The Open Agriculture Journal
DOI: 10.2174/0118743315305194240408034912, 2024, 18, e18743315305194
ISSN: 1874-3315

RESEARCH ARTICLE
OPEN ACCESS

Classification of Various Plant Leaf Disease Using Pretrained Convolutional Neural Network On Imagenet

Geetabai S Hukkeri[1], B C Soundarya[2], H L Gururaj[2,*] and Vinayakumar Ravi[3,*]

[1]*Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India*
[2]*Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India*
[3]*Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia*

Abstract:

Introduction/Background: Plant diseases and pernicious insects are a considerable threat in the agriculture sector. Leaf diseases impact agricultural production. Therefore, early detection and diagnosis of these diseases are essential. This issue can be addressed if a farmer can detect the diseases properly.

Objective: The fundamental goal of this project is to create and test a model for precisely classifying leaf diseases in plants.

Materials and Methods: This paper introduces a model designed to classify leaf diseases effectively. The research utilizes the publicly available PlantVillage dataset, which includes 38 different classes of leaf images, ranging from healthy to disease-infected leaves. Pretrained CNN (Convolutional Neural Network) models, including *VGG16, ResNet50, InceptionV3, MobileNetV2, AlexNet, and EfficientNet*, are employed for image classification.

Results: The paper provides a performance comparison of these models. The results show that the *EfficientNet* model achieves an accuracy of 97.5% in classifying healthy and diseased leaf images, outperforming other models.

Discussion: This research highlights the potential of utilizing advanced neural network architectures for accurate disease detection in the agricultural sector.

Conclusion: This study demonstrates the efficacy of employing sophisticated CNN models, particularly *EfficientNet*, to properly identify leaf diseases. Such technological developments have the potential to improve disease detection in agriculture. These improvements help to improve food security by allowing for preventive actions to battle crop diseases.

Keywords: AlexNet, CNN, EfficientNet, InceptionV3, Leaf disease, Machine learning, MobileNetV2, ResNet50, Transfer learning, VGG16.

Received: January 18, 2024
Revised: March 14, 2024
Accepted: March 19, 2024
Published: May 20, 2024

*Address correspondence to these authors at the Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia; Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India; E-mail: vinayakumarr77@gmail.com and gururaj.hl@manipal.edu

Cite as: Hukkeri G, Soundarya B, Gururaj H, Ravi V. Classification of Various Plant Leaf Disease Using Pretrained Convolutional Neural Network On Imagenet. Open Agric J, 2024; 18: e18743315305194.
http://dx.doi.org/10.2174/0118743315305194240408034912

1. INTRODUCTION

Agricultural vegetation is profoundly impacted by diseases, leading to significant losses in the agricultural economy [1]. Common diseases cause substantial harm to plants, resulting in reduced yields. Agriculture forms the backbone of a nation's economy, with approximately 70-75% of India's population relying on it [2]. Plant leaf diseases not only affect crop quality but also reduce the overall yield. Various factors such as air pollution, soil contamination, water pollutants, temperature fluctuations, insects, microorganisms, and changing climate conditions contribute to the susceptibility of plants to diseases [3].

Farmers, being the backbone of India [3], face challenges in maintaining both the quality and quantity of crops. Traditional disease identification methods, primarily relying on naked-eye observations are time-consuming, expensive, and require significant effort to identify infected leaves [4]. Many farmers lack formal education, which hampers their ability to identify plant leaf diseases promptly and effectively, exacerbating agricultural losses [5, 6]. Accurate identification of these diseases is essential to prevent financial losses and conserve resources.

This paper focuses on accurately diagnosing disease-infected leaves. Deep learning techniques, particularly convolutional neural networks (CNNs) such as pre-trained *AlexNet*, *ResNet50*, *MobileNet*, *VGG16*, and *EfficientNet*, offer the advantage of automatically extracting features from images [7, 8]. Early disease detection greatly benefits agricultural production. Researchers, scientists, and pathologists in India are actively seeking innovative solutions to mitigate agricultural losses [9, 10]. The existing methods face challenges in providing superior accuracy, efficient feature extraction, adaptability to diverse datasets, computational efficiency, scalability, robustness to data quality issues, and interpretable features. In contrast, the *EfficientNet* method introduced in this paper excels in overcoming these limitations, outperforming many existing methods in these aspects.

The main contribution of this paper includes:

1. To propose a model that can classify different (38 in this work) types of plant leaf images and find the classification accuracy.

2. To compare and analyse the performance of the various models in classifying the different species of plant leaf diseases.

3. Based on photos of healthy and diseased leaves, various convolutional neural network (CNN) architectures, including *InceptionV3, ResNetV2, MobileNetV2, and EfficientNetB0*, have been developed.

4. Traditional convolution has been substituted by depth-wise separable convolution in *InceptionV3* and *ResNetV2*, which significantly decreased the number of parameters while maintaining the same performance-accuracy level.

5. The implemented *InceptionV3* and *ResNetV2* architectures are quicker and require fewer parameters than their regular counterparts.

6. On a *MobileNetV2* and an *EfficientNetB0* model, a transfer-learning-based CNN was deployed. We removed certain layers from each model while keeping the weights stable. Then, we added new layers such as activation, batch-normalization, and dense layers. To prevent the model from becoming too specific, we used dropout layers with different values. Additionally, we applied L1 and L2 regularization techniques to simplify the models due to the abundance of features.

In summary, this paper reviews prior research in Section 2, outlines the methodologies used for disease classification comparative analysis in Section 3, presents the results from various models in Section 4, and concludes with suggestions for future enhancements.

2. RELATED WORK

For the identification of leaf diseases based on computer vision, there are two main methods: segmentation and classification. Hyperspectral and LAB-based hybrid segmentation algorithms are used for disease segmentation. Hybrid segmentation combines multiple types of customer segmentation models to form a unique segmentation strategy then the segmented images were trained to ConvNet for image classification. The SegCNN approach can locate sicknesses in plants [4].

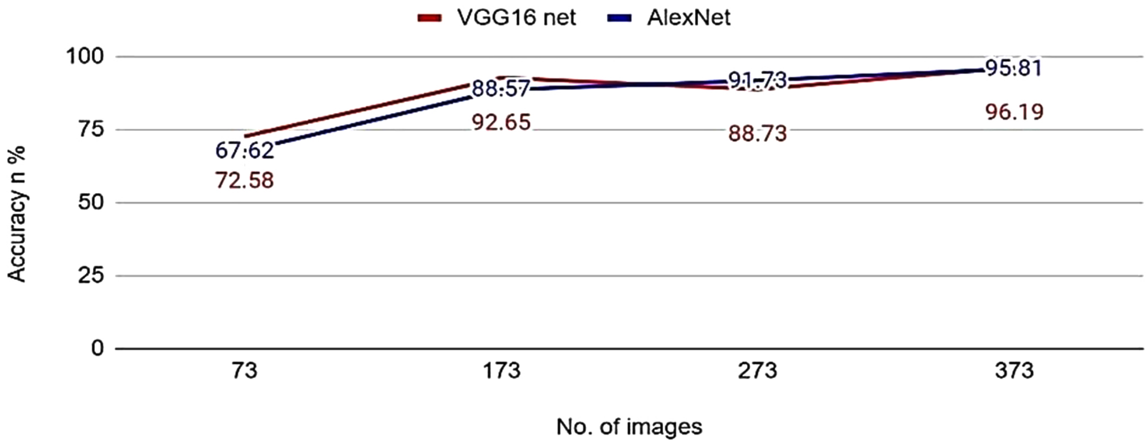


Fig. (1). Classification accuracy of *VGG16* and *AlexNet*.

In classifying guava plant diseases such as Black rot, Algal leaf spot, Rust, and Wilt, the CNN model achieved an accuracy of 70%, and the obtained result was analyzed with the help of a confusion matrix [11, 12].

Artificial Intelligence techniques were used for the classification of healthy or diseased tea leaf images [13]. A multilayer convolutional neural network model was used for classifying the disease-infected region on the mango leaf [14]. For Capsicum leaf disease detection, segmentation was done first, then the feature extraction techniques were used to extract the features of the diseased part. SVM classifier was used for classification [15, 16].

Deep Learning models like *AlexNet* and *VGG16* were employed to classify and detect 13,262 tomato images as diseased or healthy [17, 18]. Using *AlexNet*, an accuracy of 97.49% was achieved, while *VGG16* reached 97.32%. Different parameters such as image count, minibatch size, and learning rates were adjusted to measure performance, with *AlexNet* outperforming *VGG16* in accuracy.

