performance in plant disease detection. But these models are not specifically trained for such purpose and therefore, lead to chances for precision error. Based on the literature survey, it can be established that it is desirable to develop a deep CNN model specifically designed to efficiently detect plant diseases with high accuracy and less computational burden.

III. MATERIALS AND METHODS

This section covers complete details related to the development of deep CNN (Conv-3 DCNN) model to investigate the diseases in apple crops. The process of training and validating the proposed deep CNN model is presented along with the detailed mathematical formulations.

A. DATASET AND PRE-PROCESSING

The *apple* leaf images are taken from the PlantVillage dataset available openly through the PlantVillage project [47]. The dataset contains 3171 RGB images of *apple* plant leaves that are divided into four classes. The three classes correspond to 3 apple diseases namely *black rot, scab*, and *apple cedar rust*. The remaining one class represents healthy/uninfected leaf images. The expertly curated *apple* leaf images of size 256 × 256 for all four categories were captured with a simple background at various plant development stages in laboratory condition [48]. Figure 1 shows sample images from each class. The disease classes are named after the apple diseases and healthy class represents leaves without any disease. Plantvillage dataset has been used in a large number of research works. However, some other datasets such as Guava [49] and Citrus [50] are also available.

It is crucial to have a variety of images of leaves in the dataset so that model can learn important variations during the training. It helps to improve the generalizability of the deep CNN model. Augmentation is an approach to create variations of the image artificially. Some transformations such as shift, shearing, scaling, zoom, and flipping are employed in this work to transform the images. These transformations create minor variations in images that help to introduce variety in the training set. It in turn assists to reduce overfitting and helps the model to achieve better tolerance and ability to generalize. Figure 2 illustrates the process of data augmentation schemes. Some samples generated by augmentation are given in Fig. 3.

B. DEEP CONVOLUTIONAL NEURAL NETWORK

The deep CNN is a feed-forward ANN based deep learning technique. Here, the word deep refers to higher number of layers in a CNN. Usually, a deep CNN is constructed by stacking several building blocks such as convolutional layers with a typical non-linear activation unit, pooling layers, and fully connected layers. It has exhibited certain advantages over the state-of-the art ML based methods as it doesn't require any additional efforts for feature engineering. It is successfully used in several applications including image/text classification, NLP, and precision agriculture, etc. [6]. The initial layers of deep CNN extracts different types of lower level features.

1) CONVOLUTIONAL LAYER

The convolutional layer is responsible to perform convolution a filter (kernel) on an input image. The convolutional layer produces feature maps by finding the local conjunction that appears in the previous layers. Fundamentally, the convolutional layer is a combination of two components: a linear convolution operation and a non-linear activation unit.

The convolution operation is performed over the volumes of images with more than one channels such as RGB images [51] and expressed as in (1):

$$conv\,(I,K)_{x,y} = \sum_{i=1}^{n_H}\sum_{j=1}^{n_W}\sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k} \qquad (1)$$

where the kernel $K\,(f_h, f_w, n_C)$ convolve with the image $I\,(n_H, n_W, n_C)$ of different size but of similar no. of channels n_C and generate a feature map $O\,(o_H, o_W, z)$. The f_h, f_w represent the height and width of the kernel. And, n_H, n_W denote the height and width of the given image. Conventionally, the kernel is considered as an odd dimensional square window, i.e., $f_H = f_W = f$. The dimension of the generated feature map [51] is defined as in (2):

$$Feature\_map\,(o_H, o_W, z) = \Big(\lfloor\frac{n_H + 2p - f}{s} + 1\rfloor, \lfloor\frac{n_W + 2p - f}{s} + 1\rfloor, z\Big) \qquad (2)$$

where symbol p denotes the value of padding and s is stride, and z is the number of kernels convolved with input image.

(a) Black Rot (b) Apple Scab (c) Cedar Apple Rust (d) Healthy leaf

FIGURE 1. Samples of the apple leaf images of four classes namely Black rot, Scab, Cedar rust, and healthy leaves collected from the PlantVillage dataset.

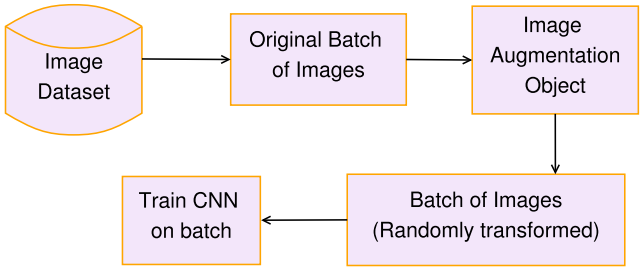

FIGURE 2. Representation of the process of data augmentation/manipulation.

The rectified linear unit (ReLU) is the most widely employed activation function [52]. ReLu doesn't activate all neurons at the same time. The neurons are activated only when output of the convolution unit or other linear transform is either equal to or greater than zero. It is expressed as in (3):

$$f(z) = \max(0, z) \tag{3}$$

2) POOLING LAYER

The pooling layer is used for downsampling of the feature maps generated by the convolutional layers. It reduces the size of activation maps that contain a large number of parameters. Thus, it reduces the computational burden, controls overfitting and reduces the training time. The max, min, average are the major pooling operations. But, max pooling is widely employed and takes the maximum value from each input patch. The max pooling operation is given in (4):

$$Max\_Pooling : y_j = \max_{i \in R_j}(P_i) \tag{4}$$

where R denotes a receptive field containing P pixels. The dimension of generated feature map [52] is defined in (5).

$$Feature\_map\,(o_H, o_W, n_C) = \Big(\lfloor \frac{n_H + 2p - f}{s} + 1 \rfloor, \lfloor \frac{n_W + 2p - f}{s} + 1 \rfloor, n_C\Big) \tag{5}$$

The pooling operation only modifies the dimensions n_H and n_W whereas n_C remains unchanged.

3) FULLY CONNECTED LAYER

The fully connected (FC) or dense layers in CNN are similar to the layers in traditional neural networks and typically connected at last stages of a CNN to construct output layers with a desired number of outputs. The FC layers operate on 1-D data. The flatten layer arranges the 2-D output of previous layers in a 1-D representation. The FC layers conduct two types of functioning: linear and non-linear transformations. These transformations can be expressed as in (6)-(7):

$$Z = W^T \cdot X + b \tag{6}$$

$$O = f(Z) \tag{7}$$

where, X is the input feature map, W is weight, and b denotes bias terms, and O denotes the output of the fully connected layer.

For better prediction, optimal weights are eventually needed to reduce the loss function. Gradient descent approach is the most widely used technique to find the optimal weights. Adam algorithm is another technique to get less noisy and smoother path during optimizing the gradients [53]. It performs learning rate annealing based on finding the adaptive estimates of the lower order moments.

C. PROPOSED METHOD

The proposed deep CNN (Conv-3 DCNN) model composed of 3 convolutional layers and two fully connected layers after the three max-pooling units. ReLU is explored as a nonlinear activation function at each convolution layer and at first dense layer. Softmax function is employed at the output layer to classify *apple* plant diseases. The softmax function is responsible for multiclass classification and assumes that each sample belongs to exactly one class. Figure 4 illustrates the pipelining of various layers along with the activation functions used in the developed deep CNN model.

A dropout layer is also used additionally at the third max pool layer to the effect of overfitting. The dropout unit basically eliminates some random selected neurons. The network could not rely on any one feature therefore some neurons are ignored to spread out the weights for better generalization.

Initially, at the first convolution level 32 filters (3×3) with valid padding and stride (1, 1) is selected to convolute

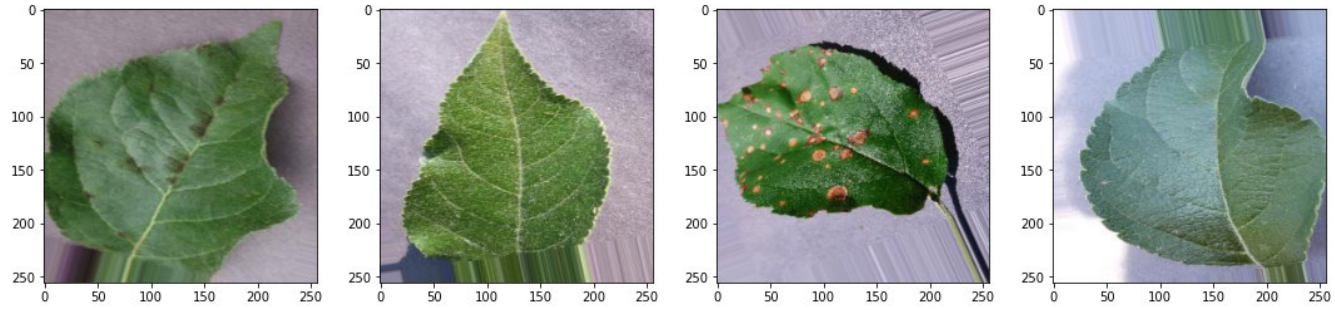

FIGURE 3. Random samples of augmented images through various transformations.

