

Smart Farm-Care using a Deep Learning Model on Mobile Phones

Mercelin Francis ¹, Kalaiaarasi Sonai Muthu Anbananthan ^{2*}, Deisy Chelliah ¹,
Subarmaniam Kannan ², Sridevi Subbiah ¹, Jayakumar Krishnan ²

¹ Thiagarajar College of Engineering, Madurai, Tamilnadu, India.

² Faculty of Information Science and Technology, Multimedia University, Selangor, Malaysia.

Abstract

Deep learning and its models have provided exciting solutions in various image processing applications like image segmentation, classification, labeling, etc., which paved the way to apply these models in agriculture to identify diseases in agricultural plants. The most visible symptoms of the disease initially appear on the leaves. To identify diseases found in leaf images, an accurate classification system with less size and complexity is developed using smartphones. A labeled dataset consisting of 3171 apple leaf images belonging to 4 different classes of diseases, including the healthy ones, is used for classification. In this work, four variants of MobileNet models - pre-trained on the ImageNet database, are retrained to diagnose diseases. The model's variants differ based on their depth and resolution multiplier. The results show that the proposed model with 0.5 depth and 224 resolution performs well - achieving an accuracy of 99.6%. Later, the K-means algorithm is used to extract additional features, which helps improve the accuracy to 99.7% and also measures the number of pixels forming diseased spots, which helps in severity prediction.

Keywords:

Deep Learning;
Convolution Neural Network;
Feature Extraction; Segmentation;
Classification.

Article History:

Received:	10	July	2022
Revised:	29	November	2022
Accepted:	21	December	2022
Available online:	14	February	2023

1- Introduction

Smart farming is initiated to meet the difficulties arising in agricultural production concerning productivity, food safety, and sustainability. This makes agriculture more efficient and effective with the help of high-precision algorithms. Various sensors are incorporated to monitor the entire crop, resulting in a large amount of data. Apart from the varying climatic conditions and the soil types, accurate and early detection of diseases is an important factor in improving the production of any crop, particularly apples. This helps in taking appropriate preventive measures, which reduce crop loss and increase production in terms of quality and quantity [1–3].

Pathogenic agents that cause plant diseases are fungi, bacteria, and viruses. Our work concentrates on disease classification and the visualization of diseased spots on apple leaf images. Fungal diseases like apple scabs, black rot, and cedar rust appeared on the leaves are shown in Figure 1, along with healthy ones. Apple Scab is caused by the fungus *Venturia inaequalis* where grey-brown lesions appear on the surface of the leaves. Cedar apple rust is a fungal disease caused by *Gymnosporangium juniperi-virginianae* where bright orange-yellow leaf spots develop on the upper surfaces of the leaves. The *Botryosphaeria obtusa* fungus causes black rot, which appears as a yellowish-brown spot on the upper surface of the leaves that enlarges into a circle [3, 4].

Various Information and Communication Technology (ICT) initiatives like mKisan and AGRITECH are developed by researchers and agricultural experts for easy farming guidance. This is accomplished using traditional image processing, data mining, and machine learning methods. Some traditional ICT-based systems can be either mobile-based

* **CONTACT:** kalaiaarasi@mmu.edu.my

DOI: <http://dx.doi.org/10.28991/ESJ-2023-07-02-013>

© 2023 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

or web-based decision support systems. Nowadays, mobile phones are prominently used among farmers in various agricultural applications, like receiving information about the market value of the product, specific crops to be planted, the probability of disease, pesticides to be applied, etc. [1].

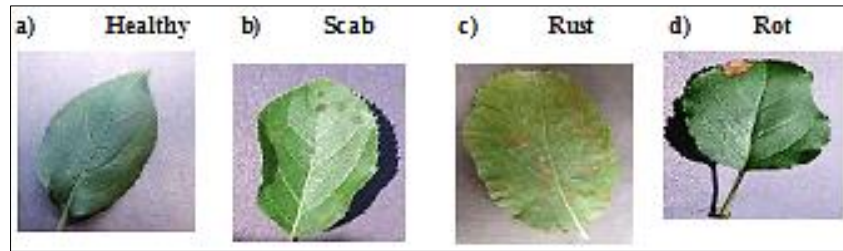


Figure 1. Input dataset categories

Classification is the main task in machine learning. The main aim is to classify objects based on the features of binary or multiclass classes. Disease diagnosis is one of its applications. Earlier, disease diagnosis and classification were performed for the hand-crafted features, like HOG, SIFT, SURF, and LBP. These descriptors did not perform well as they took a lot of time. Segmenting images from the complex background also enhances the classification process. Many traditional approaches were used for classification, such as selecting the local minimum based on the threshold cut-off from the histogram of intensities constructed [5, 6]. Additional features such as color, texture, and shape can be extracted using the Gray Level Co-occurrence Matrix (GLCM) [7]. Based on the features extracted, the K-means algorithm can be used to segment the regions, and the K-nearest neighbour (KNN) classifier can be used for prediction from the extracted features [8–10]. Later, artificial intelligence was incorporated, and an optimized MSF-AdaBoost model was developed for classifying and monitoring powdery mildew on winter wheat. A Multiplier Classifier System (MCS) is created based on different machine learning algorithms such as Support Vector Machine (SVM) and Random Forest Classifier for pattern recognition of wheat leaf diseases and groundnut diseases, respectively [11–13]. K-means clustering is also proposed for identifying three classes of wheat leaf diseases. A Bayesian network-based diagnosis model is implemented for stripe rust, providing accurate identification and prediction [14, 15].

Hence, intelligent image data analysis is an important area of research in the agricultural domain for various agricultural applications where a combination of different Machine Learning (ML) and Deep Learning (DL) algorithms are applied [16–18]. A survey on deep learning in agriculture and the efforts taken by the researchers, starting from dataset acquisition to performance evaluation metrics, helps in understanding the merits and demerits of the different models [19]. Despite various limitations, like small datasets with low resolution or noisy images, the survey concludes that the application of deep learning in agriculture outperforms the traditional image processing technique by offering better performance [20–22].

The state-of-the-art methodologies use images for classification. So, image classification is the major domain where deep learning models are used. Challenges that degrade the performance of the classification systems developed are, firstly, the similarity in the appearance of diseases in the input image dataset. Secondly, visual interferences occurred while capturing the image due to reflection, equipment jitter, too much exposure to illumination, etc. [23, 24]. A Multi-Layer Perceptron (MLP) called CNN is the base of all deep learning models. CNN models were developed to perform well with a gain in accuracy as the layers are stacked [25]. The ILSVRC challenge exploited various deep models trained and tested on the ImageNet database published in 2009, with a depth of 8 to 152 [26, 27]. The standard structure of all deep learning models is convolutional, linearly stacked layers, followed by pooling and fully connected layers. A survey reveals that increasing hidden layers is the main key to success. An increase in the number of convolutional layers increases parameters and computational complexity [25–27].

Creating a new model from scratch on a new dataset is a time-consuming and tedious process. Therefore, models developed and trained on the ImageNet dataset are retrained on the new dataset. Such models are called pre-trained models, and the process of applying the pre-trained models to the new dataset is called transfer learning. Therefore, a MobileNet model developed by Google, pre-trained on the ImageNet dataset, is retrained on the apple leaf image dataset retrieved from the PlantVillage database. The retrained model results show greater accuracy than the AlexNet, VGG, and Inception models, with smaller sizes and less computational cost.

The main novelty and contribution of the research are listed:

- A pre-trained MobileNet model is fine-tuned to get accuracy greater than that of the existing models with less size and computational cost.
- A pre-trained MobileNet model is deployed on an android mobile for disease diagnosis in agricultural crops.
- Also, disease spots are visualized, and the size of the spot is measured in inches to predict the severity of the disease using K-means segmentation.