Fig. (**1**) shows the accuracy obtained by the *AlexNet* and *VGG16net* models in classifying different numbers of images. For classifying the 373 images, *AlexNet* achieved accuracies of 95.81% and 96.19%.

DCNN (Deep convolutional neural networks) model was used for estimating the cucumber diseases. The result of DCNN was estimated by conducting a comparative experiment using conventional classifiers [19] such as Random Forest, SVM, and *AlexNet*. The prediction was 97% by using the ResNet34 model in identifying four corn leaf diseases. It consists of a data augmentation technique to generate more data samples [20]. Using deep convolutional neural networks achieved 98.48% accuracy in detecting 38 types of pests and diseases [21]. Deep Learning-based classification ImageNet was to train the model and the 8-layer CaffeNet model was used to detect 13 different species of paddy pests and diseases. The CaffeNet model obtained 80% accuracy in 5,000 iterations and 87% in 3000 iterations [22].

The *ResNet-50*, *InceptionV3* and *MobileNetV2* are CNN-based architectures. The modeling method used transfer learning for comparing experimental data based on the performance of individual models and achieved 91.2% accuracy in classifying the diseases [23]. A *DenseNet* is used for the classification of corn diseases. The DenseNet model uses fewer parameters compared to *VGG19Net*, *NASNet*, and *Xception* and has achieved 98.06% accuracy [24].

For citrus leaf disease detection, two types of CNN architectures used are MobileNet and self-structured convolutional neural network. The accuracy achieved by using the MobileNet CNN model was 98% with 92% validation accuracy at epoch 10, and the accuracy achieved by using SSCNN was 98% accuracy and 98% validation accuracy at epoch value 12 [25]. Image processing techniques with CNN [26] were used to identify 15 different classes of leaf diseases. Leaf disease detection using three DL meta-architectures including the SSD, RCNN, and RFCN was applied [27, 28] to identify 13,842 sugarcane leaf disease identification. Using these techniques achieved 95% accuracy [29].

In this CaffeNet deep learning model was used by researchers which progressively computed the features such as color, shape, and appearance from input images [30]. SSD with Inception module and Rainbow concatenation model is trained to identify 26,377 apple plant leaf diseases. SSD performs well compared to Faster R-CNN [31]. CaffeNet includes eight learning layers, five convolution layers, and three related layers [32].

Based on DenseNet-121 deep convolutional network, the methods applied to detect the apple leaf diseases of 2462 images are regression, multi-label classification, and focus loss function [33]. CNN architectures such as *AlexNet, AlexNetOWTBn, GoogleNet, overfeat and VGG* were trained for the detection and classification of 58 leaf diseases. Among these CNN architectures, VGG achieved good accuracy *i.e.* 99% [34].

Two deep learning models were used, *AlexNet* and *SqueezeNet*. Tomato images were trained on the Nvidia Jetson TX1 which is a supercomputer. *SqueezeNet* is the best deep learning model for mobile deep learning classification [35]. In a study [36] combined *VGGNet* and Residual network (*ResNet*) models were used.

Inception-v3, ResNet-50, VGG-19, and *Xception* models were used for plant leaf disease detection. FT 100% and 75% showed higher classification rates [37]. In this model, UAV images were used for training to classify the soybean leaf disease [38].

3. MATERIALS AND METHODS

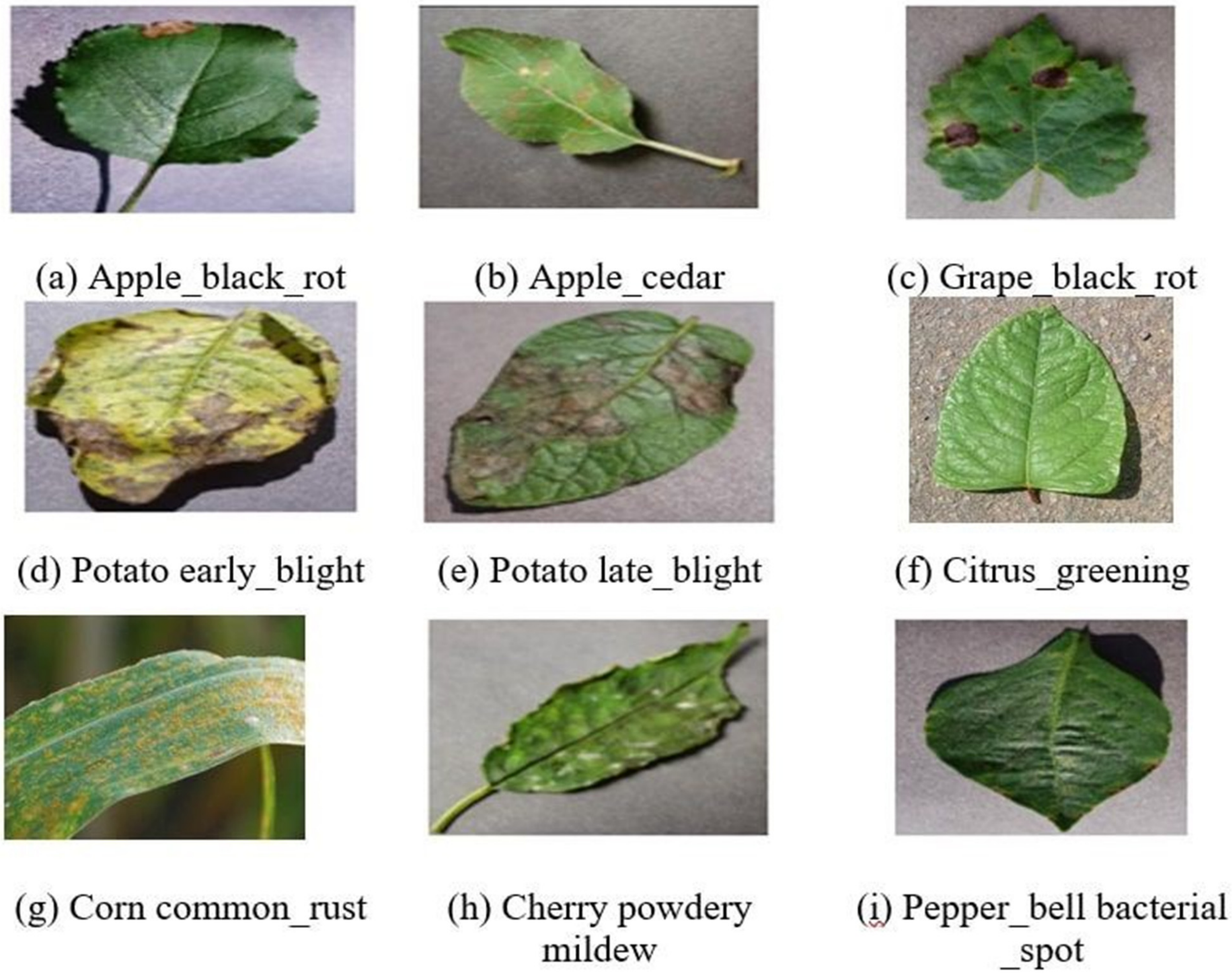
This section discusses the experimental setup for the datasets along with a brief description and steps of the preprocessing applied. Then, a thorough explanation of the architectures of the six models is given.

3.1. Experimental Setup

Our experiments have been performed on Windows 10 OS, with 8 GB RAM, and an Intel core I5 processor. For the implementation of six proposed models (*AlexNet*, *ResNet50*, *VGG16*, *Inception V3, MobileNet V2, EfficientNet)* on datasets, the Google Colab pro, Python version 3.8, PyTorch tool, NVIDIA GTX 1070 GPU, OpenCV 3.4.2. have been used.

3.2. Dataset

In the pre-trained CNN with the Transfer Learning method for various plant leaf disease classifications, PlantVillage dataset is used for training and testing which contains 38 different classes and 14 different plant species in total, among 38 classes 12 of which are healthy, 26 of which are diseased. A balanced dataset has been considered in this work. Grayscaled images of varying sizes have been used for training and testing. The 14 species of crops, including apple, blueberry, cherry, grape, orange, peach, pepper, potato, raspberry, soy, squash, strawberry, Citrus and tomato are included in the dataset as shown in Fig. (**2**).


(a) Apple_black_rot

(b) Apple_cedar

(c) Grape_black_rot

(d) Potato early_blight

(e) Potato late_blight

(f) Citrus_greening

(g) Corn common_rust

(h) Cherry powdery mildew

(i) Pepper_bell bacterial _spot

Fig. (2a-i). Healthy and diseased leaf images of the different plant [9].