TABLE 3. Structure of the deep CNN model employed for apple plant leaf disease detection.

No.	Layer Name	Filter Size	Number of Filters	Stride	Padding	Weights	Bias	Activation
1	Input (Image)							$256 \times 256 \times 3$
2	Convolution + ReLU	3×3	32	[1 1]	valid	$3 \times 3 \times 3 \times 32$	$1 \times 1 \times 32$	$254 \times 254 \times 32$
3	Pooling (Max)	2×2		[2 2]	valid			$127 \times 127 \times 32$
4	Convolution + ReLU	3×3	16	[1 1]	valid	$3 \times 3 \times 32 \times 16$	$1 \times 1 \times 16$	$125 \times 125 \times 16$
5	Pooling (Max)	2×2		[2 2]	valid			$62 \times 62 \times 16$
6	Convolution + ReLU	3×3	8	[1 1]	valid	$3 \times 3 \times 16 \times 8$	$1 \times 1 \times 8$	$60 \times 60 \times 8$
7	Pooling (Max)	2×2		[2 2]	valid			$30 \times 30 \times 8$
8	Flatten							$1 \times 1 \times 7200$
9	Fully Connected + ReLU					16×7200	16×1	$1 \times 1 \times 16$
10	Fully Connected + Softmax					4×16	16×1	$1 \times 1 \times 4$
11	Output							$1 \times 1 \times 4$

over RGB images of size 256×256. It generates 32 feature maps of size 254×254. In output feature map the number of channels is same as the number of filters applied. At the first pooling layer, the previously generated feature maps are downsampled by a kernel of size 2×2 and 32 feature maps of size 127×127 are generated. The same kernels are used at respective higher layers. The working process of the proposed deep CNN method can be understood with the help of layered architecture illustrated in Fig. 5. All the values and parameters related to the development of the deep CNN model are summarized in Table 3.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The deep CNN model is trained and evaluated on the dataset described in Section III-A. The dataset is partitioned into training, validation, and test with a ratio of 70-20-10 using hold-out validation method. Accordingly, the training, validation, and test sets are comprised of 228, 634, and 319 images respectively, and are outlined in Table 4.

A. EXPERIMENTAL SETUP

The deep CNN model is implemented in Python using Keras, Scikit-learn, Opencv, and Pillow libraries on the Google Colab virtual platform consisting of Nvidia P100 GPU with 12 GB RAM. The results are visualized with the help of Matplotlib library. With a learning rate of 0.0001 and mini-batch size of 50, the model is trained for 1000 epochs. The categorical cross entropy approach is employed to estimate the loss. This approach is suitable for multiclass classification problem and defined using (8).

$$loss(L) = -\sum_{m=1}^{N} y_{i,m} \log\left(p_{i,m}\right) \tag{8}$$

where N denotes no. of classes such as ($N > 2$), y represents a binary indicator that indicates correct labeling of class m and observation i, and p is the probability of predicting that the observation i belongs to the class m.

The Adam optimization algorithm is used for gradient optimization with $\beta_1 = 0.9$, $\beta_2 = 0.999$ exponential decay rates for first and second moments. The hyperparameter

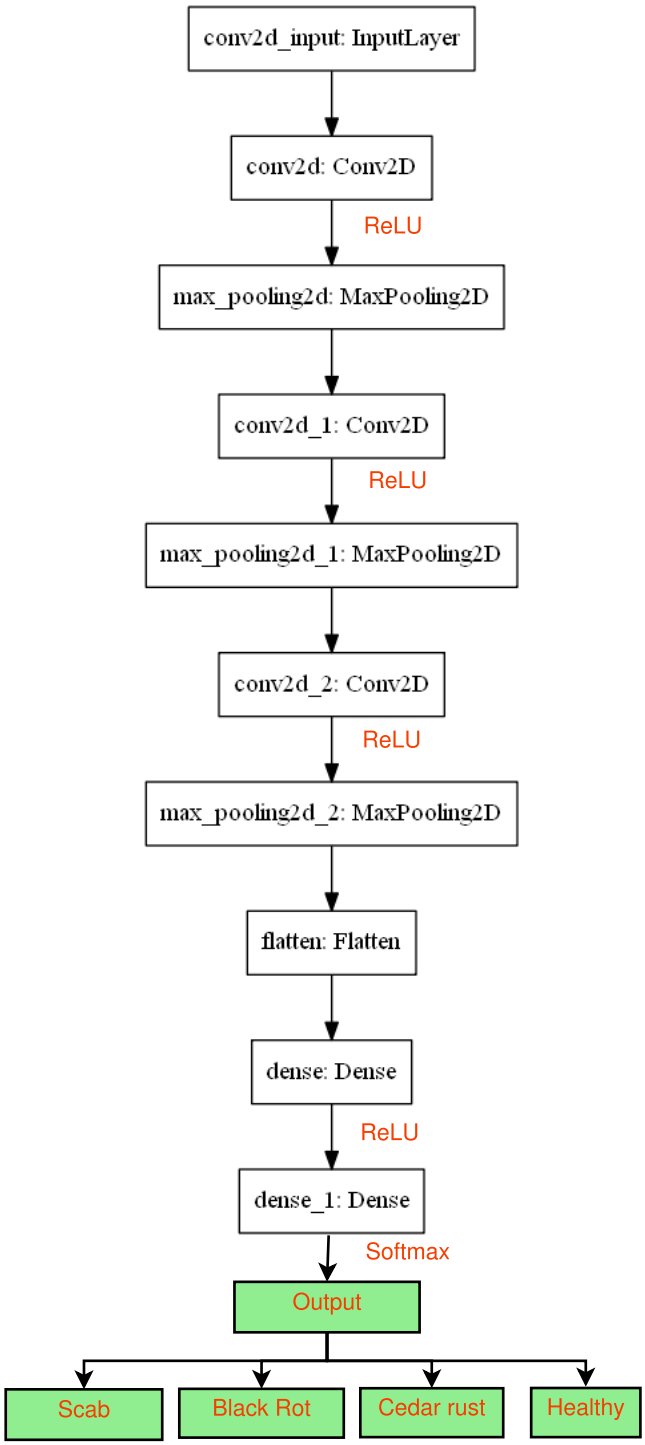

FIGURE 4. Pipelining of layers in the deep CNN model.

tuning is performed using random search techniques to select the best suitable hyperparameters. The details of optimized hyperparameters related to the developed deep CNN model are reported in Table 5.

Model effectiveness is checked by comparing the results with those achieved for the pre-trained built-in models like VGG-19 [9], ResNet-152 [10], DenseNet-201 [11], MobileNetV2 [12], ResNet-50 [10], VGG-16 [9], InceptionV3 [13], Xception [14], and MobileNet [15]. All the pre-trained models are applied on the same training and test dataset. These models are used through the transfer learning approach which improves the model generalization and allows rapid progress.

B. PERFORMANCE METRICS USED FOR MODEL EVALUATION

The model performance is evaluated in respect of various metrics which can be derived from confusion matrix.

1) Accuracy: Accuracy is a commonly used performance metric that provides the ratio of correct prediction and the total no. of data samples or the total prediction made by the classifier. Both overall and individual class accuracy can be determined from the confusion matrix.
2) Precision: It gives the fraction of all positive predictions that are actually positive.
3) Recall or Sensitivity: It is the measure of ratio of correctly predicted positive samples and all positive samples.
4) Specificity / True Negative Rate (TNR): It defines the ratio of correctly predicted negative samples and all negative samples.
5) F1-Score: It gives the harmonic mean of the precision and recall, and refers to the no. of correctly classified instances. It ranges from 0 to 1.
6) Cohen Kappa: Cohen Kappa is a measure of inter-rater reliability. Its higher value defines the model to be more reliable. It ranges from 0 to 1.
7) AUC-ROC (Area Under the Curve of Receiver Operating Characteristics): Area under Curve is a measure of ability of the model to distinguish the classes. The higher AUC value indicates that the model is better in distinguishing the classes. ROC curve is employed to visualize the behavior of a multi-class classifier and represented as the probability curve that plots model sensitivity against the false-out rate.