2- Literature Review

2-1- Relevant Studies

Detecting disease in agricultural plants or fruits is a computer vision area that focuses on identifying the presence of a diseased part in an image, which leads to smart farming. Smart farming is employed to face the challenges of agricultural production concerning productivity, food safety, and sustainability. The increasing population needs to increase food production to maintain high nutritional quality without affecting nature [28–30]. Identification of diseases by segmenting the area of interest from a real-world environment are the different operations performed for different agricultural applications [31, 32–34]. A system can be developed to perform these operations using various algorithms, like computer vision and image processing.

Deep Learning (DL) is an emerging technique drawing various researchers' attention to object segmentation, detection, recognition, and classification. Deep learning models can be supervised, semi-supervised, or unsupervised. Convolutional Neural Network is the base of all supervised deep learning models using images or video as input datasets. These models consist of multiple layers, in which each layer computes the convolutional transforms followed by non-linear and pooling operations. Hence, CNN can be used either for extracting features or as an end-to-end classifier. As a feature extractor, CNN extracts features that may then be fed into an SVM classifier for classification. CNN as a classifier can be directly used for image classification [20, 30, 33]. This paper focuses on how a pre-trained model functions and can be applied to real-time applications.

The performance of existing deep learning models like LeNet [35], AlexNet [26], CaffeNet [36], VGG [25], GoogleNet [27], Inception [37], Inception-ResNet [38], and MobileNet [39, 40] were compared when applied to the ImageNet dataset. Each CNN differs from the other based on the layer count and structure. The performance of these models motivated the researchers to apply them in various applications using medical images, satellite images, and agricultural images. Medical images like Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans are processed to correctly diagnose various diseases. Satellite images are processed for different applications, namely oceanography, agriculture, warfare, etc. The application of image processing in agriculture results in improved decision-making for vegetation, irrigation, fruit sorting, etc. [24, 41, 42].

Objects Recognition and classification of the input images have various uses in real-world applications. Objects recognized may vary based on the application – for instance, face, gesture, etc. Face recognition is used as a security measure for smartphone devices, and gesture recognition is used as a touchless user interface for driving. These systems were developed using the concept of transfer learning. Hence, transfer learning is a technique used for training the model where the knowledge of the previously trained model is transferred and used for a different but related purpose [43, 44].

The paper concentrates on the performance of different models applied in agricultural applications, restricted to plant disease detection and diagnosis, which improve productivity. Detecting the leaf area and & diagnosis of diseases found in the plant is achieved via monitoring the crop by capturing the images either manually or using the sensors. Unmanned Aerial Vehicle (UAV) imagery is used for crop type classification and plant/leaf area recognition. From the captured images, relevant information in the form of features is extracted, which can be later used to identify, classify, and segment the specific area of interest, like leaves, roots, soil, weeds, seeds, etc.

Transfer learning of pre-trained models and application-specific or author-defined CNN architectures are also developed for solving the above tasks, such as plant or disease identification [30, 44]. Different models were developed, and their performances are listed in Table 1.

Table 1. Models and Dataset

Author	Architecture	Purpose	Dataset source	Accuracy
Ferentinos, (2018) [44]	AlexNet, Googlenet	Compared 2 architectures for 26 plant diseases	Hughes Salathe Dataset	99.53
Brahimi et al. (2017) [45]	AlexNet, Googlenet	Compared the 2 architectures with the SVM & Random Forest	Tomato leaves from PlantVillage dataset	99.18
Fuetes et al. (2017) [46]	Author defined CNN	9 different tomato diseases and pests	PlantVillage and other captured images	Satisfactory performance – not high
Amara et al. (2017) [47]	LeNet	Banana leaf disease classification	PlantVillage	96.00
Mohanty et al. (2016) [23]	GoogleNet, AlexNet	364 classes of images– 54306 images	PlantVillage	99.35
Sladojevic et al. (2016) [48]	CaffeNet	13 classes of images- 4483 images	Images created by the authors	96.30

Among the models developed for disease detection and classification, VGG outperformed other models. VGG attained a classification accuracy of 99.53% in classifying images to their respective classes from an image dataset comprising 17,548 images with a very low error rate but a large size or number of parameters. More profound and complex networks are developed to achieve high accuracy. More profound or deeper networks can be constructed by

inserting the convolution and pooling layers. The convolution and pooling layers help in extracting features. The count of layers ranges from 5 to 22 layers [44]. Irrespective of transfer learning, the deep residual learning method effectively predicts disease accurately within a minimum amount of time after taking tomato leaf images. This method outperforms the VGG model in a normal CPU environment without extra hardware [49].

Researchers are currently interested in deploying various AI applications on mobile devices, specifically smartphones. Hence, in this paper, the major focus is on deploying the application to mobile devices for disease diagnosis in agricultural fields. It is said that within 2 years, the applications deployed on mobile devices or smartphones will help the different agricultural experts or farmers in predicting as mobile phones will become an inevitable thing for all people. Mobile phones currently play a vital role in determining which crop can be planted in a particular area during a specific season, from sowing to cultivation [1]. However, disease diagnosis via mobile phones is not yet available to farmers. Therefore, research is focused on developing a model for disease diagnosis via mobile devices to identify 4 classes of apple leaf diseases: scab, rust, rot, and healthy leaf images.

Hence, this paper considers a model called MobileNet, which is developed for classifying objects into 1000 classes using a trained ImageNet dataset using Neural Networks (NN) with multiple layers. Learning occurs in these layers based on multiple attributes and features identified from the input dataset [39, 40]. For identifying a leaf image, multiple features are extracted at the end of NN. The first few layers of NN are used to determine properties such as color, hue, etc. Additional layers are added over this network to learn similar features. Initial layers have to distinguish and train on the most salient features. These features can be trained over the final layers to produce a specific output for identifying the type of disease present on a particular plant.

The MobileNet architecture uses only up to 4 million parameters and is designed to run efficiently on mobile devices, whereas VGG and ResNet have 130M and 25M parameters, respectively. Similarly, 300 MFLOPs of calculations are performed in the MobileNet model, which makes the model faster when compared to the other larger model, which performs up to 4 GFLOPs of calculations [25, 39, 44]. Based on accuracy, the current paper has high accuracy, similar to VGG.

The other major advantages are:

- Automatically learns the features from the training set of images;
- Can work in a real-world environment;
- It can be used on mobile;
- Fewer parameters;
- High accuracy;
- Mobile in agriculture.

2-2- Role of Image Segmentation

Different tasks, namely image segmentation, feature extraction, and target recognition, belong to the computer vision field to process any image. In order to extract useful information from certain parts of the image, the image segmentation technique is used to divide the input image into several meaningful target areas based on the features like color, texture, edge, etc. The main goal of image segmentation is to divide the image into components or regions based on the application domain. The segmented objects in the image are the target objects on focus during analysis. Hence, it is easier to detect the target object and extract its features and similar objects in the image [50].

Image segmentation is the second step that is considered to be the most crucial because diagnostic precision plays an important role in detection results. With the deepening of research, image segmentation technology also has made great progress in selecting the target object. In Guo et al. [50], images are segmented based on the results of Region Proposal Network (RPN) algorithm. The segmented leaves are given as input to the transfer learning model for further analysis. The model is then evaluated on a dataset following training. The accuracy was 83.57 percent, which is higher than the accuracy obtained using the traditional method.

Al-Tarawneh [51] converted the images taken for study to $L * a * b$ color space to enhance the analysis and classification process, given the images were collected from a real-world environment affected by uncontrolled illumination. Dataset taken for study are olive leaves. The converted images were classified using fuzzy C-means algorithm, and it gives an accuracy of 86%. From these images, the diseased pixels can be identified. To enhance the diseased spots, median filtering is applied. The severity percentage was computed by dividing the number of pixels classified as diseases by the total leaf area.