In this work, the comprehensive dataset consisting of 24,305 snapshots has been collected from the Kaggle website. For our experiments, 17,121 snapshots were utilized for training purposes, 6,176 for testing, and 1,008 for validation. The models were trained over a span of 30 epochs.

Images were collected from https://www.kaggle.com/abdallahalidev/plantvillage-dataset.

Pre-trained models are used with the identical optimization approach as used in the training of the ImageNet dataset. Accordingly, the *VGG16* version uses the SGD (Stochastic Gradient Descent) optimization approach, and all different models which include *AlexNet*, *Inception V3*, *ResNet50*, *MobileNet V2* and

Table 1. Values used for pre-trained CNN models.

Models	Input Ratio	Optimzation Method	Learning_rate
VGG16	224*224	SGD	0.001
AlexNet	227*227	Adam	0.001
MobileNet V2	224*224	Adam	0.001
EfficientNet	227*227	Adam	0.001
Inception V3	256*256	Adam	0.001
ResNet50	224x224	Adam	0.001

EfficientNet use the Adam Optimization Technique. The learning charge for the Adam approach was decided on as 0.001 and for the SGD method, it turned into the set to 0.01 as shown in Table **1**.

For all the fashions, the validation step turned into a set of at least one. In the observation of pre-trained CNN with transfer learning approach for various plant leaf disease detection, every pixel in the unique datasets is normalized dividing *via* 255. Images had been set to 227x227 pixels for the *AlexNet* model, 224x224 pixels for the *ResNet50* version and *VGG16* model, and 299x299 pixels for the Inception V3 model. Based on the hardware obstacles enter image resolutions had been resized for all fashions of the *EfficientNet* structure.

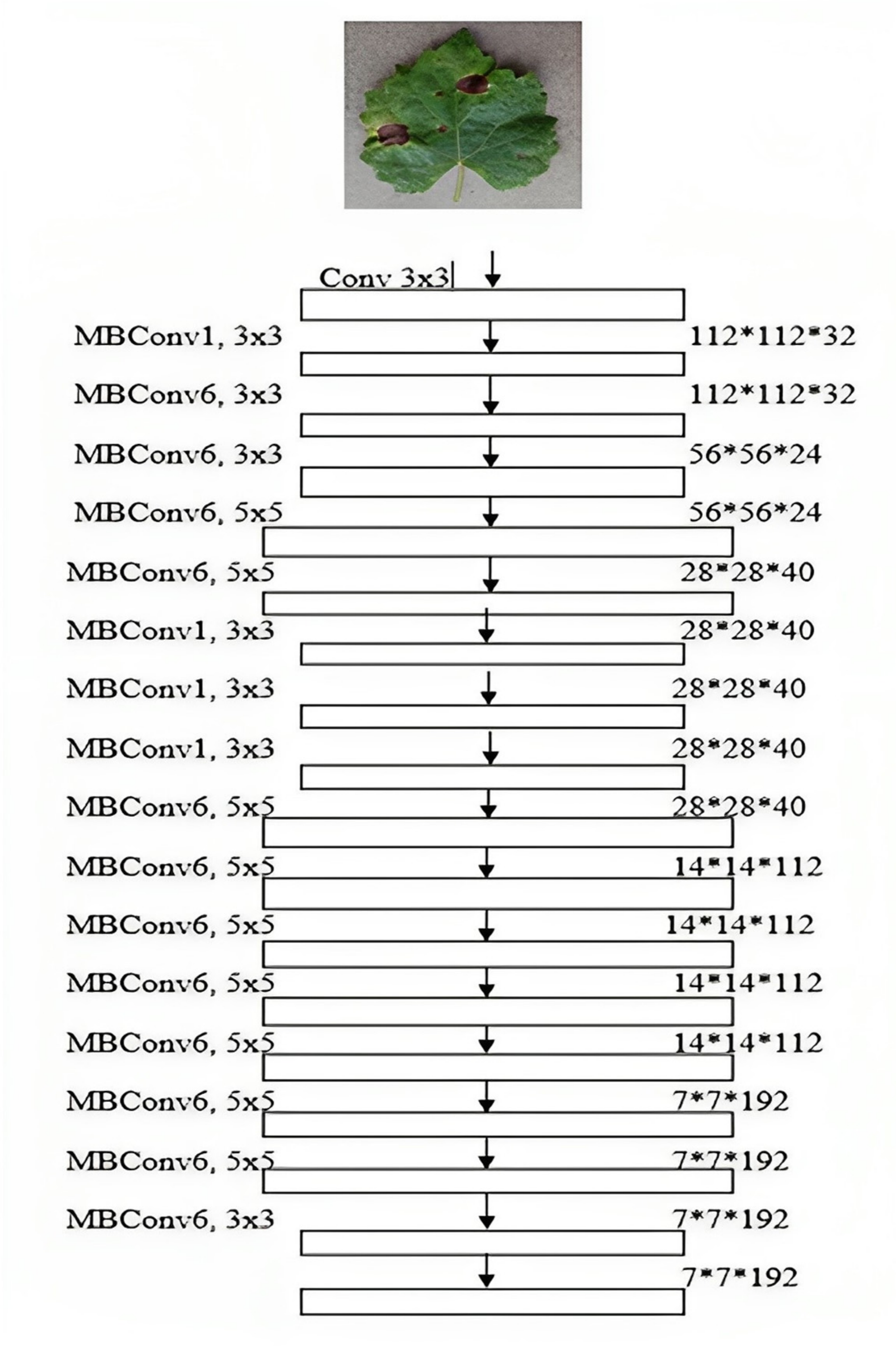


Fig. (3). Schematic representation of *efficientnet* architecture.

Deep learning architectures are intended to expose efficient techniques with smaller fashions. *EfficientNet*, in contrast to other contemporary fashions, achieved more efficient outcomes by scaling intensity, width, and backbone while scaling down the model. The initial step within the compound scaling approach is to look for a grid to find out the relationship between a few of the precise scaling dimensions of the baseline community underneath a set useful resource constraint.

In MBConv, blocks encompass a layer that first expands and then compresses the channels, so direct connections are used amongst bottlenecks that are a part of plenty fewer channels than boom layers. So direct connections are used between bottlenecks that join far fewer channels than expansion layers. This structure incorporates depthwise separable convolutions, which significantly reduce computations by a factor of k compared to traditional layers, where k represents the kernel size, denoting the width and height of the 2D convolution window. The representation of the *EfficientNet* structure is proven in Fig. (**3**).

The steps involved in the proposed framework for various plant leaf disease classifications are shown in Fig. (**4**).

3.2.1. Data Preprocessing

Data pre-processing is a necessary step for preparing raw data to construct and train models, which also enhances accuracy [5]. It helps improve the quality of data to facilitate the extraction of meaningful insights. NumPy, Pandas and Matplotlib are the core libraries for pre-processing. Before running the method, the dataset is pre-processed to check for missing values, noisy data, and other irregularities. For Machine Learning, the data should be in a stunning structure.

3.2.2. Feature Extraction

Features can be primarily based on coloration, shape, and texture. Nowadays, most researchers are applying texture functions for the detection of plant diseases. The convolutional layers collect features from the resized photographs [8]. ReLU is completed after convolution and precise varieties of pooling.

3.2.3. Classification

This is to decide if the entered plant leaf image is healthy or unhealthy. Classification uses completely associated layers and for characteristic extraction, it uses convolutional and pooling layers [8].

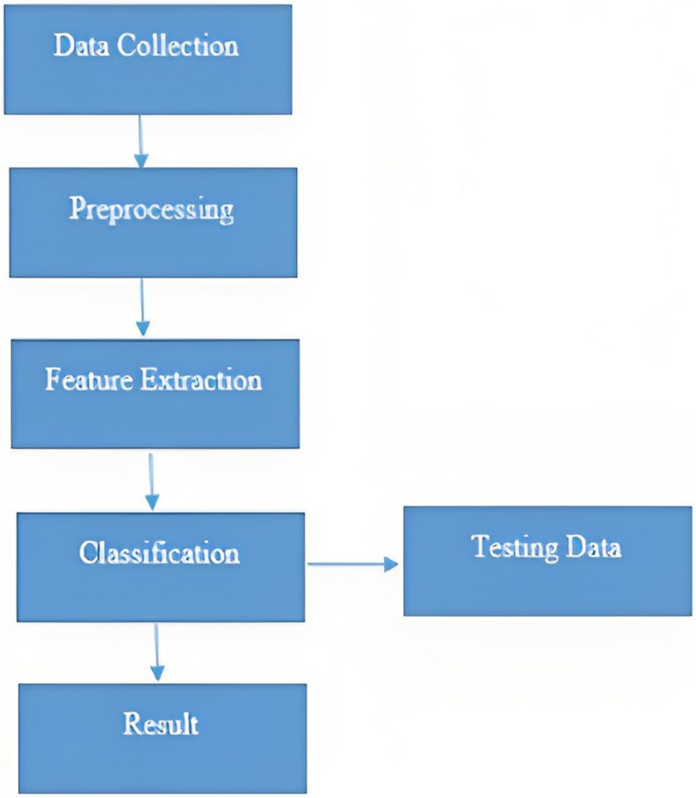


Fig. (4). Flowchart for disease classification using traditional feature-based approaches.