C. RESULT ANALYSIS

1) VISUAL ANALYSIS OF FEATURE MAPS

The deep CNN reduces the need of additional feature engineering processes. The kernels applied at different layers generate the ''activation maps'' or generally called ''feature maps''. The feature maps obtained at different convolutional and pooling layers for an *apple* leaf infected with *Cedar Rust* are demonstrated in Fig. 6-8.

The early convolutional layers learn the features like edges, corners, and simple texture and preserving the most of the information in the input image. Pooling layers downsample the feature maps. Figure 6 illustrate 32 feature maps generated at first convolutional block (conv1+ pool1). It can be observed from the figures that orange, pale yellow lesions are highlighted within the leaf image. But, deeper layers defines an abstracted form of original image and encodes high level information such as disease spots and complex

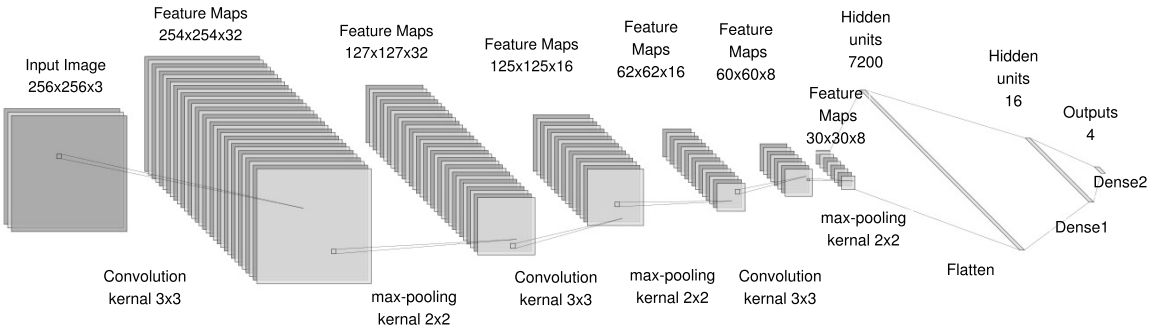

FIGURE 5. Layered architecture of proposed deep CNN model for apple plant leaf disease detection.

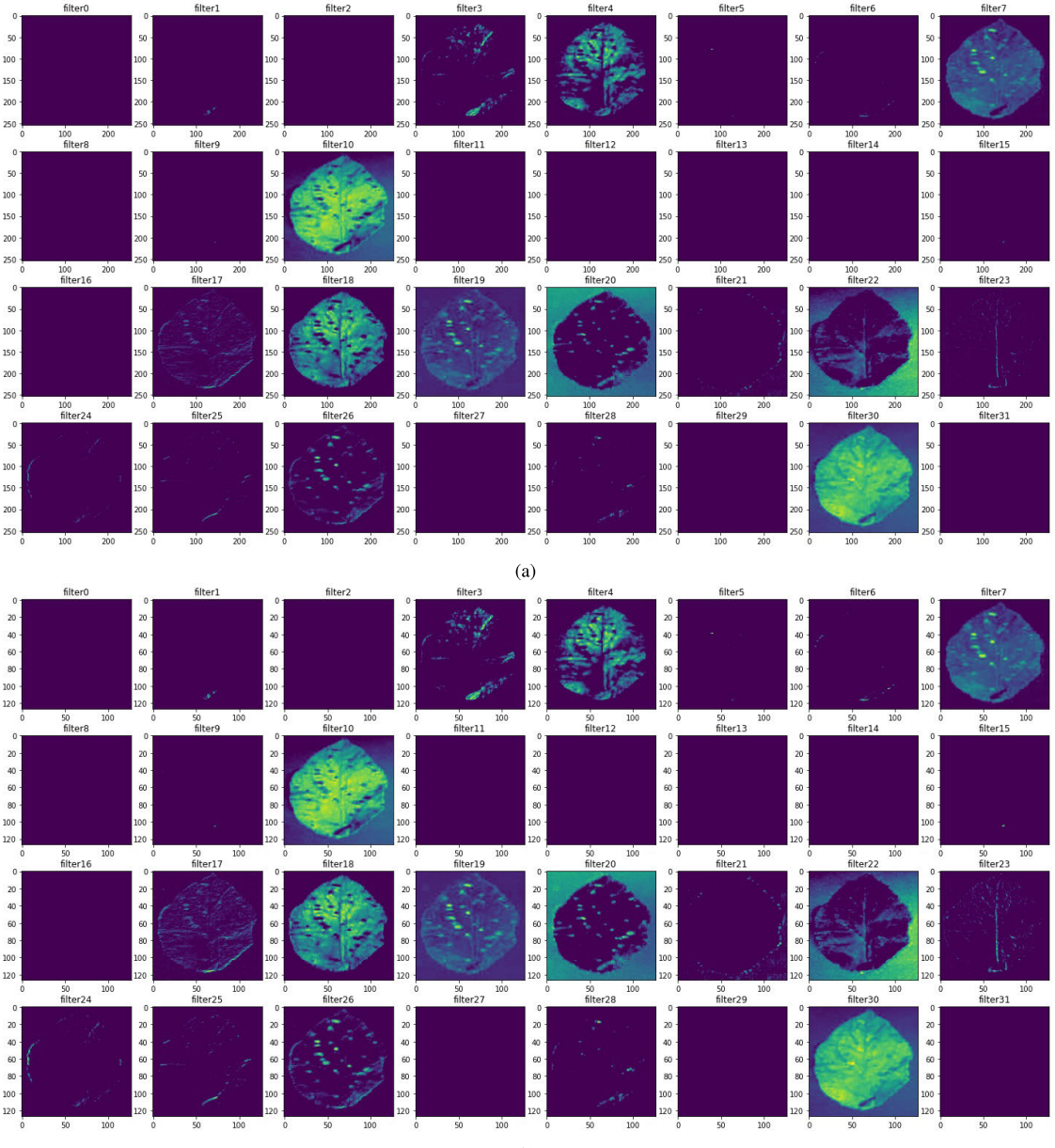

FIGURE 6. Feature maps generated at (a) first convolutional layer, and (b) first pooling layer in deep CNN model.

texture. Specifically, the feature maps at second convolutional layers tells less about the image and more about the lesions specific to a disease. Sixteen feature maps specific to second convolutional black (conv2+pool2) are illustrated in Fig.7.

TABLE 4. Splitting the apple plant leaf dataset into three subsets training, validation and testing with the 70-20-10 strategy.

Infection	Training Set Size	Validation Set Size	Test Set Size	Total
Scab	441	126	63	630
Black rot	434	124	63	621
Cedar rust	192	55	28	275
Healthy	1151	329	165	1645
Total Images	2228	634	319	3171