Entuni et al. [52] also computed the severity of the disease in plant leaves by combining Fuzzy C-Means and YCbCr colour space. It was concluded that YCbCr colour space has a greater detection rate when compared with RGB, HSV, $L * a * b$, as YCbCr can separate luminance from chrominance more effectively. Combining Fuzzy C-Means and YCbCr algorithm gave an accuracy of 96.81%.

The survey shows that features can be extracted from a segmented image using the GLCM method. For segmentation purpose, the K-means algorithm is used on the pre-processed image to extract diseased spots that occurred due to Bacterial Blight and Leaf spots, which led to the identification and classification of diseases in pomegranate leaves. Another alternate approach learned in the survey is based on converting to the color space model $L * a * b$ and performing classification of diseased spots using the K-means algorithm [53].

As Deep Learning (DL) also started attracting researchers with its promising results, an attempt was made to compare the features extracted via the Gray Level Covariance Matrix (GLCM) and the pre-trained model for classification. Using GLCM, 12 texture features are extracted, and using a pre-trained AlexNet model, 1000 features are extracted and gave an accuracy of 93.85%, far better than the GLCM feature extraction-based method [54].

A novel dilated dense encoder-decoder architecture with a custom dilated spatial pyramid pooling block is designed to localize the selected region accurately for segmentation. The dilation enables better spatial understanding, and dense connectivity preserves the features learned for better localization, currently applied in the medical domain. Also, a custom 2D dilated dense UNet architecture is used to localize and segment the target area in medical images [55]. These functions are similar to the DL models for classification, and features are extracted automatically for segmentation. It is currently used in medical images but can be extended its usage to the agricultural domain. The F-CNN model and S-CNN model are trained using the original full images without segmentation and segmented images. S-CNN performed better on test images not previously seen by the model than F-CNN [56].

From the survey, it is understood that using the DL model for segmentation gives promising results. But it increases the complexity and size of the model when merged along with any classification model. Therefore, a simple machine learning technique named K-means is chosen to segment the image spots. Features are fed to the pre-trained m-ADD model for further classification to improve accuracy. Also, segmented severity can be calculated from the diseased spots, and the features are fed to the m-ADD. Hence, the model is named Agricultural Disease Diagnosis and Severity Estimation using k-means.

3- Research Method, Design and Implementation

The proposed mobile-based disease diagnosis for apple leaf image disease is illustrated in Figure 2. It utilizes a pre-trained deep learning model named MobileNet to collect the features found in the different classes of diseased and healthy plants. The proposed system helps the user identify the diseases that occurred at its early stage with the aid of a smartphone in a real environment.

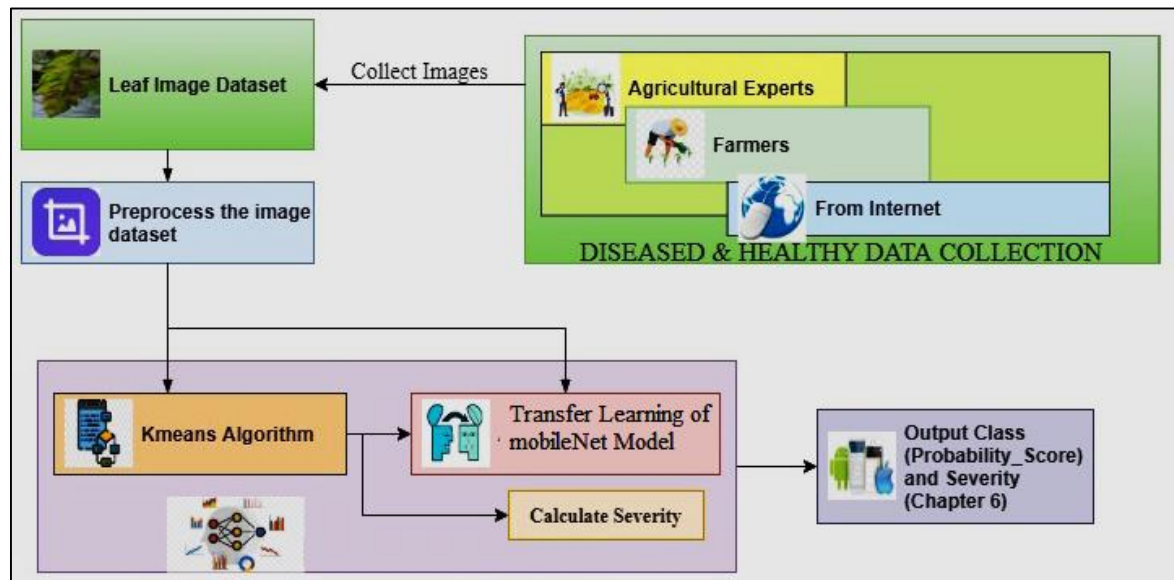


Figure 2. Proposed Model for Disease Diagnosis

The MobileNet model is deployed on an android device, and the phone is held parallel to the leaf such that the entire leaf is visible inside the frame. A Tensorflow platform is used to deploy a smartphone-based disease detection model to identify 4 classes of disease found in leaves: leaf scab, cedar rust, black rot, and healthy.

Transfer learning of a MobileNet model pre-trained on ImageNet dataset is utilized in the apple leaf image dataset consisting of 3171 images retrieved from PlantVillage. Based on the training results of 6000 iterations on NVIDIA Tesla GPU, the highest accuracy is achieved with less size and computational cost. Therefore, the same model is selected to test on a mobile device in the field. The K-means method is applied to images to segment the diseased spots and predict the severity of the disease. The workflow design of the experiment is shown in Figure 3.

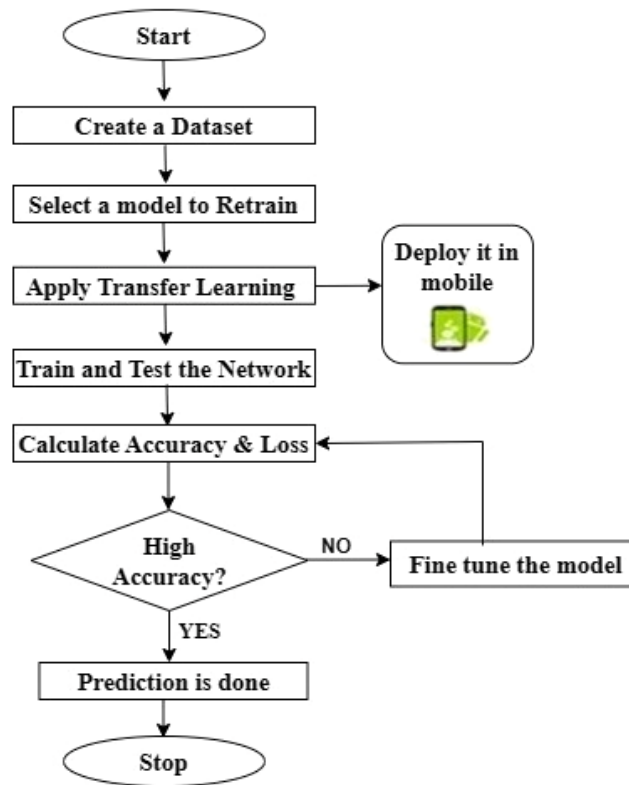


Figure 3. Workflow of the Proposed System

Initially, the model is trained, validated, and tested on the desktop to calculate the size, accuracy, and other performance metrics. Later, a mobile app is created to deploy the model on the mobile platform.

3-1- Creating Dataset and Pre-processing

The apple leaf image dataset of JPEG images is retrieved from the PlantVillage website. The data set consists of 4 classes comprising 1645 healthy images, 630 leaf scab images, 621 black rot leaf images, and 275 cedar rust leaf images. Initially, all the images on the disk are analyzed after resizing the image to 224×224 and splitting the dataset into training and testing sets.