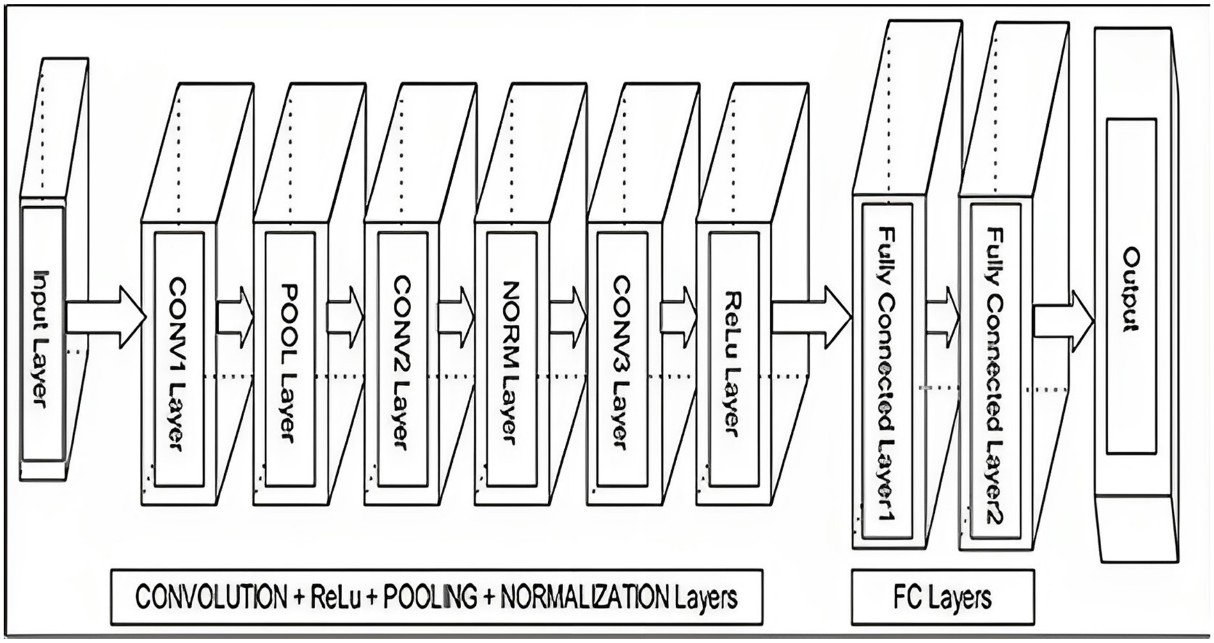


Fig. (5). Schematic presentation of utilized CNN model architecture used for leaf disease detection.

3.3. CNN Structure Design

CNN processes unstructured image inputs and accurately assigns them to the appropriate output classes [15]. It has been discovered to be powerful and is moreover the most broadly applied in numerous programs of computer vision. CNN operates on unorganized picture inputs and transforms them into the right output classes [15]. In this study, an image shape has been built for this set of rules. This form is crafted from several layers as shown in Fig. (5), which illustrates the structure that we used to construct the CNN.

3.3.1. Convolution Layers

The convolutional layer is the constructing block of CNN. It is likewise referred to as the Conv layer. The features of the image can be obtained which have a set of automatic learnable parameters (weights). Filters contain the matrix with (MxMx3) dimension. The result is the 2D matrix of the feature map of mathematical operation. By way of constructing the characteristic maps of all filters alongside the vertical height size, the outcome length of the conv layer can be obtained. The output of this layer in CNNs can be written as Eq. (**1**),

$$X_j{}^p = \sum_{i \in N_j} X_i^{p-1} x k_{{ij}^{p+B_j}{}^p} \quad \textbf{(1)}$$

where p^{th} layer is represented by p, Nj denotes the number of filters, X_i^{p-1} denotes feature map, K_{ij} denotes convolutional kernel, and B_j denotes bias term.

3.3.2. Non-Linear Layers

It is brought after each convolution layer. It includes an activation feature that produces non-linearity at the input facts, as a result enhancing the generalization capability of the model even better. It creates an activation map as an output. ReLU is a popularly used activation function. The result is 0 for the negative enter values in any other case, producing the enter values of the x matrix as it's far. The result can be written as below,

$$f(x) = \{0,\ x < 0\ x,\ x \geq 0 \quad \textbf{(2)}$$

3.3.3. Pooling Layers

Convolutional neural networks' building blocks include pooling layers. Pooling layers combine the features discovered by CNNs, whereas convolutional layers retrieve features from images. Its goal is to gradually reduce the spatial dimension of the representation to reduce the number of parameters and calculations in the network. Pooling layers are used to progressively deprecate the spatial period (top and width) of each feature map and hold the intensity intact. It minimizes the number of parameters to analyse which in turn enables to lower the hazard of overfitting and quantity of computation finished within the community. Pooling layers are not educated during backpropagation. The following are the types of pooling layers:

3.3.3.1. Max Pooling

It is normally the usage of pooling operation in most neural networks. This operation consumes the extra price from the characteristic map overlaid by the filter. A max-pooling normally picks out the pool length size as 2 x 2 with a stride of 2x2. Overfitting can be reduced by using the input length of the top and width of each function map.

3.3.3.2. Global Average Pooling (GAP)

The linked layer in classical CNNs can be reduced by pooling operation. It is likewise referred to as the GAP layer and it performs a maximal kind of downsampling, in which the characteristic map with dimensions h×w×d is contracted to have dimensions 1×1×d array by taking the common of all of the factors in each function map. Overfitting can be reduced by reducing the entire quantity of parameters within the model. It makes the model extra robust for spatial translations.

3.3.3.3. Fully Connected Layer

It is also called a dense layer. The nodes within the linked layer have all to all verbal exchanges with the preceding and subsequent layers. The output of the final convolutional might be exceeded as entering to this accretion.

$$F(x) = relu(xw^T\ B) \tag{3}$$

where the input data is represented by x, w denotes the weight vector and bias term is represented by B.

3.3.3.4. Output Layer

The output layer is the layer that has complete connectivity with the previous layer and receives the input from it. It makes use of softmax activation for predicting the target output (elegance) with excessive opportunity. The softmax can be mathematically written as,

$$S(z^{i}) = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}}\ for\ i\ =\ 1, \dots, n \tag{4}$$

where, the number of classes is represented by n, and the value of (Z^i) is always a positive value within the range of (0,1). The numerator value is the input to the denominator and added with other positive numbers, the numerator takes any real value from the input vector. The value 1 can be summed up to output probability values.

3.4. AlexNet

A key feature of *AlexNet*'s design includes 8 convolutional layers, comprising 5 convolutional layers and 3 artificial neural network (ANN) layers. Each convolutional layer is immediately followed by a max-pooling layer, contributing to its straightforward architecture. Notably, *AlexNet* utilizes Rectified Linear Unit (ReLU) activation, a non-linear function, which differs from the commonly used sigmoid or tanh functions in deep learning networks. Unlike sigmoid and tanh, ReLU activation prevents quick saturation during training, ensuring effective learning especially when utilizing GPUs. This unique choice of activation function enhances the model's training efficiency. Moreover, *AlexNet* incorporates overlapping max pooling. The input to this model consists of images with dimensions 227X227X3 [7].

3.5. VGG16

VGG16 is also referred to as an *OxfordNet* model. It comprises 16 weight layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers primarily use small 3x3 kernels, and max-pooling layers with 2x2 filters follow most convolutional layers. This design choice of using small-sized kernels enables the model to learn intricate patterns in the data. The consistent use of 3x3 kernels provides a deeper network while maintaining a receptive field, enhancing the model's ability to capture complex features. One notable aspect of *VGG16* is its substantial number of parameters, totaling almost 138 million. The input images fed into the *VGG16* model are typically resized to 224x224x3 pixels as shown in Fig. (**6**). This standardized input size is compatible with the architecture's design, ensuring uniformity in the processing of images.