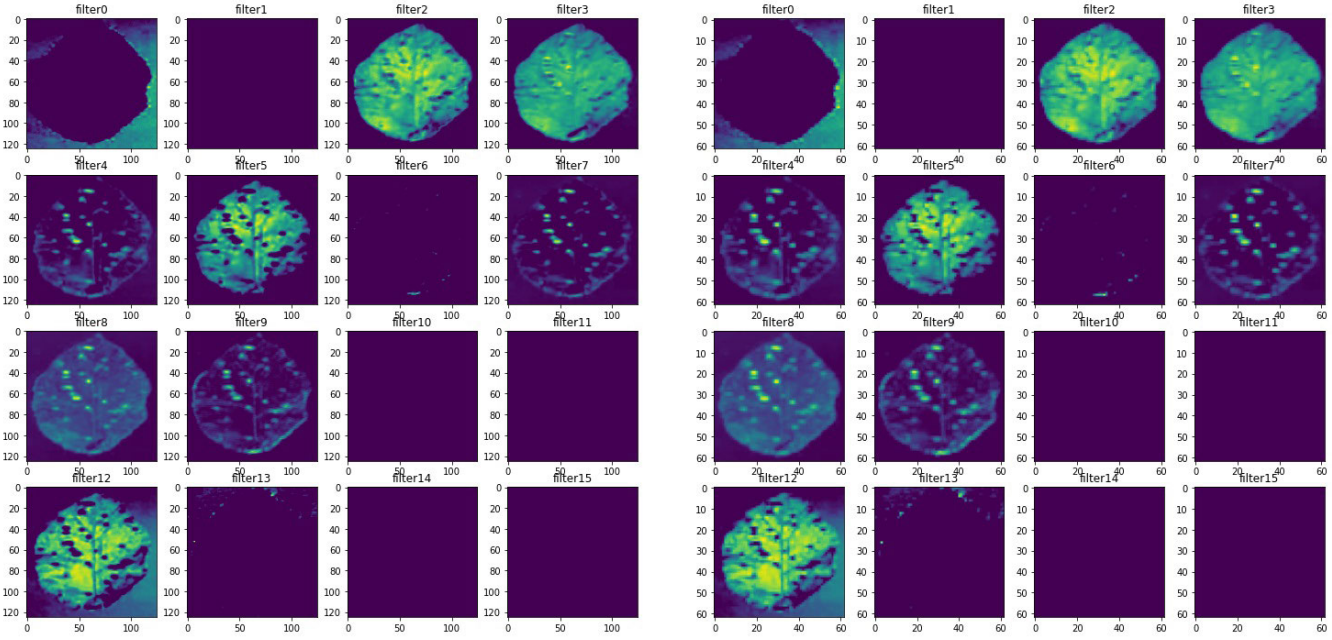

FIGURE 7. Feature maps generated at (a) second convolutional layer, and (b) second pooling layer in deep CNN model.

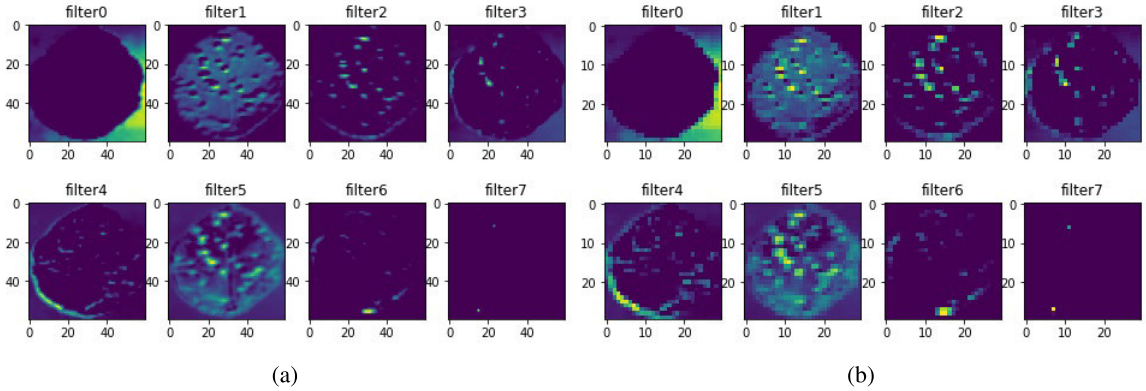

FIGURE 8. Feature maps generated at (a) third convolutional layer, and (b) third pooling layer in deep CNN model.

Moreover, the feature maps get sparser at more deeper layers that detect specific information about patterns and shapes. Figure 8 shows 8 features maps related to third convolutional block (conv3+pool3). These feature maps represent only the complex information such as lesion shape and size. The fully connected layers learn to connect high level features to the classes.

2) ACCURACY ANALYSIS

In this experiment, the proposed CNN model is evaluated for its accuracy. The training accuracy is plotted along with validation accuracy against the number of epochs in Fig. 9(J). It can be observed that there is only a small difference in training and validation accuracy curves. It declare that the proposed model fits well for the addressed problem. It can

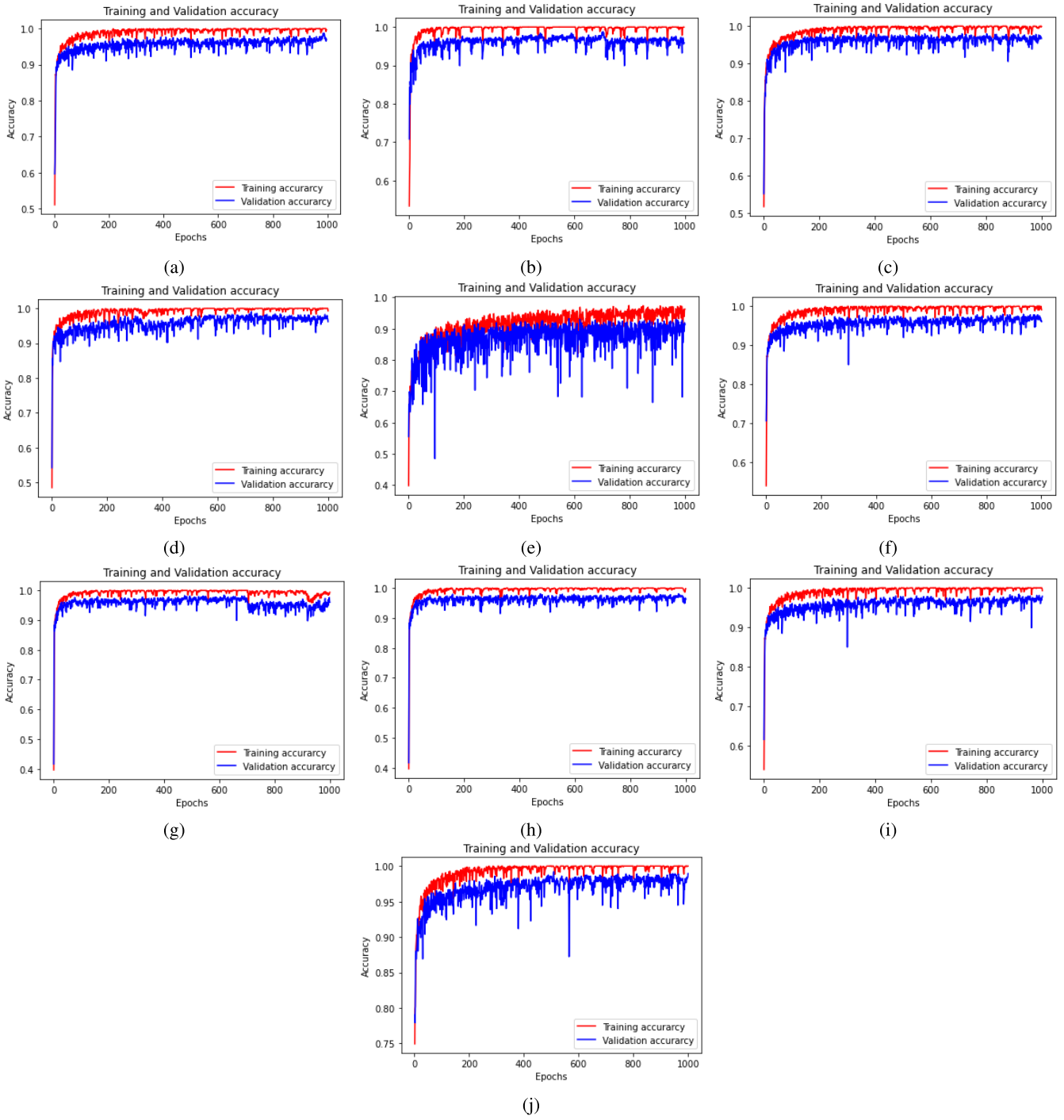

FIGURE 9. Accuracy versus epochs: a. VGG-19, b. ResNet-152, c. DenseNet-201, d. MobileNetV2, e. ResNet-50, f. VGG16, g. InceptionV3, h. Xception, i. MobileNet, j. Proposed deep CNN.

also be observed from the graph that model accuracy initially increases sharply with increasing epochs. Later it improves slowly. The plots for loss function of the model in Fig. 10(J) also provide similar information. Initially, when the model starts learning, the validation loss is apparently higher than the training loss. Once the model learned the features of disease classes, the validation loss is decreased and the validation accuracy is increased. Specifically, the accuracy of the model starts saturating at 1000 epochs and the validation error decreases apparently.