Every image is reused multiple times during training; therefore, bottleneck values are calculated for each image. The bottleneck is an informal term used as an image representation, as each image takes a lot of time to process. So, these bottleneck values are cached and can be used for classification as there is no major change.

3-2- Architecture of the Proposed Model

MobileNet is an efficient class of CNN designed by researchers at Google called MobileNet is said to be the first mobile-based architecture which can be run within the minimum time on the mobile phone. The main difference between the traditional CNN and the MobileNet architecture is that depthwise and pointwise convolution layers are used in the latter model. In contrast, convolution layers are used in the former. The usage of depthwise and pointwise convolution reduces the computational cost and time.

Figure 4 and Table 2 show the layers of the MobileNet architecture. Column 1 of Table 2 shows the type of convolution and the stride performed - with the number of filters and its kernel size in the second column. Each layer's input image width and height are listed in column 3 of Table 2. Column 4 and 5 in Table 1 lists the number of parameters and computations performed using the defined formulas. To compute the number of parameters in each layer is given:

$$\text{Number of parameters} = (n * m * k) * L \quad (1)$$

where n is the width of the filter, m is the height of the filter, k is the number of input feature maps, and L is the number of output feature maps. The number of computations performed in each layer varies based on the layers. The number of computations is computed by counting each layer's Multiply-Accumulates (MACCs). The model's speed can be predicted based on the number of computations. Less computation faster the model.

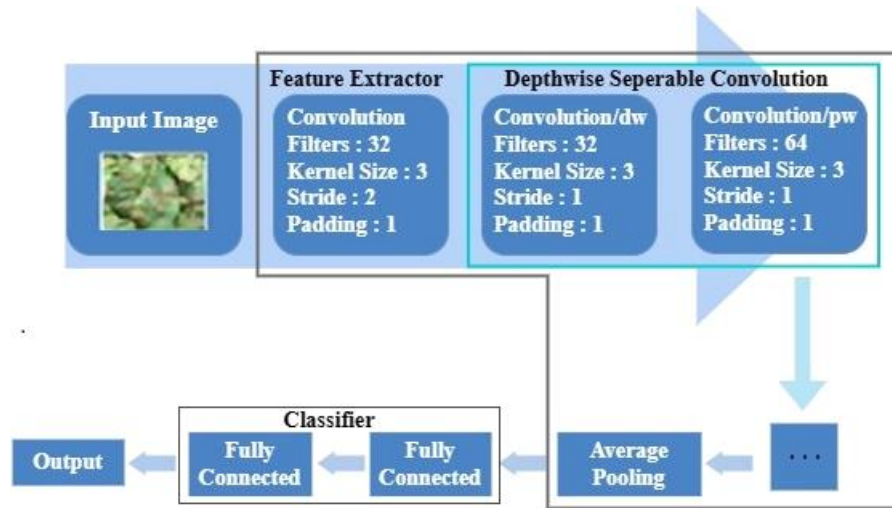


Figure 4. MobileNet Layers

Table 2. Number of Parameters and Computational Cost of MobileNet Architecture – Layerwise

Block	Type/Stride	Filter Shape x number of filters	Input Size	No. of Parameters (Size)	Computational Cost
0	Conv1/s2	3×3×32	224×224×3	3×3×3×32 = 864	3×3×3×32×112×112 = 10M
1	Convdw/s1	3×3×32 dw	112×112×32	864	3×3×32×112×112 = 3M
	Conv/s1	1×1×32×64	112×112×32	2048	32×112×112×64 = 25M
2	Convdw/s2	3×3×64 dw	112×112×64	576	3×3×64×56×56
	Conv/s1	1×1×64×128	56×56×64	8192	64×128×56×56
3	Convdw/s1	3×3×128 dw	56×56×128	1152	3×3×64×56×56
	Conv/s1	1×1×128×128	56×56×128	16384	128×128×56×56
4	Convdw/s2	3×3×128 dw	56×56×128	1152	3×3×128×28×28
	Conv/s1	1×1×128×256	28×28×128	32768	128×256×28×28
5	Convdw/s1	3×3×256 dw	28×28×256	2304	3×3×128×28×28
	Conv/s1	1×1×256×256	28×28×256	65536	256×256×28×28
6	Convdw/s2	3×3×256 dw	28×28×256	2304	3×3×256×14×14
	Conv/s1	1×1×256×512	14×14×256	131072	256×512×14×14
7-11	5 × convdw/s1 Conv/s1	3×3×512 dw 1×1×512×512	14×14×512 14×14×512	5 × (4608 + 262144) = 1333760	5 × [(3×3×256×14×14) + (512×512×14×14)]
12	Convdw/s2	3×3×512 dw	14×14×512	4608	3×3×512×14×14
	Conv/s1	1×1×512×1024	14×14×512	524288	512×1024×7×7
13	Convdw/s2	3×3×1024 dw	7×7×512	9216	3×3×512×7×7
	Conv/s1	1×1×1024×1024	7×7×1024	1048576	1024×1024×7×7
14	AvgPool/s1	Pool 7×7	7×7×1024	-	0
15	FC/s1	1024×1000	7×7×1024	1024000	7×7×1024
16	Softmax/s1	Classifier	1×1×1000		

3-2-1- Standard Convolution Layer

In this layer, convolution is applied to all the input channels. The layer receives an RGB image of size $W \times H \times D$ as input representing the width, height and depth of the image to produce low-level features represented using a feature map. The low-level features can be edges or curves. There are mainly 4 parameters for a standard convolution; the number of filters (k), filter size ($K \times K$) with Stride (S) and Padding (P). Algorithm I (Figure 5) shows the operation performed during convolution.

Algorithm I. Convolution Layer Operation

Inputs: A receptive field of size $(n \times n)$ from an RGB image I of size $(W \times H \times C)$, where $W \times H \times C$ maps the width, height and number of input channels of N images. The number of filters (k) of size $(K \times K)$ is strided over the input image to obtain k feature maps.

Output: k feature maps contain low-level features

// $P = 0$ - for no zero padding; and $P=1$ for zero padding

```

 $P = \frac{k-1}{2}$  //to preserve the dimensionality of the feature map
For (int n=0; n < N ; ++n){
    For (int d=0; d < D; ++d) {
        For ( int w=0;w < W; ++w) {
            Float sum = 0;
            For ( int c=0; c<C; ++c) {
                For ( int kx = 0; kx < K ; ++kx ) {
                    For ( int ky=0 ; ky < K ; ++ky ) {
                        sum = sum + Dot_product( kernel ( kx , ky , c , d ), input (w+kx, h+ky , c ,
n));
                    }
                }
            }
            Output_feature_map [w , h , c , n ] = sum
        }
    }
}
Update feature_map[ ] = max ( 0, feature_map[k]) // Perform ReLU for non-linearity

```

Figure 5. Algorithm Convolution Layer Operation

The input for mobilenet is a $224 \times 224 \times 3$ color image that passes through the first standard convolution layer with 32 feature maps or filters having a size 5×5 and a stride of 2. The image dimensions change from $224 \times 224 \times 3$ to $112 \times 112 \times 32$ resulting in the number of parameters =864 and the number of computations = 10M, as shown in Table 2. For a convolutional layer with kernel size K , the number of MACs is:

$$\text{Number of MACs} = K * K * C_{in} * H_{out} * W_{out} * C_{out} \quad (2)$$

where $H_{out} * W_{out}$ represents the output feature map, $K * K$ kernel containing input weights, C_{in} - the input channels, C_{out} - the number of output channels or the kernels.