3.6. MobileNet V2

MobileNet V2 is green for cellular devices. With the *MobileNet* version, users can benefit from a brilliant output with mathematical operations and lower parameters. *MobileNetV2* includes three convolutional layers. The growth layer is the primary layer with dimensions 1x1 as shown in Fig. (**7**). The goal of the expansion layer was to increase the records before going into intensity clever convolution. In this deposit number of input, channels are lesser as compared to the number of output channels. The working system is opposite to the projection layer. The growth factors can be used to outline information to amplify the amount. The enlargement element is six. The primary layer converts the channel into an ultra-modern new tensor with 10*6=60 channels when a tensor has 10 channels.

3.7. Inception V3

Google evolved the *Inception V3*, which is the third launch in the Deep Learning evolutionary architectures collection. Batch normalization has become finished in Inception V2. In *Inception V3*, the concept of factorization has been added. We can minimize the number of connections and parameters without lowering the performance of the community using factorization. The schematic representation of *Inception V3* is shown in Fig. (**8**). The model includes common pooling, max pooling, dropouts, and connected layers for *Inception V3*, in the remaining layer which has the Softmax characteristic, consists of 42 layers in trendy and the input layer takes images of 299x299 pixels.

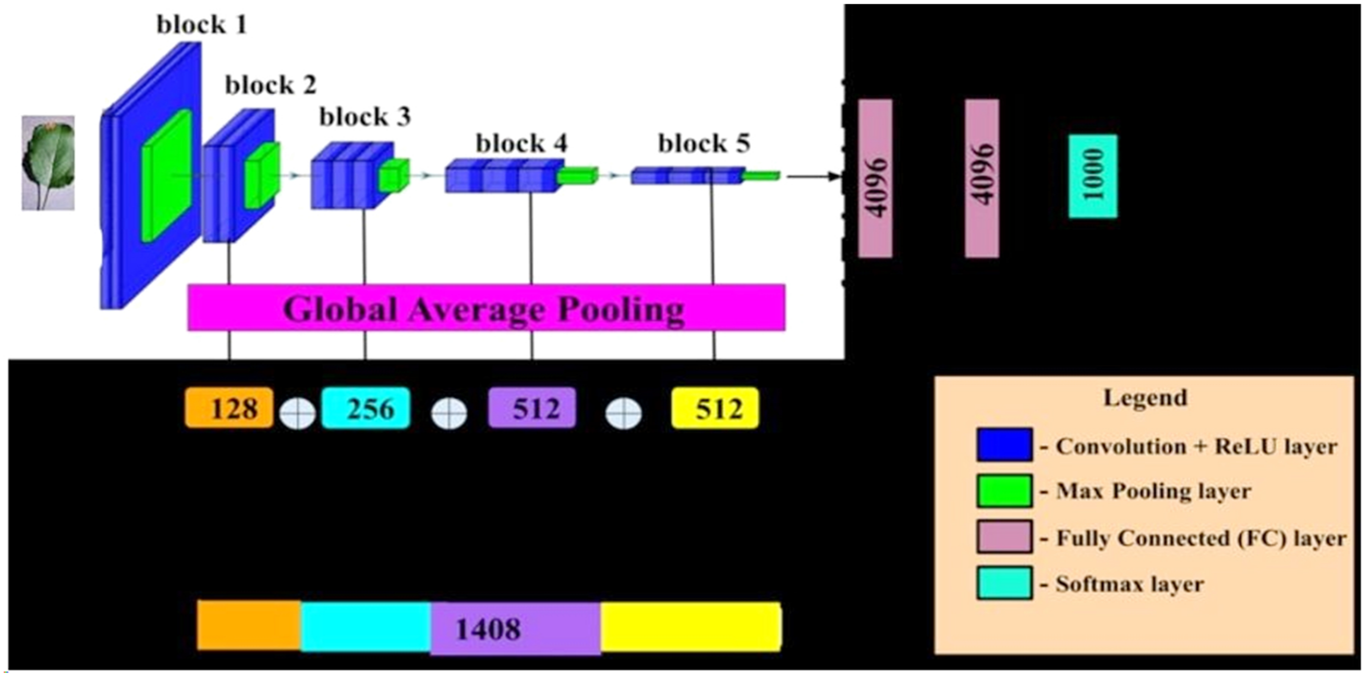


Fig. (6). Schematic representation of *VGG16*.

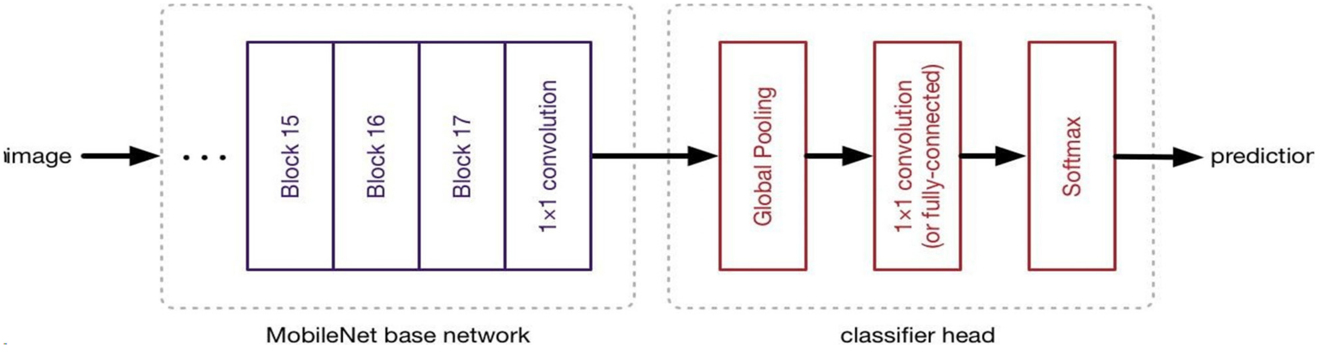


Fig. (7). Schematic representation of mobilenet.

3.8. ResNet50

ResNet50 presents a significant advancement in the realm of deep learning, especially in applications like plant leaf disease detection. At its core, *ResNet50* addresses the issue of vanishing gradients and the degradation problem faced by deep networks with numerous non-linear layers. Traditional deep networks often struggle to learn meaningful identity mappings when multiple layers are stacked together [1]. *ResNet50* pioneers the use of residual connections, allowing these networks to learn identity mappings effectively. This innovative approach ensures that even as networks deepen, they remain trainable and do not suffer from diminishing gradient problems. The architecture of *ResNet50* is built upon the concept of residual blocks, each containing multiple stacked residual units. These residual units, or residual blocks, facilitate the flow of gradients during backpropagation, enabling the successful training of deep networks. Fig. (9) illustrates the network's structure, showcasing its stacked residual units. Similar to *VGG16*, *ResNet50* employs 3x3 filters, capturing intricate patterns within the data. The input images processed by *ResNet50* are standardized to dimensions of 224x224 pixels. This consistency in input size ensures uniformity in processing and enables the network to effectively analyze plant leaf images for disease detection tasks.

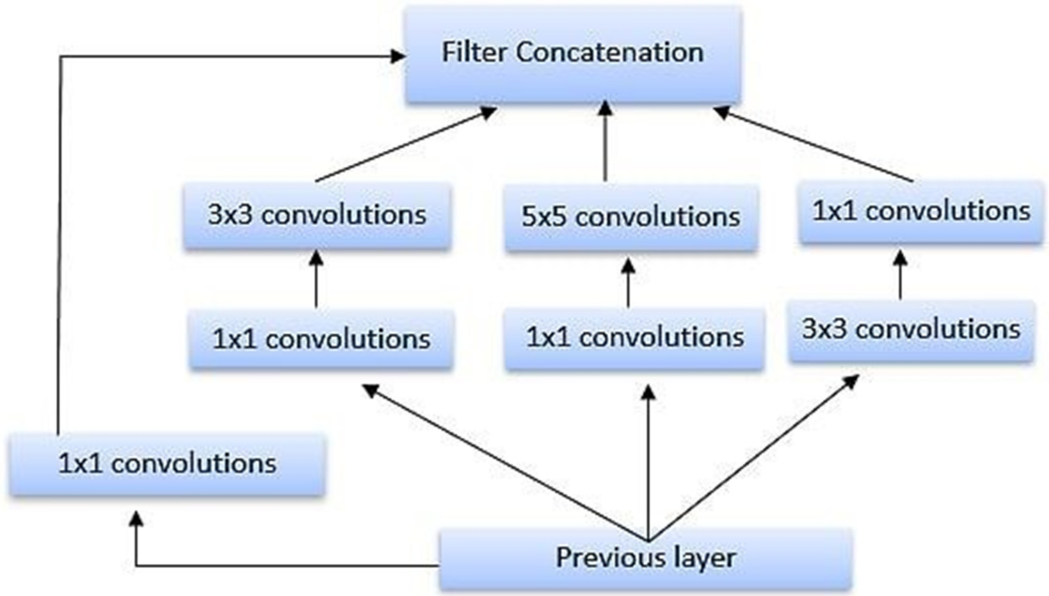


Fig. (8). Schematic representation of inception V3.