Figure 11 illustrates the confusion matrix obtained for the model. The most of the samples lie along the diagonal of the confusion matrix indicating a good performance. Various parameters including recall, precision, Kappa, f1- score, and AUC-ROC can be estimated through confusion matrix. The overall accuracy of the model at parameters specified in the

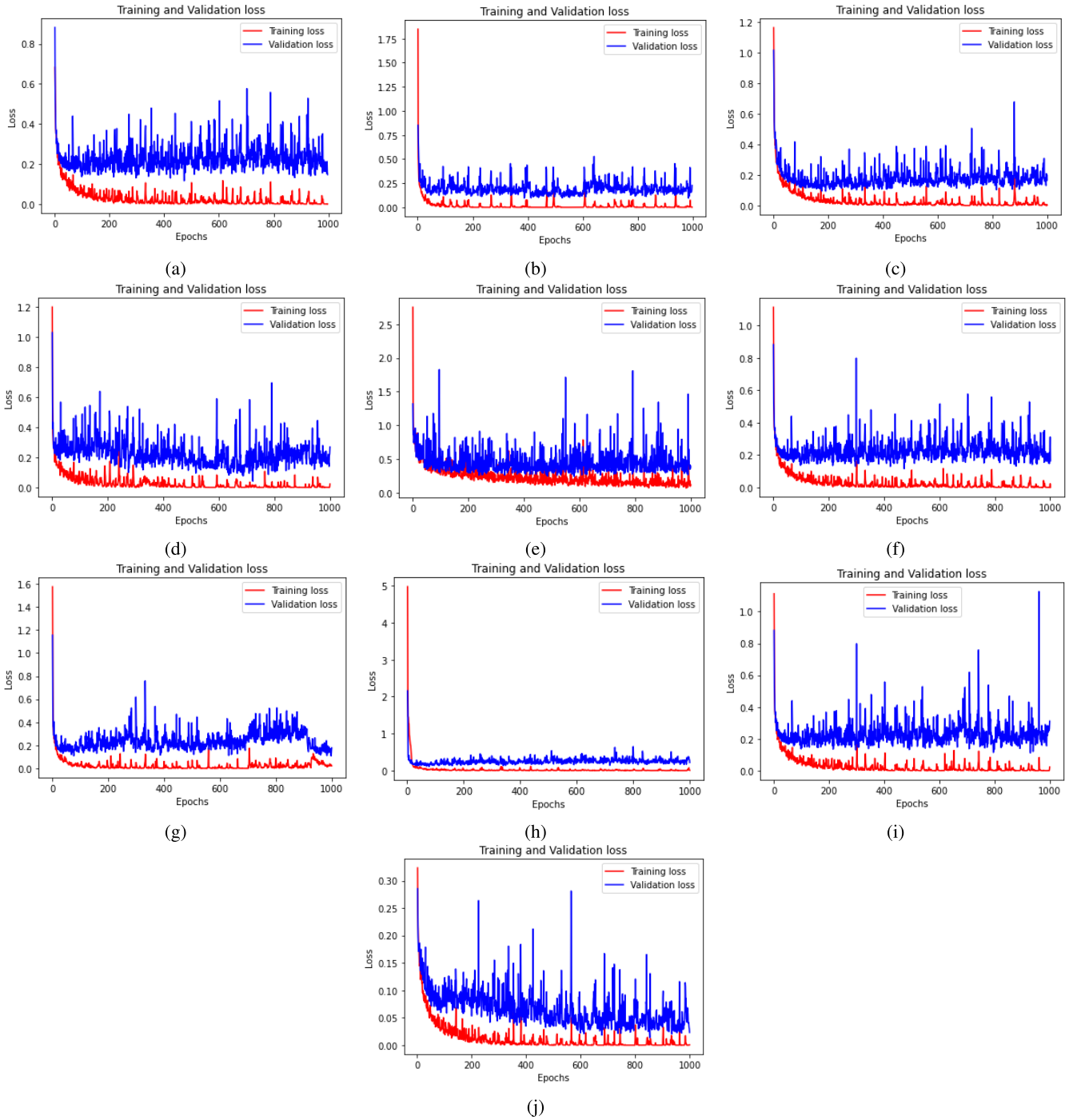

FIGURE 10. Loss function versus epochs: a. VGG-19, b. ResNet-152, c. DenseNet-201, d. MobileNetV2, e. ResNet-50, f. VGG16, g. InceptionV3, h. Xception, i. MobileNet, j. Proposed deep CNN.

Table 5 is obtained as 98%. On evaluating the accuracy for individual diseases, it is observed that the proposed model identifies each disease with a good accuracy as given in Fig. 12.

The AUC-ROC curves for the model are given in appendix VI. It can be observed from the curves that the model has good ability to distinguish among the classes. In a good test result the curve should be closer to upper left corner. The ROC curve plot for *apple_scab* class is the lowest whereas the curve for the class *black_rot* is highly closer to the upper left corner. Therefore, the model is able to identify the leaf images infected with *black_rot* disease more accurately in comparison to other classes.

Comparison With Other Methods: The performance of the model is contrasted with pre-trained models including VGG-19 [9], ResNet-152 [10], DenseNet-201 [11], MobileNetV2 [12], ResNet-50 [10], VGG-16 [9], InceptionV3 [13], Xception [14], and MobileNet [15] using

TABLE 5. Hyperparameters for the proposed deep CNN.

Parameter	Description
No. of Epochs	1000
Mini-batch size	50
Learning rate	0.0001
Training set Size	2218
Validation set size	634
Test set size	319
Validation step	13
Activation function	ReLU
Dropout Rate	0.4
Adam β_1	0.9
Adam β_2	0.999

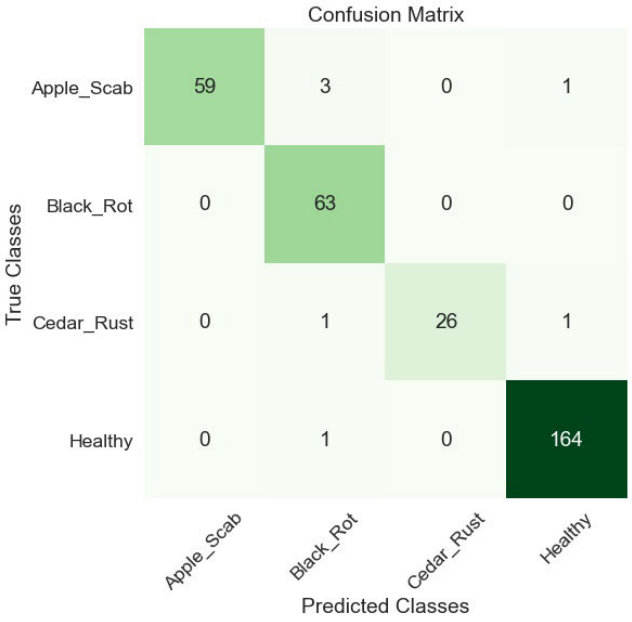

FIGURE 11. Confusion matrix of the proposed model.

transfer learning approach. The developed model is compared with pre-trained models in terms of overall accuracy, time consumption (training and testing time), FLOP, size (storage space), and depth of the model as given in Table 6. In addition, the models are also compared in terms of their precision, recall, F1- score, and ROC AuC values. The impact of epochs on the performance of these models is also evaluated. For a particular model, the training epochs varies from 1 to 1000. Both training and validation accuracies are indicated against the number of epochs and compared with the plot of our model in Fig. 9. The uneven variation between the training and validation accuracies of pre-trained models shows that the proposed model best fit for apple disease detection. In addition, the loss function for all these models is also evaluated and given in Fig. 10 with the plot of proposed model. The rigorous observation of the results reveals that the proposed model outperforms in comparison to

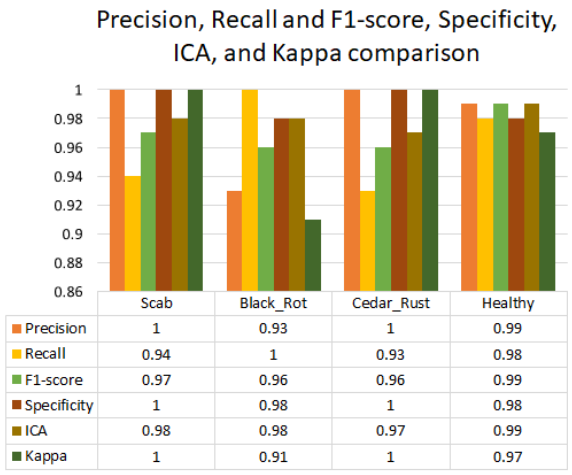

FIGURE 12. Class-wise performance in respect of precision, f1-score, specificity, recall, individual class accuracy (ICA), and Kappa coefficient.