3-2-2- Depthwise Separable Convolution

Algorithm II (Figure 6) explains the operation of the depthwise separable convolution layer. Different steps followed in depthwise separable convolution are the following:

Algorithm II. Depthwise Separable Convolution Layer Operation

Inputs: A receptive field (R) of size $(n \times n)$ from an RGB image I of size $(W \times H \times D)$, number of filters (k) , filter size $(m \times m)$ with Stride (S) and Padding (P)

Output: k feature maps containing low level features

Repeat for $I [1 : D]$

$P = 0$ // for zero padding

for any_receptive_field {

initialize sum=0

for any_kernel_size {

sum = sum + dot_product (R, K)

}

Return feature_map [sum]

$R[] = R [+S : n ; 1 : n]$ //Perform Striding S for row wise movement

$R[] = R [1 : n ; +S : n]$ // perform striding S for column wise movement

Update feature_map[] = max (0, feature_map[k]) // Perform ReLU for non-linearity

Figure 6. Algorithm Depthwise Separable Convolution Layer Operation

- **Depthwise Convolution:** We give the input image a convolution without changing the depth in depthwise convolution.

$$\text{Number of MACs} = K * K * C * H_{out} * W_{out} \quad (3)$$

where $K * K$ depthwise convolution applied on a $H_{out} * W_{out}$ feature map with C number of input channels

- **Pointwise Convolution:** The pointwise convolution is so named because it uses a 1x1 kernel or a kernel that iterates through every single point. This kernel has a depth of however many channels the input image has

$$\text{The number of MACs computed} = C_{in} * H_{out} * W_{out} * C_{out} \quad (4)$$

where $C_{in} * H_{out} * W_{out}$ is the output feature map from the depthwise convolution and is projected to C_{out} dimensions.

3-2-3- Activation Layer

An activation layer utilizes the non-linear operators, such as Tanh, Sigmoid and Rectified Linear Unit, in the neural network model, after each convolution layer. This helps to build a powerful network by transforming the weighted sum of input from one layer to an output.

3-2-4- Pooling Layer

To reduce the spatial dimension of the input, retaining the unique features extracted pooling layer is used. A kernel or filter is used, which takes the maximum or average value within the filters applied to the image representation. Such a process is called max pooling or average pooling, respectively. Figure 7(i) shows the activation and pooling operation process. For example, consider a matrix representing a feature obtained after activation, as shown in the LHS equation in the Figure 7(i). The result obtained after max pooling and average pooling is shown in the RHS of the equation shown in Figure 7(i).

(a) Max Pooling	(b) Average Pooling
$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1.5 & 0 \\ 0 & 0 \end{bmatrix}$

ii) Fully Connected Layer

[Scab Rot Rust healthy]
[1.0 0.4 0.2 0.3]

Figure 7. Pooling & Fully Connected Layer

3-2-5- Fully Connected Layer

It generates a vector of feature values taking the feature map matrices as input, obtained as output from various convolution, separable convolution and pooling layers after performing activation after each convolution and separable convolution layers. Here in the proposed system, the second fully connected layer takes the input features to map into 4 categories of classes. The labels and the output vector obtained in the proposed system are shown in Figure 7-ii. Figure 7-ii shows that the classifier has predicted the output class and its probability of the specific input image as 100% scab, 40% rot, 20% rust and 30% healthy.

3-2-6- Softmax Function

This function determines which disease is occurred finally. The output of the fully connected layer is fed to the softmax layer with no activation. Hence the softmax layer of MobileNet model is modified for the proposed classification system, which contains 4 classes of images to be classified. Hence the output of the softmax layer will be of size 4, providing a 100% confidence in total. Table 3 shows an example output of the softmax layer.

Table 3. Softmax Layer Output

Class	Prediction Probability (%)
Scab	99.1
Rot	0.4
Rust	0.2
Healthy	0.3
Total Confidence	100

3-3- Transfer Learning

Instead of creating a model from scratch, existing deep learning networks trained and tested on the different datasets can be used to solve many classification-related problems. This process is called transfer learning. The entire transfer learning process can be performed in the following steps:

- Select a pre-trained model;
- Prepare the dataset;
- Fine-tune the model.

Fine-tuning a network can be done in 3 ways, viz.

- Train the entire model: The model is created and trained from scratch. This is a tedious and time-consuming process.
- Train some layers and leave the others frozen: It freezes some of the convolution layers and trains the other layers that are not freeze.
- Freeze the convolutional base: it acts as a feature extractor where all the convolution layers are freeze. The classifier, which contains the fully connected and the softmax layer, is opened to modify with respect to the application.

4- Results and Discussion

The proposed system consists of 2 phases: training and testing, built-in Ubuntu 16.04 Operating system, tested and ran on NVIDIA Tesla, containing 2 GPUs. Both phases perform some steps in common. Following are the different image processing steps performed on the digital image:

- Image acquisition;
- Image pre-processing;
- Segmentation;
- Feature extraction;
- Classification.

4-1- Image Acquisition for Dataset Creation

Experimentation is conducted on leaf images of paddy and tomato obtained from the agricultural field, agricultural colleges database, and the web [57]. Since the count was significantly less, the accuracy of predicting the disease didn't reach the maximum. But it worked well for segmenting the diseased spots and estimating the severity. Therefore, a dataset with a sufficient number of images was considered to prove km-ADDS as the best system for intelligent disease diagnosis and severity estimation. Therefore freely available leaf images of apple, banana, tomato and cucumber were taken from the PlantVillage dataset [23]. In addition to the above images, some additional images collected from the agricultural field are also added.

In addition to the leaf images collected from the web, many more images are collected from a real-world environment using a mobile camera or other camera sensors. The collected images are taken as input images in RGB (Red, Green, Blue) format. Some sample images of tomato and paddy are shown below in Figures 8 and 9, respectively.

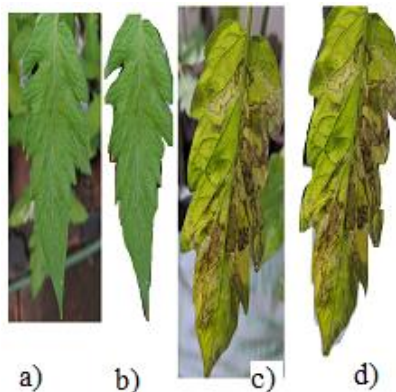


Figure 8. Sample tomato leaf images (a) Healthy-original, (b) Healthy-segmented, (c) Diseased- original and (d) Diseased- segmented



Figure 9. Sample images captured (a) Healthy, (b) Bacterial blight and (c) Leaf blast

4-1-1- Image Pre-Processing

After collecting the images from various sources, they are enhanced and augmented for further analysis. The images may be of different sizes when captured. Therefore to accelerate the image processing - all images are resized to the same size, i.e., 224×224 , from the center of the captured image. In order to distinguish between the diseased and non-diseased parts, the contrast stretching technique is used to enhance the contrast. Several color conversion techniques are also adopted to identify the leaf region and the diseased spots, as shown in Figure 10.