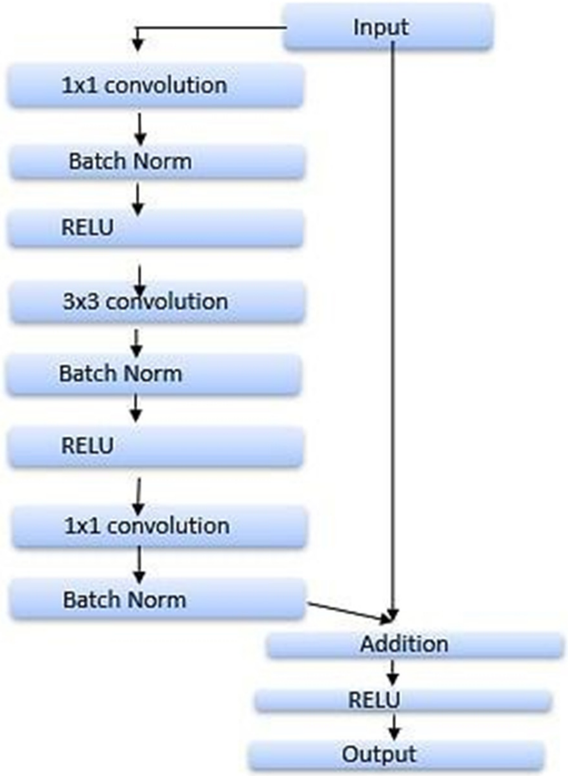


Fig. (9). Schematic representation of *ResNet50*.

Table 2. Deep learning architecture features.

Deep Learning Model	Key Features
LetNet	Parameters are compared to other CNN models. The capability of computation is limited.
AlexNet	Considered the first modern CNN. Better performance can be achieved using ReLu. Overfitting can be avoided using Dropout techniques.
OverFeat	The larger set of parameters as compared to *AlexNet*. The first version is used for the detection, localization, and category of objects through CNN.
VGG	3 × 3 receptive fields had been considered to include more wider variety of non-linearity functions which made selection characteristic discriminative.
GoogLeNet	Few parameters are compared to the *AlexNet* model. Better accuracy at its time.
DenseNet	There is a Dense connection between the layers. It is possible to achieve better accuracy with lower parameters.
SqueezNet	Considered 1 × 1 filters in place of 3 × 3 filters. Similar accuracy as *AlexNet* with 50 times lesser parameters.
Xception	A depth-sensible separable convolution technique. Performed higher than VGG, ResNet, and Inception-v3.

Table **2** gives an overview of the differences of other deep learning models. This key feature helps to choose a good model for classification to achieve good accuracy.

Accuracy is the percentage of correctly categorized images in all samples. "The ratio of correct predictions to all predictions made" is called accuracy and it is shown in Eq. (**5**) [39, 40].

$$Accuracy = \frac{True\ positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad \textbf{(5)}$$

Precision is the measure of accurately categorized positive samples (True Positive) relative to the total correctly categorized data. It reflects how relevant the discovered objects are in a model. Precision is calculated by dividing the actual positives by the total positives, as defined in Eq. **6** .

$$Precision = \frac{True\ Positive}{true\ Positive + true\ Negative} \quad \textbf{(6)}$$

Recall, also known as sensitivity, measures the accuracy of instances predicted positively, indicating how many were correctly identified. It represents the proportion of all relevant results that the algorithm accurately recognized as relevant. Recall is calculated using Eq. **7** [39].

$$\text{Recall} = \frac{True\ positive}{True\ Positive + False\ Negative} \quad \textbf{(7)}$$

The F1-Score is a fundamental evaluation metric in machine learning that combines precision and recall, two metrics often in conflict. It provides a summary of a model's predictive power. The F1-Score is calculated using Eq. **8** .

$$\text{F1} = \frac{2 \times Precision \times Recall}{Precision \times Recall}. \quad \textbf{(8)}$$

4. RESULTS AND DISCUSSION

The objective of the developed model is to reduce plant damage. Farmers can also minimize costs by solving the problem themselves at an early stage. This study compared the performance of six architectures, including *VGG16*. The major goal of this section is to compare the results of state-of-the-art models in the literature with the performance of *EfficientNet*'s deep learning architecture in classifying plant leaf disease. The dataset is applied for all experimental research. In this section experimental results obtained from various models in classifying 38 different classes of leaf diseases are discussed. To evaluate the model performance, parameters like Precision, Accuracy, Recall and F1 score were considered. Table **3** shows the comparison of test accuracy data of the six models.

Table 3. Accuracy obtained for various models.

Models	*VGG 16*	*AlexNet*	*MobileNet V2*	*InceptionV3*	*ResNet50*	*EfficientNet*
Accuracy in %	91.5	87.1	93.5	93.8	89.7	**97.5**
Epochs	13	22	9	18	25	5

Using the various models, an attempt has been made to overcome the leaf disease. Initial lower layers of the network learn very generic features from the pre-trained model. To achieve this initial layer's weights of pre-trained models were frozen and not updated during the training. Higher layers are used for learning task-specific features. Higher layers of pre-trained models are trainable or fine-tuned.

EfficientNet excels in classifying plant leaf diseases due to its parameter efficiency, effective feature extraction capabilities, and adaptability to diverse image variability. With its ability to achieve remarkable accuracy using fewer parameters, it enables faster training and reduced memory usage. The architecture is adept at capturing both low-level and high-level features essential for disease classification. The incorporation of advanced regularization techniques like dropout and batch normalization prevents overfitting, ensuring the model generalizes well to new and unseen data. Overall, *EfficientNet*'s optimized design, combined with its ability to handle diverse leaf characteristics and its regularization methods, contributes to its superior performance in accurately classifying plant leaf diseases, making it a valuable tool in agricultural research and disease management efforts.

The result in this study shows that the *EfficientNet* achieves the best result *i.e.*, 97.5% compared to the other models. If the model is trained for a greater number of epochs *i.e.*, 100 or 200, the higher accuracy can be achieved.

The model *VGG16* has achieved 91.5% accuracy, *AlexNet* has achieved 87.1% accuracy, *MobileNetV2* has achieved 93.5% accuracy, *Inception V3* has achieved 93.8% accuracy, *ResNet50* has achieved 89.7% accuracy and *EfficientNet* has achieved 97.5% accuracy. Compared to all other models, *EfficientNet* model has achieved the highest accuracy. The models were trained for 30 epochs. The lowest training time per epoch was achieved by the *AlexNet* model.

Fig. (**10**) shows the graphical representation of accuracy obtained from various models. It shows the comparison between various models such as *VGG16*, *ResNet50* and so on. Finally, it concludes that *EfficientNet* model has performed well by obtaining an accuracy of 97.5% in classifying the various disease-infected plant leaves.

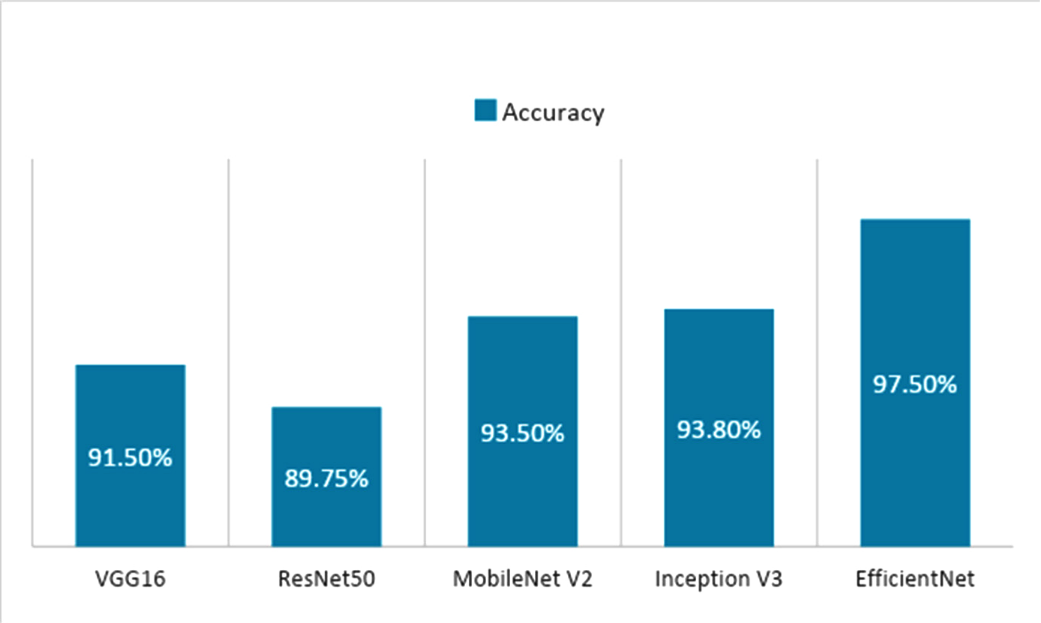


Fig. (10). Accuracy representation.