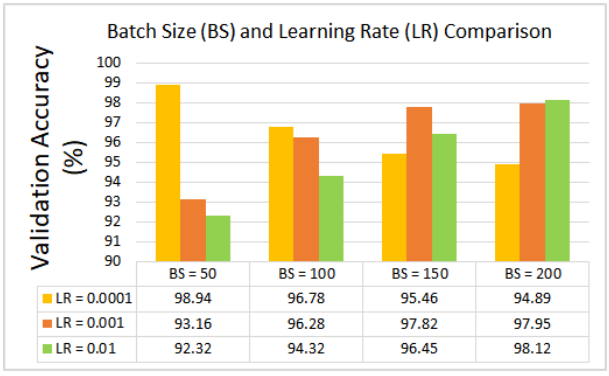

FIGURE 13. Validation accuracy achieved at different learning rates and mini-batch sizes.

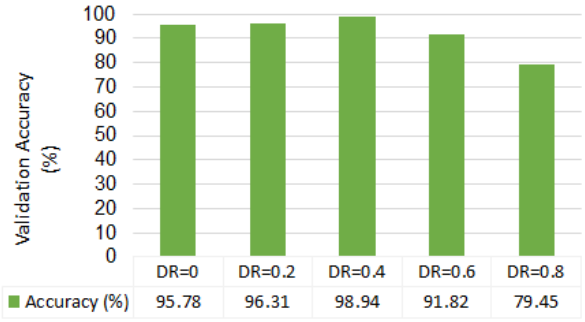

FIGURE 14. Validation accuracy achieved at different dropout rates.

pre-trained models. And, among all other pre-trained models MobileNetV2 model performed better.

The proposed model needs lesser storage and lower execution time in comparison to other models as observed from Table 6. The pre-trained models have a large number of weights that require additional storage. On the contrary, proposed model involves the smaller no. of parameters than the pre-trained models evaluated in this work. Generally, a high number of parameters and intermediate output/feature maps or a high number of convolutions require more storage space and more time to execute. The proposed model performs well

TABLE 6. Comparison of proposed Deep CNN model with popular pre-trained models.

Model	No. of Parameters	Training Time (h)	Testing Time (s)	Size	FLOP	Depth	Accuracy	Precision	Recall	F1-Score	ROC AuC
VGG-19	143,667,240	25	257	549 MB	1.96E+10	19	0.96	0.94	0.92	0.92	0.97
ResNet-152	58,772,356	19	73	230 MB	1.13E+10	152	0.95	0.95	0.91	0.93	0.96
DenseNet-201	20,242,984	17	48	80 MB	4.37E+09	201	0.94	0.95	0.92	0.93	0.94
MobileNetV2	3,538,984	9	54	14 MB	3.14E+8	53	0.97	0.96	0.93	0.94	0.93
ResNet-50	25,636,712	12	20	98 MB	1.10E+10	50	0.90	0.90	0.88	0.88	0.89
VGG-16	138,357,544	15	16	528 MB	1.58E+10	23	0.96	0.94	0.95	0.94	0.94
InceptionV3	23,851,784	12	14	92 MB	2.00E+09	159	0.95	0.95	0.94	0.94	0.97
Xception	22,910,480	13	113	88 MB	1.10E+10	126	0.95	0.93	0.94	0.93	0.96
MobileNet	4,253,864	10	82	16 MB	5.69E+11	88	0.96	0.95	0.94	0.94	0.97
Deep CNN	121,964	8	11	11 MB	1.32E+08	3	0.98	0.98	0.97	0.97	0.99

TABLE 7. Comparison of classification accuracy of proposed model with the methods exploited in prior researches in the domain.

References	Method	Crop/Plant	Dataset Used	No. of Classes	No. of Parameters	Accuracy
[34]	CNN	Tomato	PlantVillage	10	208,802	91%
[33]	CNN	Assorted cultures	PlantVillage	39	1,156,516	98%
[41]	ResNet-50	Tomato	PlantVillage	10	25,636,712	97%
[39]	VGG-INCEP	Apple	Apple Leaf images	5		97%
[31]	CNN	Assorted cultures	PlantVillage	39	212,543	96%
[45]	InceptionV3	Apple	PlantVillage	4	23,851,784	97%
[46]	AlexNet	Apple	PlantVillage	2	62,378,344	87%
[54]	RCNN & ResNet-50	Apple	PlantVillage	4	25,636,712	96%
[44]	MobileNet	Apple	Apple Leaf images	2	4,253,864	74%
	ResNet-50				25,636,712	78%
[27]	SVM	Apple	PlantVillage	4		97%
[20]	SVM	Apple	PlantVillage	3		96%
This work	Proposed CNN	Apple	PlantVillage	4	121,964	98%

with a lower number of convolution operations and parameters thus it requires less storage space and low computation time. Additionally, a less number of add-multiply operations are performed in the proposed CNN, therefore it requires a less number of FLOP. Due to these benefits, this model can be easily deployed in the form of a mobile app in smartphones to assist the non-pathologist/ farmers to diagnose and identify *apple* diseases in the orchards at real-time.

A comparative study is also performed with other existing methods. For this, only those research works are considered that use similar experimental setup and PlantVillage dataset. The comparison of such methods is given in Table 7 on the basis of various parameters. The table revels that the introduced model gives considerably better results than most of the other methods. Its performance is slightly inferior to the model used by [33] but the proposed model involves much fewer parameters leading to lesser training time.

3) INFLUENCE OF LEARNING RATE AND BATCH SIZE

The CNN model is trained with mini-batch gradient descent, which splits the training dataset to small subsets. Mini-batch calculates the gradient of loss function and updates the weights, which results robust convergence of the network [55]. The hyperparameters batch size (BS) and the learning rate (LR) of the network are highly correlated [56]. In this experiment, the model behavior is analyzed on various mini-batch sizes and learning rates. The results are obtained with batch size of 50, 100, 150 and 200 and learning rate of 0.0001, 0.001, and 0.01 as reported in Fig. 13. The proposed model obtained the highest validation accuracy at learning

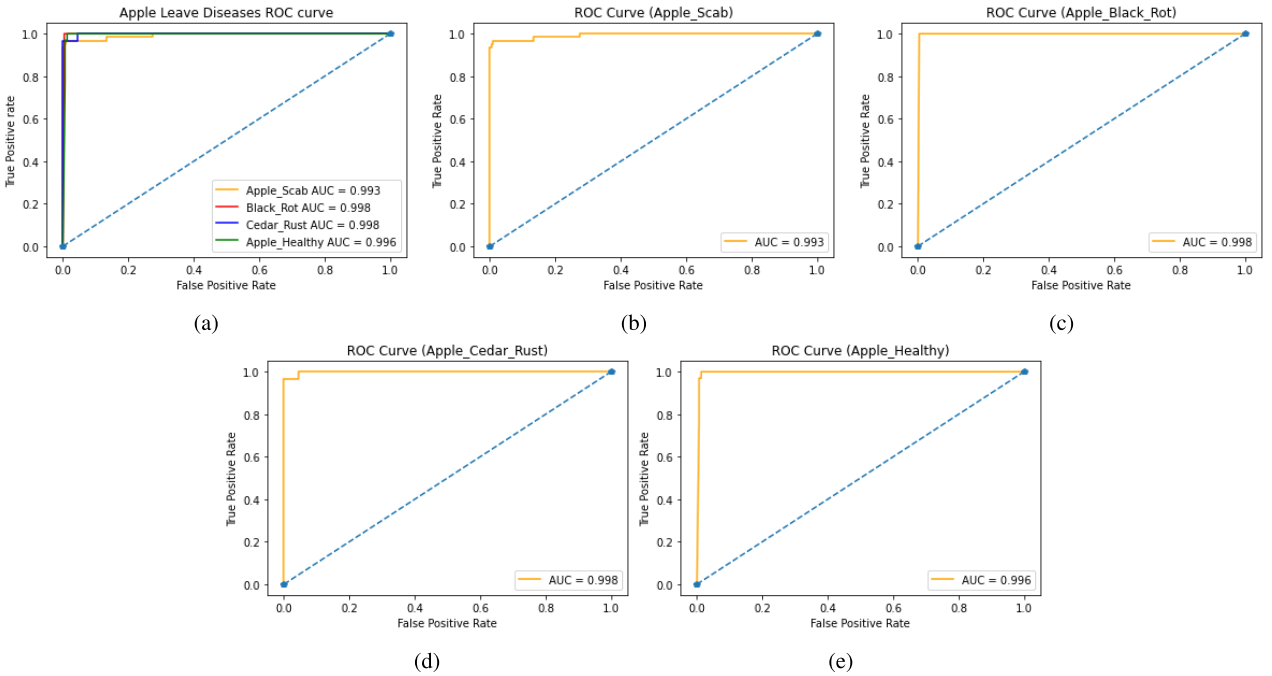

FIGURE 15. ROC Curves of a) proposed CNN model, b) Apple_Scab disease, c) Apple_Black_Rot disease, d) ROC Curve of Apple_Cedar_Rust disease, and e) Apple_Healhty category.

rate 0.0001 and mini-batch size 50. It is observed that smaller learning rate is more influential with smaller batch size. With larger batches, the higher learning rate gives better accuracy. Overall, smaller batch with smaller learning rate could be recommended for such problems.