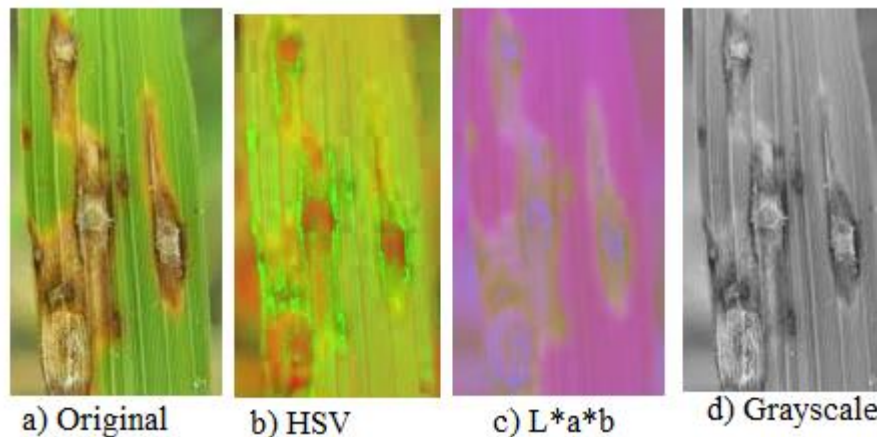


Figure 10. Sample images after color conversion

4-1-2- Image Segmentation & Severity Estimation

K-means clustering is chosen to separate the infected and healthy regions of the pre-processed image based on k clusters among the different segmentation techniques. It can detect the segment of interest of the image and find the similarity groups in data by minimizing the sum of the squares of the distance between the corresponding cluster and the object [28]. Therefore, the K-means clustering algorithm is applied to extract the diseased spots after enhancement. The segmented images were divided into different classes: healthy, leaf blast, and bacterial blight. The Image processing algorithm was developed in Python3.7, using the OpenCV library.

The proposed method uses K-means clustering to separate the diseased lesion, from which the affected ratio is calculated to predict the severity. K-means algorithm performed on km-ADDS is as follows [52, 58-61]:

- Step1: Take the pre-processed image as input;
- Step2: The image is converted from RGB to L^*a^*b color space, consisting of 2 chromaticity layers in a^* and b^* channels and a luminosity layer in the L^* channel;
- Step 3: Colors are classified using k-means clustering in a^*b^* space, and simultaneously the difference between the 2 colors is evaluated with the help of the Euclidean distance metric;
- Step 4: Each pixel of the image is labeled with its assigned cluster index;
- Step 5: The pixels present in the input image is separated by color using pixel labels which produce different image segment.

$K=3$ is used in the proposed model. Figure 11 shows the segmentation results and the different steps involved in segmentation of the diseased spot from the cropped image using K-means via applying the algorithm. The image is

segmented to 3 clusters. Figure 12 shows the features extracted from the segmented clusters containing the region of interest for the Rice Blast infected paddy leaf image. Disease severity scale for evaluation of Bacterial Leaf Blight is shown in Table 4.

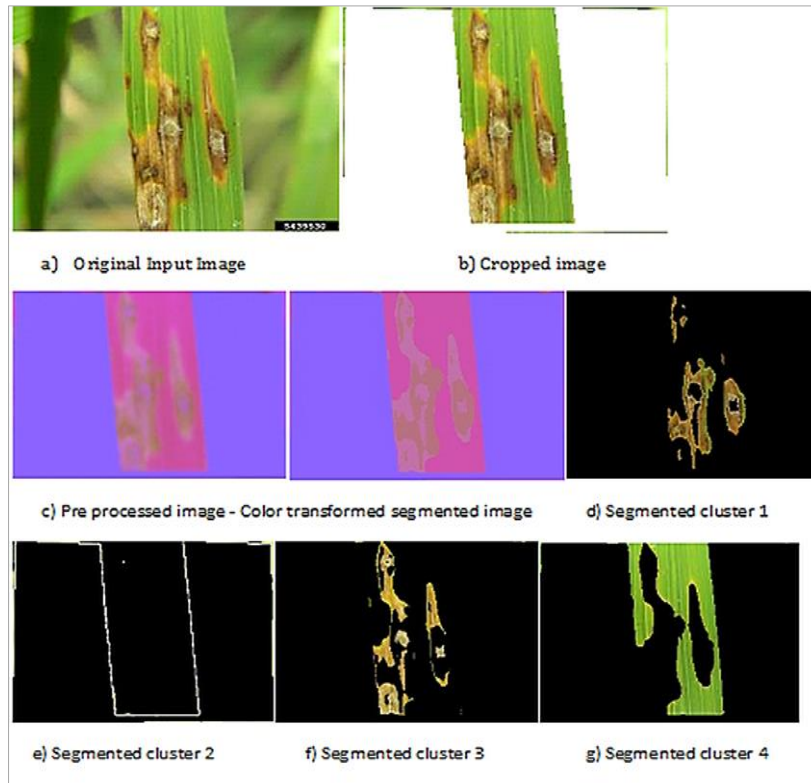


Figure 11. Sample output images after segmentation using k-means

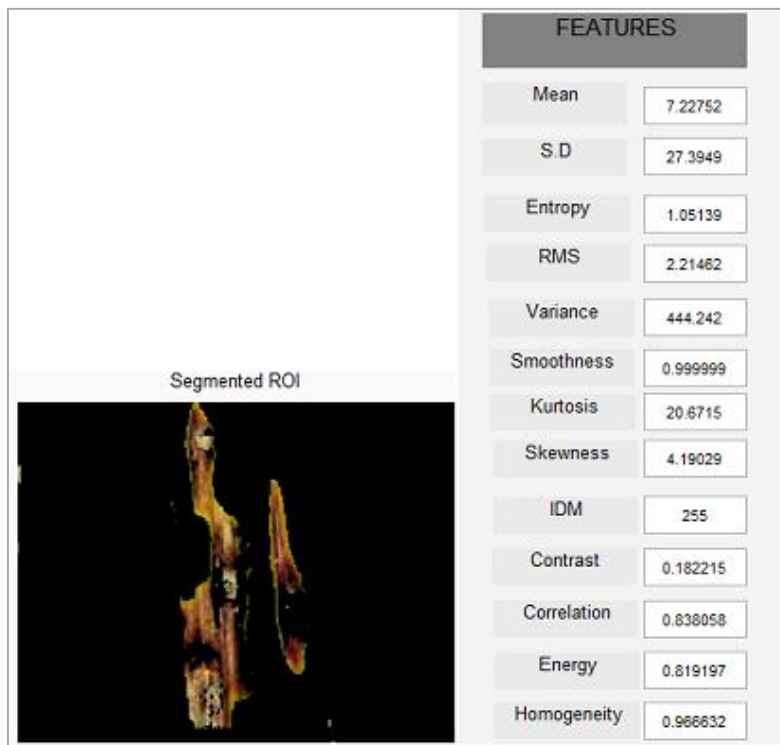


Figure 12. Features extracted from the selected segmented clusters

Table 4. Disease severity scale for Bacterial Leaf Blight (BLB)

Disease Rating	0	1	3	5	7	9
Lesion size (% of Leaf length)	0	>1-10%	> 11-30%	> 31-50%	> 51-75%	>76-100%

4-1-3- m-ADD Model for Feature Extraction and Classification

Among the various pre-trained models MobileNet model is chosen to perform transfer learning on the input dataset. This model automatically extracts the original image's features before segmentation and extracts features after performing the segmentation. With these extracted features, it performs classification. It was found that the classification accuracy increased from 99.6 to 99.7%. Hence segmented features give an additional attribute to perform classification with minimum parameters. Hence this ensembling of machine learning techniques together with a deep learning model provides the best result for deploying it in a smartphone for disease identification and classification.

4-2- Performance Evaluation

Accuracy can be measured as the number of correctly classified images among all the classes of images in the test category. Cross entropy measures loss. Both accuracy and cross-entropy are visualized in a tool named tensorboard as shown in Figure 13 using the proposed model. In Figure 13, the train and validation accuracy is plotted by taking the number of steps on the x axis. It is observed that the training and validation accuracy reached 100% and 98%, respectively, taking an average time of 6m 35secs at 6000 steps after applying a smoothing factor of 0.99.