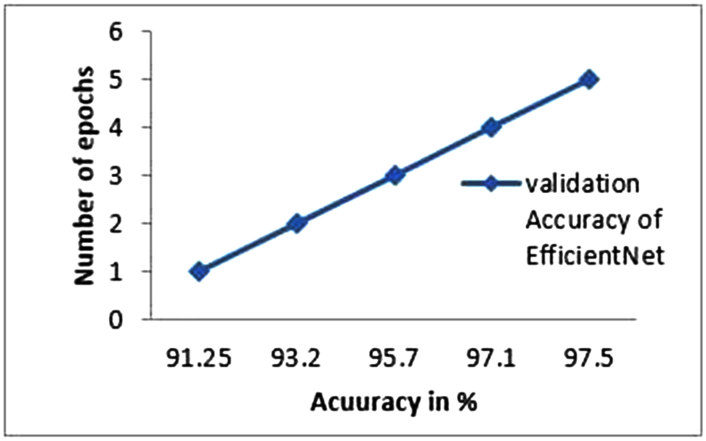


Fig. (11). Validation accuracy.

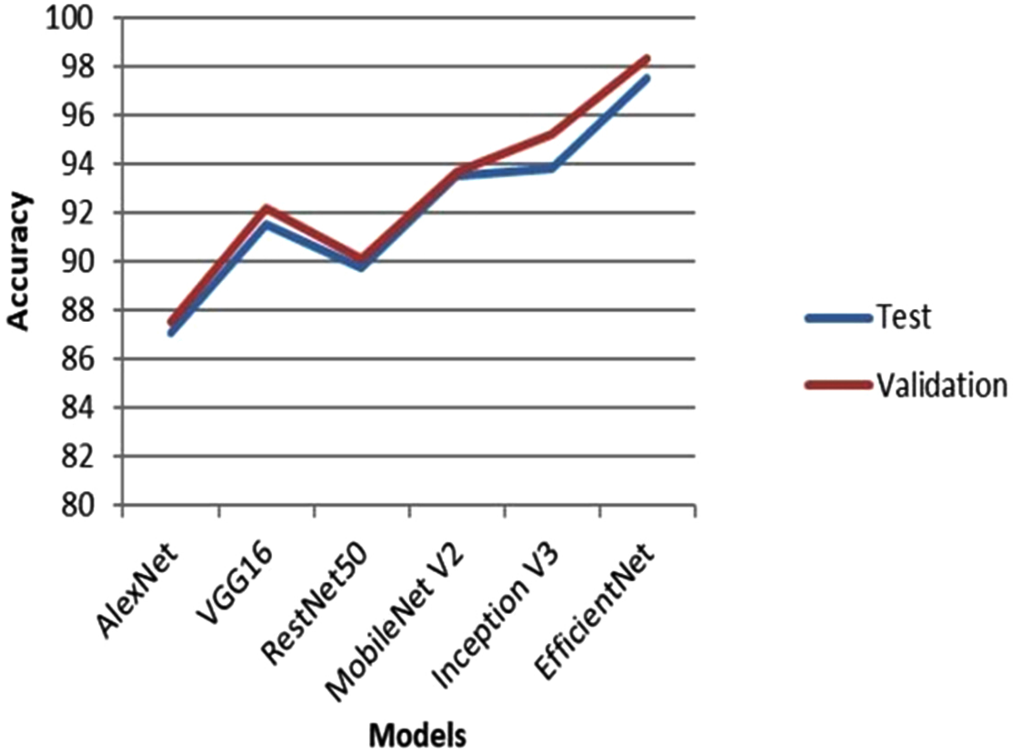


Fig. (12). Test and validation accuracy.

Fig. (**11**) shows the graphical plot of the validation accuracy obtained by the *EfficientNet* model per epoch. The top 1 accuracy obtained by the *EfficientNet* model is 91.25%. The top-5 accuracy obtained from the *EfficientNet* model is 97.5%. It can be observed that for each epoch model performance was improved.

Fig. (**12**) shows the graphical plot of the comparison of test and validation accuracy between various models such as *EfficientNet*, *ResNet50*, *MobileNetV2*, *AlexNet*, *VGG16* and *Inception V3* model w.r.t epochs. *EfficientNet* model has achieved the highest test accuracy compared to all other models.

Table **4** shows the results of comparing accuracy for various transfer learning architectures. This study concludes that the *EfficientNet* model has achieved the highest accuracy of 97.5% with a precision of 94%, Recall of 93%, and with F1 score of 93%, in classifying 24305 datasets compared to various models. The Precision, Accuracy, Recall and F1-score are calculated using the Eqs. (**5**-**8**).

Table **5** compares the accuracy obtained by various models in classifying a different dataset. This study concludes that the *EfficientNet* model has achieved the highest accuracy of 97.5% with a precision 94%, Recall 93% and with F1-score of 93%, in classifying 24305 datasets compared to various models.

4.1. Discussion

The discussion section of this study delves into the elaborate technical details regarding the experimental setup, dataset characteristics, model architectures, and performance evaluation of various deep learning models for plant leaf disease classification.

The experiments were meticulously conducted on a Windows 10 OS platform, leveraging 8 GB RAM and an Intel Core i5 processor. For model implementation, Google Colab Pro, Python version 3.8, PyTorch, NVIDIA GTX 1070 GPU, and OpenCV 3.4.2 were meticulously employed.

The utilization of the PlantVillage dataset, comprising 38 different classes and 14 plant species, sourced from Kaggle, provided a robust foundation for training and testing, comprising 24,305 snapshots. The dataset underwent meticulous balancing over 30 epochs to ensure unbiased model learning. Rigorous preprocessing techniques were meticulously applied to enhance dataset integrity, including checks for missing values, noise, and irregularities. Feature extraction procedures were elaborately designed to encompass color, shape, and texture-based features, with convolutional layers adeptly extracting meaningful features from resized images.

Table 4. Comparison of pretrained weights on results.

Metrics	*AlexNet*	*ResNet50*	*VGG16*	*Inception V3*	*MobileNet V2*	*EfficientNet*
Accuracy	0.871	0.897	0.915	0.938	0.935	0.975
Precision	0.82	0.84	0.87	0.89	0.89	0.94
Recall	0.84	0.84	0.89	0.86	0.91	0.93
F1	0.83	o.84	0.88	0.90	0.91	0.93

Table 5. Comparison of proposed plant leaf disease classification method to other methods.

Models/Refs.	Dataset	Species	Classification Performed	Accuracy in %
AlexNet [6]	Plant Village	Tomato	Yes	95.4
CNN [12]	Internet	Guava	Yes	70
Multilayer CNN [14]	Real environment	Mango	Yes	83.7
SVM [15]	Plant Village	Capsicum	Yes	93.2
AlexNet , *VGG16* [17]	Internet	Tomato	Yes	97.49 *AlexNet*
CaffeNet [22]	Plant Village	Multiple	Yes	80
MobileNet SSCN [25]	Internet	Citrus	Yes	98 MobileNet
DCNN [19]	Internet	Cucumber	Yes	96.4
Proposed Proposed models *(AlexNet, ResNet50, VGG16, Inception V3, MobileNet V2, EfficientNet)*	**Plant Village**	**14 specvies**	**Yes**	**97.5** **EfficientNet**

Model architectures, including AlexNet, ResNet50, VGG16, InceptionV3, MobileNetV2, and EfficientNet, were carefully selected to represent diverse deep-learning frameworks. Each model was finely tuned for the plant leaf disease classification task, employing optimization techniques such as SGD and Adam. Learning rates were meticulously optimized to ensure efficient convergence during training. Transfer learning methodologies, combined with pre-trained CNN models, facilitated intricate feature extraction from plant leaf images, resulting in robust disease classification outcomes.