4) INFLUENCE OF DROPOUT RATE

Dropout is a regularization technique, which allows ignoring some units during the model training and to reduce the overfitting. The influence of the dropout rate (DR) is analyzed by training the model at distinct dropout values starting from 0.0 to 0.8 with a step size of 0.2. The validation accuracy of the model against varying dropout rate is provided in Fig. 14. The highest validation accuracy of 98.94% is achieved at dropout rate 0.4. It is also observed from the figure that higher dropout rate is not good for accuracy.

V. CONCLUSION AND FUTURE WORK

A deep CNN model was proposed in this work to identify diseases in apple crops with the help of leaf images. It can assist the non-expert farmers in *apple* orchards and lower the stress on plant pathologists. The model was trained on 3171 *apple* leaves for 1000 epochs. The accuracy of the model was evaluated to 98% on PlantVillage dataset. The rigorous investigation manifests the proposed model to be better than various pre-trained CNN models. The method was also found better than some other existing methods on the basis of various parameters including accuracy and memory requirements. The model achieves good accuracy for different diseases between 97% to 99%. The model successfully balanced the accuracy and precision. The AUC-ROC curve showed that the proposed model is reliable and consistent.

The possible extension of this work include collection of more leaf images of *apple* plants from various geographically different areas with varying image quality at different plant growth stages in different cultivation conditions. The large dataset with improved image variability would allow more rigorous experiments helping to improve the model to detect diseases at different stages for a variety of *apple* crops.

VI. APPENDIX A THE AUC-ROC CURVE DEMONSTRATING THE PROPOSED MODEL FOR ALL CLASSES

The ROC curve of the deep CNN model is illustrated in Fig. 15(a), additionally the class-wise ROC curve are also given here. See Fig. 15(b)-15(e).

ACKNOWLEDGMENT

The authors are extremely thankful to the Department of Computer Science, College of Computing, Khon Kaen University, Thailand.

REFERENCES

[1] T. Deshpande, "State of agriculture in India," *PRS Legislative Res.*, no. 113, pp. 1–29, 2017. [Online]. Available: https://www.prsindia.org/policy/discussion-papers/state-agriculture-india

[2] *The State of Food Security and Nutrition in the World 2020. Transforming Food Systems for Affordable Healthy Diets*, FAO and UNICEF, FAO and IFAD and UNICEF and WFP and WHO, Rome, Italy, 2020, Accessed: 25-June-2021. [Online]. Available: http://www.fao.org/documents/card/en/c/ca9692en/

[3] Horticulture Statistics Division (Government of India). (2018). *Horticultural Statistics at a Glance 2018*. [Online]. Available: https://agricoop.nic.in/sites/default/files/Horticulture%20Statistics%20at%20a%20Glance-2018.pdf

[4] *World Production, Markets, and Trade Report–Fresh Apples, Grapes, and Pears: World Markets and Trade*, United States Dept. Agricult. Foreign Agricult. Service, Foreign Agricultural Service/USDA, Washington, DC, USA, Accessed: Jul. 12, 2021. [Online]. Available: https://www.fas.usda.gov/data/fresh-apples-grapes-and-pears-world-markets-and-trade

[5] National Horticulture Board. *Report on Apple production in India*. Accessed: Dec. 2020. [Online]. Available: http://nhb.gov.in/report_files/apple/APPLE.htm

[6] V. K. Vishnoi, K. Kumar, and B. Kumar, "Plant disease detection using computational intelligence and image processing," *J. Plant Diseases Protection*, vol. 128, no. 1, pp. 19–53, Feb. 2021, doi: 10.1007/s41348-020-00368-0.

[7] S. Kaur, S. Pandey, and S. Goel, "Plants disease identification and classification through leaf images: A survey," *Arch. Comput. Methods Eng.*, vol. 26, no. 2, pp. 507–530, Apr. 2018, doi: 10.1007/s11831-018-9255-6.

[8] V. K. Vishnoi, K. Kumar, and B. Kumar, "A comprehensive study of feature extraction techniques for plant leaf disease detection," *Multimedia Tools Appl.*, vol. 81, no. 1, pp. 367–419, Jan. 2022, doi: 10.1007/s11042-021-11375-0.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556, doi: 10.48550/ARXIV.1409.1556.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385, doi: 10.48550/ARXIV.1512.03385.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, *arXiv:1608.06993*.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," 2018, *arXiv:1801.04381*.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015. [Online]. Available: https://arxiv.org/abs/1512.00567, doi: 10.48550/ARXIV.1512.00567.

[14] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016. [Online]. Available: https://arxiv.org/abs/1610.02357, doi: 10.48550/ARXIV.1610.02357.

[15] 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. [Online]. Available: https://arxiv.org/abs/1704.04861, doi: 10.48550/ARXIV.1704.04861.

[16] U. Mokhtar, N. El Bendary, A. E. Hassenian, E. Emary, M. A. Mahmoud, H. Hefny, and M. F. Tolba, "SVM-based detection of tomato leaves diseases," in *Intelligent Systems'2014*, D. Filev, J. Jabłkowski, J. Kacprzyk, M. Krawczak, I. Popchev, L. Rutkowski, V. Sgurev, E. Sotirova, P. Szynkarczyk, and S. Zadrozny, Eds. Cham, Switzerland: Springer, 2015, pp. 641–652.

[17] A. S. Deshapande, S. G. Giraddi, K. G. Karibasappa, and S. D. Desai, "Fungal disease detection in maize leaves using Haar wavelet features," in *Information and Communication Technology for Intelligent Systems*, S. C. Satapathy and A. Joshi, Eds. Singapore: Springer, 2019, pp. 275–286.

[18] S. Zhang, H. Wang, W. Huang, and Z. You, "Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG," *Optik Int. J. Light Electron Opt.*, vol. 157, pp. 866–872, Mar. 2018, doi: 10.1016/j.ijleo.2017.11.190.

[19] S. Zhang, X. Wu, Z. You, and L. Zhang, "Leaf image based cucumber disease recognition using sparse representation classification," *Comput. Electron. Agricult.*, vol. 134, pp. 135–141, Mar. 2017, doi: 10.1016/j.compag.2017.01.014.

[20] S. Chakraborty, S. Paul, and M. Rahat-uz-Zaman, "Prediction of apple leaf diseases using multiclass support vector machine," in *Proc. 2nd Int. Conf. Robot., Electr. Signal Process. Techn. (ICREST)*, Jan. 2021, pp. 147–151, doi: 10.1109/ICREST51555.2021.9331132.

[21] V. K. Vishnoi, K. Kumar, and B. Kumar, "Crop disease classification through image processing and machine learning techniques using leaf images," in *Proc. 1st Int. Conf. Adv. Comput. Future Commun. Technol. (ICACFCT)*, Dec. 2021, pp. 27–32.

[22] A. N. I. Masazhar and M. M. Kamal, "Digital image processing technique for palm oil leaf disease detection using multiclass SVM classifier," in *Proc. IEEE 4th Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Nov. 2017, pp. 1–6.

[23] S. Kaur, S. Pandey, and S. Goel, "Semi-automatic leaf disease detection and classification system for soybean culture," *IET Image Process.*, vol. 12, no. 6, pp. 1038–1048, Jun. 2018. [Online]. Available: www.ietdl.org

[24] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "SVM and ANN based classification of plant diseases using feature reduction technique," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 3, no. 7, p. 6, 2016.