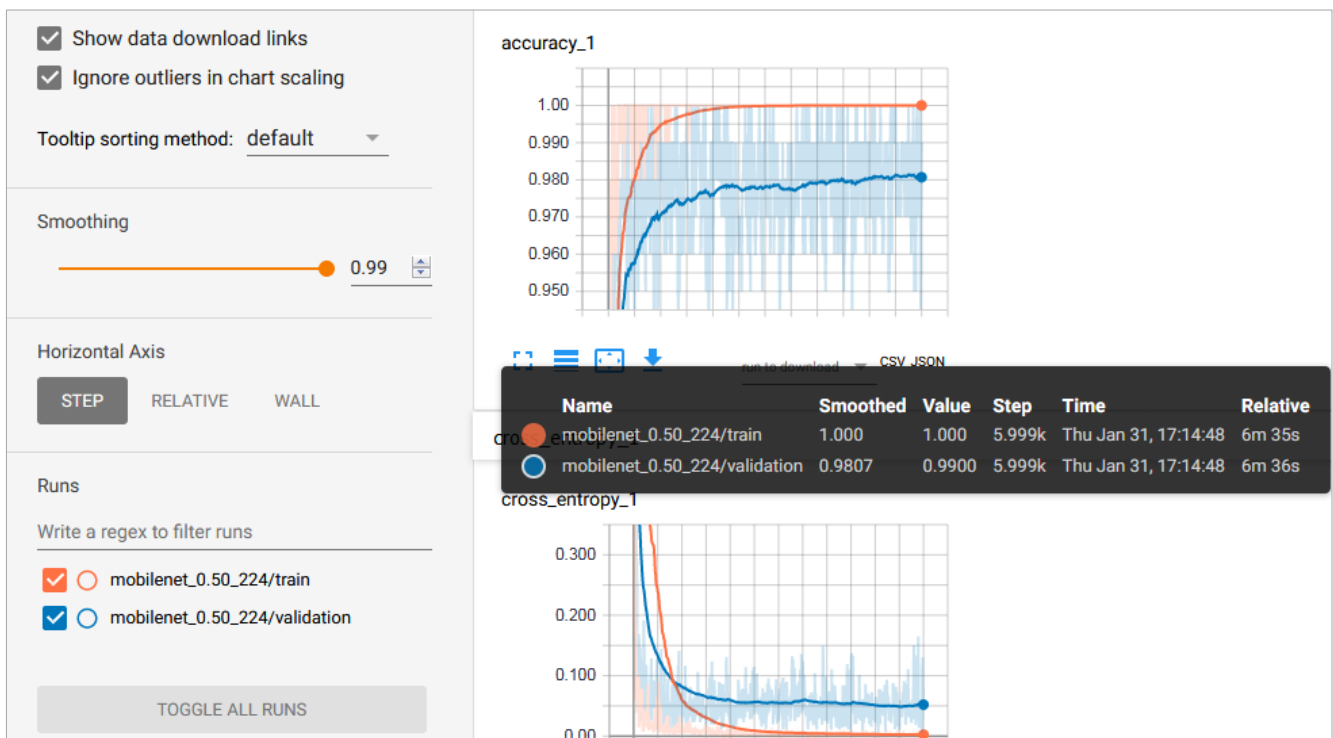


Figure 13. Accuracy and loss in tensorboard

Different MobileNet models are retrained on the apple leaf image dataset to compare their performance on mobile devices. Table 5 shows the accuracy obtained while executing the proposed model, i.e., transfer learning of the MobileNet models on the leaf image dataset. Hence, an efficient accuracy of 99.6 is obtained when the MobileNet model with 0.5 widths and 224 resolution is fine-tuned with the parameters viz. number of steps (6000), Learning rate (0.01), and train-test ratio (90:10). Learning rate and training steps are considered hyperparameters for fine-tuning.

Table 5. Performance of the proposed model

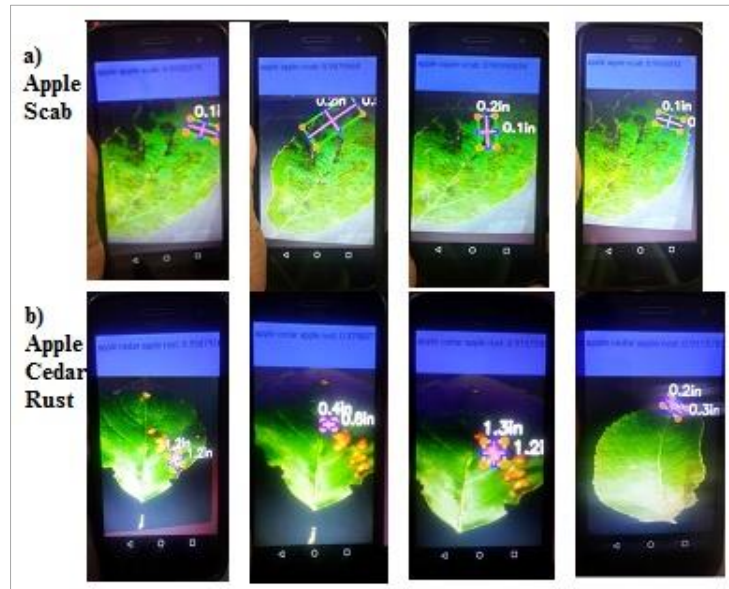
Number of Steps	Learning Rate	Train-Test Ratio	mADD0.75_224 Accuracy	mADD0.50_224 Accuracy
6000	0.01	90:10	99.4	99.6
4000	0.01	90:10	99.2	99.4
6000	0.001	80:20	98.5	99.15
4000	0.001	80:20	98.5	99.15

Table 6 compares the proposed model using K-means with the proposed work without K-means, Inception_v3 and VGG, on the apple leaf image dataset based on accuracy, a number of parameters, and Multiply and Accumulate (MAC) operation. The study summarizes that the proposed model gives better accuracy with less size and computational cost when compared to the other existing models.

Table 6. Comparison of the proposed model with existing models

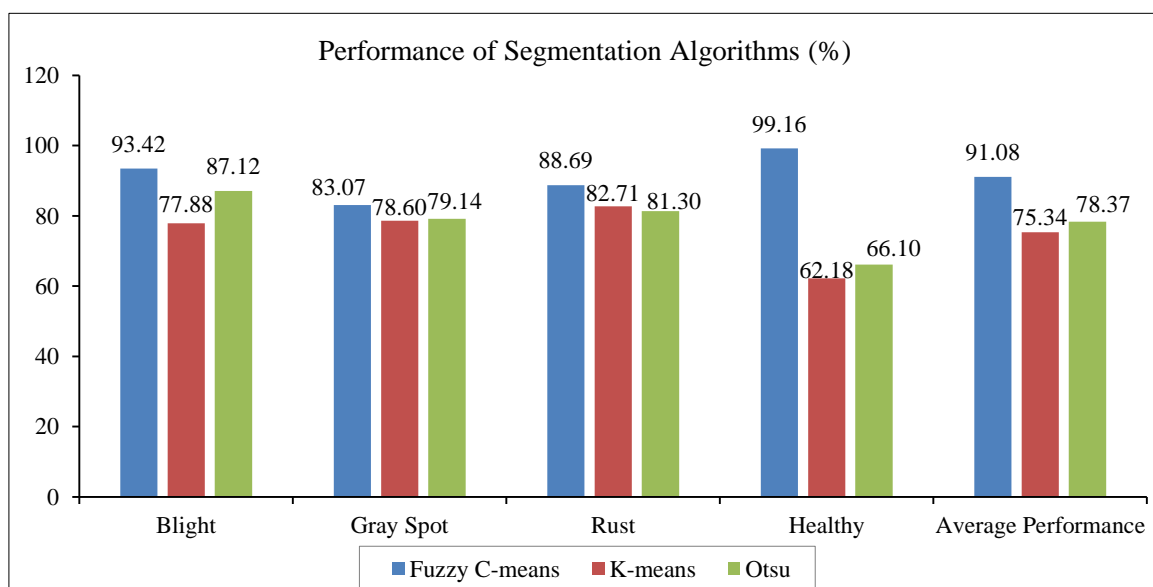
Model	Accuracy (%)	No of Parameters	MACs
Proposed model without k-means	99.6	4.2M	569M
Proposed model with K-means	99.7	4.2M	569M
MobileNetV2	97	3.47M	300M
InceptionV3	99.4	6.8M	1430M
VGG	99.5	138M	15500M

The proposed model with K-means is deployed on Android smartphones as an app and is tested in the real world environment giving a high accuracy, as shown in Figure 14.