CONCLUSION

In this paper, *EfficientNet* deep learning architecture is proposed to categorize the leaf images of 38 classes inside the Plant Village datasets. The results of the proposed method's performance have been compared and analysed with various models. Despite the extensive research conducted in the field of plant leaf disease classification, our proposed *EfficientNet* model excels notably in accuracy when compared to the models discussed in the related literature. Moreover, in *EfficientNet* architecture, even though the input photograph size is resized to 132x132, it yields greater success attaining an accuracy of 97.5% than other models. Early detection allows farmers to identify diseased plants before symptoms become visible to the naked eye. This enables timely intervention measures, such as targeted pesticide application or removal of affected plants, preventing the spread of diseases to healthy plants.

In future work, we intend to work on more leaf datasets using recent algorithms. This will help to make extra correct predictions in hard environments. With the deployment of these ideas, both agricultural scientists and farmers will be capable of speedy understanding of plant sicknesses and taking critical vital precautions.

LIST OF ABBREVIATIONS

CNNs	=	Convolutional neural networks
DCNN	=	Deep convolutional neural networks
GAP	=	Global Average Pooling
ANN	=	Artificial neural network
ReLU	=	Rectified Linear Unit

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data and supportive information is available within the article.

FUNDING

None.

CONFLICT OF INTEREST

The authors declare no conflict of interest financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Chen J, Chen J, Zhang D, Sun Y, Nanehkaran YA. Using deep transfer learning for image-based plant disease identification. Comput Electron Agric 2020; 173: 105393.

http://dx.doi.org/10.1016/j.compag.2020.105393

[2] Jepkoech J. Arabica coffee leaf images dataset for coffee leaf disease detection and classification. Data Brief 2021; 36: 107142.

[3] Chowdhury Muhammad EH. Automatic and reliable leaf disease detection using deep learning techniques. Agri Eng 2021; 3(2): 294.

[4] Angel Sheril J, Mary Eugine J, Dikshna U. Deep learning based disease detection in tomatoes. 3rd International Conference on Signal Processing and Communication (ICPSC). 13-14 May 2021; Coimbatore, India. 2021.

[5] Nanehkar Y A, Zhang Defu, Chen Junde, Yuan Tian. Recognition of plant leaf diseases based on computer vision. In . 1-5. J Amb Intell Humanized Comput 2020; 2020: 1-5.

[6] Ashok S, Kishore G, Rajesh V. Tomato leaf disease detection using deep learning techniques. Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES). 10-12 June 2020; Coimbatore, India. 2020. http://dx.doi.org/10.1109/ICCES48766.2020.9137986

[7] Moajjem CH, Shuzan IMN, Muhammad CEH, Mahbub ZB, Uddin MM, Amith K. Estimating blood pressure from the photoplethysmogram signal and demographic features using machine learning techniques. MDPI Sensors 2020; 3127: 1-24.

[8] Rahman T, Khandakar A, Kadir MA, *et al.* Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access 2020; 8: 191586-601. http://dx.doi.org/10.1109/ACCESS.2020.3031384

[9] Agarwal M, Singh A, Arjaria S, Sinha A, Gupta S. ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Comput Sci 2020; 167: 293-301. http://dx.doi.org/10.1016/j.procs.2020.03.225

[10] Karthik R, Hariharan M. Attention embedded residual CNN for disease detection in tomato leaves. ApplSoft Comput 2019; 86(4): 105933.

[11] Hukkeri GS, R H G. Detection of text from lecture video images using CTPN algorithm. Int J Eng Trends Technol 2022; 70(1): 180-4. http://dx.doi.org/10.14445/22315381/IJETT-V70I1P220

[12] Mostafa M. Guava disease detection using deep convolutional neural networks: A case study of Guava Plants. Appl Sci 2021; 12(1): 12010239.

[13] Somanth M, Munti P, Ramen P. Tea leaf disease detection using multi-objective image segmentation. Multimedia Tools Appl 2021; 80(1): 1-21.

[14] Meher JK. Disease detection in infected plant leaf by computational method. Arch Phytopathol Pflanzenschutz 2020; 2020: 1-11.

[15] Anjna A. Hybrid system for detection and classification of plant disease using qualitative texture features analysis. Procedia Comp Sci 2020; 167: 1056-65.

[16] Indra MP. An efficient citrus canker detection method based on contrast limited adaptive histogram equalization enhancement. Int J Appl Eng Res 2018; 13: 809-15.

[17] Shantkumari M, Uma SV. Grape leaf segmentation for disease identification through adaptive Snake algorithm model. Multimedia Tools Appl 2021; 80(6): 8861-79. http://dx.doi.org/10.1007/s11042-020-09853-y

[18] Zhang K, Xu Z, Dong S, Cen C, Wu Q. Identification of peach leaf disease infected by Xanthomonas campestris with deep learning. Eng Agric Environ Food 2019; 12(4): 388-96. http://dx.doi.org/10.1016/j.eaef.2019.05.001

[19] Vengateshwaran M, Kalaimani EVRM. Deep learner based earlier plant leaf disease prediction and classification using machine learning algorithm. IOSR J Eng 2018; 5: 45-51.

[20] Rangarajan AK. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comp Sci 2018; 133: 1040-7.

[21] Hu R, Zhang S, Wang P, Xu G, Wang D, Qian Y. The identification of corn leaf diseases based on transfer learning and data augmentation. Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering. 58-65. http://dx.doi.org/10.1145/3403746.3403905

[22] Ahmad AA, Chen Q, Guo M. Deep learning based classification for paddy pests and diseases recognition. Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence. 21-5.

[23] Changhua L, Long Y. Research on deep learning method of crop disease identification. Proceedings of the International Conference on Artificial Intelligence, Information and Cloud Computing 2019; 19-21.

[24] Kalaivani S. Agricultural leaf blight disease segmentation using indices based histogram intensity segmentation approach. Multimedia Tools Appl 2020; 79(2): 1-15.

[25] Navdeep S, Hiteshwari S. Convolutional neural networks-an extensive arena of deep learning. a comprehensive study. Arch Comput Methods Eng 2021; 28: 1-30.

[26] Marwan JA, Jamal A-TM. Plant leaf diseases detection and classification using image processing and deep learning techniques. 2020 International Conference on Computer Science and Software Engineering (CSASE). 259-65.

[27] Rohit N. Agriculture field monitoring and plant leaf disease detection 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA). 226-31.

[28] Li K, Lin J, Liu J, Zhao Y. Using deep learning for image-based different degrees of ginkgo leaf disease classification. Information 2020; 11: 2-95.

[29] Sammy V. Sugarcane disease recognition using deep learning. 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE) 2019; 575-8.

[30] Srdjan S, Marko A, Andras A, Dubravko C, Darko S. Deep neural networks-based recognition of plant diseases by leaf image classification. Comput Intell Neurosci 2016; 6: 1-11.

[31] Jiang P, Chen Y, Liu B, He D, Liang C. Real-Time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access 2019; 7: 59069-80. http://dx.doi.org/10.1109/ACCESS.2019.2914929

[32] Alshawwa IA. An expert system for coconut diseases diagnosis. IEEE Access 2019; 3(4): 9-13.

[33] Alshawwa IA. High efficiency disease detection for potato leaf with convolutional neural network. sn computer science. SN Comp Sci 2021; 2(4): 1-11.

[34] Liu L, Li J, Sun Y. Research on the plant leaf disease region extraction. Proceeding to the 2019 International Conference on Vedio, Signal and Image Processing. 2019: 29-31.

[35] Mercelin F, Deisy C. Disease detection and classification in agricultural plants using convolutional neural networks — a visual understanding. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN). 2019: 1063-8.

[36] Dong PS, Sang KC. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 2017; 17(9): 1063-8.

[37] Everton TC, Bruno MB, Gabriel MK, Da Adair OS, Marco A, Willian AP. AAutomatic Recognition of soybean leaf diseases using UAV images and deep convolutional neural networks. IEEE GeosciRemote Sensing Lett 2019; 99: 1-5.

[38] Divyansh T, Mritunjay A, Abhishek S, Sushanshu P, Nitish G, Suyash B. Potato leaf disease detection using deep learning. Proceedings of the International Conference on Intelligent Computing and Control Systems. 461-6.

[39] Kumar Roy S, Member S, Misra S, Member S, Singh Raghuwanshi N, Das SK. AgriSens: IoT-based dynamic irrigation scheduling system for water management of irrigated crops. IEEE Internet Things J 2021; 8(6) http://dx.doi.org/10.1109/JIOT.2020

[40] Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 2018; 145: 311-8. http://dx.doi.org/10.1016/j.compag.2018.01.009