[25] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agricult.*, vol. 150, pp. 220–234, Jul. 2018, doi: 10.1016/j.compag.2018.04.023.

[26] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damaševičius, B. Alouffi, and A. Alharbi, "AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery," *Sensors*, vol. 21, no. 11, p. 3830, Jun. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3830

[27] M. A. Khan, M. I. U. Lali, M. Sharif, K. Javed, K. Aurangzeb, S. I. Haider, A. S. Altamrah, and T. Akram, "An optimized method for segmentation and classification of apple diseases based on strong correlation and genetic algorithm based feature selection," *IEEE Access*, vol. 7, pp. 46261–46277, 2019.

[28] S. Gharge and P. Singh, "Image processing for soybean disease classification and severity estimation," in *Emerging Research in Computing, Information, Communication and Applications*, N. R. Shetty, N. H. Prasad, and N. Nalini, Eds. New Delhi, India: Springer, 2016, pp. 493–500.

[29] Z. Zhang, Y. Li, F. Wang, and X. He, "A particle swarm optimization algorithm for neural networks in recognition of maize leaf diseases," *Sensors Transducers*, vol. 166, no. 3, pp. 181–189, 2014.

[30] J. W. Orillo, J. Dela Cruz, L. Agapito, P. J. Satimbre, and I. Valenzuela, "Identification of diseases in Rice plant (oryza sativa) using back propagation artificial neural network," in *Proc. Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, 2014, pp. 1–6.

[31] G. G. and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, Jun. 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[32] J. A. Pandian, V. D. Kumar, O. Geman, M. Hnatiuc, M. Arif, and K. Kanchanadevi, "Plant disease detection using deep convolutional neural network," *Appl. Sci.*, vol. 12, no. 14, p. 6982, Jul. 2022, doi: 10.3390/app12146982.

[33] J. A. Pandian, K. Kanchanadevi, V. D. Kumar, E. Jasińska, R. Goňo, Z. Leonowicz, and M. Jasinski, "A five convolutional layer deep convolutional neural network for plant leaf disease detection," *Electronics*, vol. 11, no. 8, p. 1266, Apr. 2022, doi: 10.3390/electronics11081266.

[34] M. Agarwal, S. Gupta, and K. K. Biswas, "A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100473, doi: 10.1016/j.suscom.2020.100473.

[35] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104948, doi: 10.1016/j.compag.2019.104948.

[36] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[37] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, Jun. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0166361518305888

[38] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, "Automated tomato leaf disease classification using transfer learning-based deep convolution neural network," *J. Plant Diseases Protection*, vol. 128, no. 1, pp. 73–86, Feb. 2021, doi: 10.1007/s41348-020-00403-0.

[39] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[40] A. M. Mostafa, S. A. Kumar, T. Meraj, H. T. Rauf, A. A. Alnuaim, and M. A. Alkhayyal, "Guava disease detection using deep convolutional neural networks: A case study of guava plants," *Appl. Sci.*, vol. 12, no. 1, p. 239, Dec. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/12/1/239

[41] P. Wspanialy and M. Moussa, "A detection and severity estimation system for generic diseases of tomato greenhouse plants," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105701, doi: 10.1016/j.compag.2020.105701.

[42] S. Hernández and J. L. López, "Uncertainty quantification for plant disease detection using Bayesian deep learning," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106597, doi: 10.1016/j.asoc.2020.106597.
[43] J. Shin, Y. K. Chang, B. Heung, T. Nguyen-Quang, G. W. Price, and A. Al-Mallahi, "A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves," *Comput. Electron. Agricult.*, vol. 183, Apr. 2021, Art. no. 106042.
[44] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y. Shi, "MobileNet based apple leaf diseases identification," *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 172–180, Aug. 2020, doi: 10.1007/s11036-020-01640-1.
[45] M. B. Tahir, M. A. Khan, K. Javed, S. Kadry, Y.-D. Zhang, T. Akram, and M. Nazir, "Recognition of apple leaf diseases using deep learning and variances controlled features reduction," *Microprocess. Microsyst.*, 2021, doi: 10.1016/j.micpro.2021.104027.
[46] S. Kodors, G. Lacis, O. Sokolova, V. Zhukovs, and I. Apeinans, "Apple scab detection using CNN and transfer learning," *Agronomy Res.*, vol. 19, no. 2, pp. 507–519, 2021. [Online]. Available: http://hdl.handle.net/10492/6447
[47] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*.
[48] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, pp. 1–10, Sep. 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fpls.2016.01419/full
[49] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340, doi: 10.1016/j.dib.2019.104340.
[50] H. T. Rauf and M. I. U. Lali, "A guava fruits and leaves dataset for detection and classification of guava diseases through machine learning," 2021, doi: 10.17632/s8x6jn5cvr.1.
[51] S. Theodoridis, "Chapter 18 - neural networks and deep learning," in *Machine Learning*, 2nd ed., S. Theodoridis, Ed. New York, NY, USA: Academic Press, 2020, pp. 901–1038, doi: 10.1016/B978-0-12-818803-3.00030-1.
[52] Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wang, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3613–3632, 2019, doi: 10.1007/s11042-017-5243-3.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[54] Z. ur Rehman, M. A. Khan, F. Ahmed, R. Damaševičius, S. R. Naqvi, W. Nisar, and K. Javed, "Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture," *IET Image Process.*, vol. 15, no. 10, pp. 2157–2168, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12183, doi: 10.1049/ipr2.12183.
[55] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[56] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Exp.*, vol. 6, no. 4, pp. 312–315, Jan. 2020, doi: 10.1016/j.icte.2020.04.010.

VIBHOR KUMAR VISHNOI (Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from UP Technical University, Lucknow, India, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in computer science with Gurukula Kangri (Deemed to be University), Haridwar, India. From 2014 to 2018, he worked as an Assistant Professor with the Roorkee College of Engineering, Roorkee, India. His research interests include image processing, precision agriculture, machine vision, and deep learning.

KRISHAN KUMAR received the B.Sc. degree in mathematics from MJP Rohilkhand University, Bareilly, India, in 1997, the Master of Computer Application degree from CCS University, Meerut, India, in 2001, and the Ph.D. degree in computer science and information technology from the Institute of Engineering and Technology, MJP Rohilkhand University, in 2010. He is currently working as an Assistant Professor (level 12) with the Department of Computer Science, Gurukula Kangri (Deemed to be University), Haridwar, India. Moreover, he is having 20 years of experience in academics. He has published more than 40 research papers in various national and international journals and proceedings. Furthermore, he has written three books and two book chapters. His research interests include deep learning, natural language processing, image processing, and precision agriculture.

BRAJESH KUMAR (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in physics from C.C.S. University, Meerut, India, in 1996 and 1998, respectively, the M.Tech. degree in computer science and technology from the Indian Institute of Technology Roorkee, India, in 2001, and the Ph.D. degree from the Indian Institute of Technology Kanpur, India, in 2017.

He is currently with the Department of Computer Science and Information Technology, MJP Rohilkhand University, Bareilly, India, as an Associate Professor. His current research interests include image analysis, pattern recognition, remote sensing, precision agriculture, and natural language processing.

SHASHANK MOHAN received the bachelor's degree in computer science and engineering from Dr. A. P. J. Abdul Kalam Technical University, Lucknow, India, in 2019, and the master's degree from Michigan State University, East Lansing, MI, USA, where he is currently pursuing the Doctoral degree. He is also with the Department of Biosystems and Agricultural Engineering, Michigan State University, as a Research Assistant. He is motivated to use the state of the art technologies in computer science to model the natural systems to mitigate the threat of food scarcity and global warming.

ARFAT AHMAD KHAN received the B.Eng. degree in electrical engineering from the University of Lahore, Pakistan, in 2013, the M.Eng. degree in electrical engineering from the Government College University, Lahore, Pakistan, in 2015, and the Ph.D. degree in telecommunication and computer engineering from the Suranaree University of Technology, Thailand, in 2018. From 2014 to 2016, he was an RF Engineer with Etisalat, UAE. From 2018 to 2022, he worked as a Lecturer and a Senior Researcher with the Suranaree University of Technology. He is currently working as a Senior Lecturer and a Researcher at Khon Kaen University, Thailand. His research interests include optimization and stochastic processes, channel and the mathematical modeling, wireless sensor networks, ZigBee, green communications, massive MIMO, OFDM, wireless technologies, signal processing, and advance wireless communications.