**Figure 14.** km-ADDS deployed on Android smartphone

4-2-1- Image Segmentation

Among the various image segmentation algorithms, fuzzy C-means, K-means, and Otsu thresholding are the different methods used for analysis [62, 63]. Fuzzy C-means have been found to perform well. But due to its complexity and feasibility in real-world scenarios, it is not being used. Instead, the K-means algorithm is implemented on the input images to segment the diseased spot. Figure 15 shows the performance of k-means and the other algorithms on different images, both diseased and non-diseased.

**Figure 15.** Performance of segmentation algorithms

Similarly, it is tested on different color space models like HSV, L*a*b, RGB, and YCbCr. From Figure 16, it is evident that to get better results from K-means, the L*a*b color space model is used. Therefore, in the pre-processing stage, the input image is converted to L*a*b color space, as shown in Figures 10 and 11.

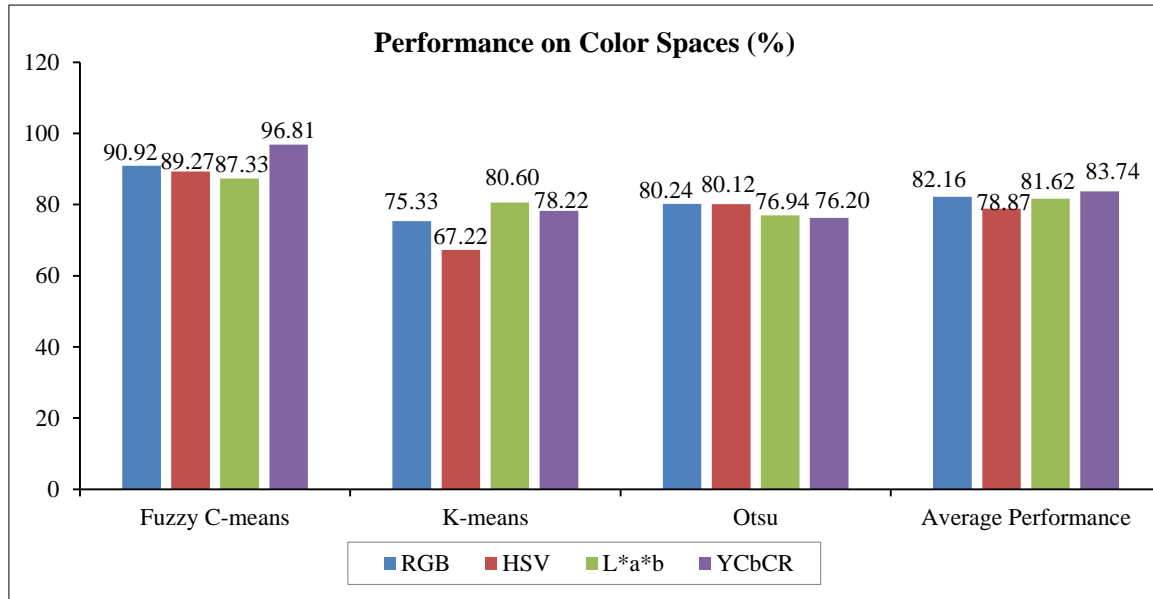


Figure 16. Performance-based on color spaces

5- Conclusion

This paper outlines the use of different deep learning models and their role in agriculture by automatically generating the features without human intervention. This also presents a novel insight into the importance of segmentation of diseased spots, transfer learning, and fine-tuning the model to develop a system for prediction. The proposed model without K-means is developed initially trained on the apple leaf dataset containing healthy and diseased leaf images. It has been done for MobileNet models that vary based on their depth and resolution multiplier to achieve maximum accuracy with less memory size, time, and complexity. The experimental results obtained after retraining the model conclude that there is a relationship between the accuracy and size of the model. The model mobilenet_0.5_128 is the most suitable model considering the tradeoff between size and complexity, but the accuracy obtained is very low. But the model proposed with K-means gives a better accuracy of 99.7% with minimum size. Therefore, for android-based disease diagnosis in apple leaf images or any other agricultural plants, mobilenet_0.5_224 is the best one, providing better accuracy with minimum size, time, and complexity.

The challenges faced were related to the dataset collection for real-time testing. Therefore, various images were captured from a real-world environment belonging to the tomato and paddy categories. The K-means algorithm is applied to paddy leaf images to extract the features, which will be given as an additional input to the proposed model developed to perform the prediction. The model is run and tested on an apple leaf image dataset; therefore, the K-means algorithm is applied on the dataset to extract the features, and fed as an input to the m-ADD model.

In future work identifying the type of disease, this model can be generalized to all plants. Also, various other metrics can be included in evaluating the system. The system can also be extended to predict severity by calculating the percentage of affected regions and suggesting necessary remedial measures.

6- Declarations

6-1- Author Contributions

Conceptualization, M.F., and D.C.; methodology, M.F and K.S.M.A.; software, S.K, and S.S.; validation, J.K and S.K., and S.S.; formal analysis, M.F.; investigation, K.S.M.A, and D.C.; writing—original draft preparation, M.F., and S.S; writing—review and editing, K.S.M.A.; visualization, S.K, and J.K.; supervision, D.C., and K.S.M.A.; project administration, D.C.; funding acquisition, K.S.M.A. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available on “<https://github.com/spMohanty/PlantVillage-Dataset>” and is freely downloadable.

6-3- Funding

This study is supported by the Faculty of Science and Technology (FIST), Multimedia University, Malaysia.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

- [1] Saravanan, R. (2012). ICTs for agricultural extension in India: policy implications for developing countries. Proceedings of 8th Asian Conference for Information Technology in Agriculture, AFITA, 3-6 September, Taipei, Taiwan.
- [2] Anbananthen, K. S. M., Subbiah, S., Chelliah, D., Sivakumar, P., Somasundaram, V., Velshankar, K. H., & Khan, M. K. A. A. (2021). An intelligent decision support system for crop yield prediction using hybrid machine learning algorithms. *F1000Research*, 10, 1143. doi:10.12688/f1000research.73009.1.
- [3] Goodridge, W., Bernard, M., Jordan, R., & Rampersad, R. (2017). Intelligent diagnosis of diseases in plants using a hybrid Multi-Criteria decision making technique. *Computers and Electronics in Agriculture*, 133, 80–87. doi:10.1016/j.compag.2016.12.003.
- [4] Kolhe, S., Kamal, R., Saini, H. S., & Gupta, G. K. (2012). Expert System for Disease Diagnosis in Soybean-ESDDS. Special issue – “AI in Agriculture”. *Journal of the Indian Society of Agricultural Statistics*, 67(1), 79-88.
- [5] Camargo, A., & Smith, J. S. (2009). An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems Engineering*, 102(1), 9–21. doi:10.1016/j.biosystemseng.2008.09.030.
- [6] Camargo, A., & Smith, J. S. (2009). Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture*, 66(2), 121–125. doi:10.1016/j.compag.2009.01.003.
- [7] Yao, Q., Guan, Z., Zhou, Y., Tang, J., Hu, Y., & Yang, B. (2009). Application of Support Vector Machine for Detecting Rice Diseases Using Shape and Color Texture Features. 2009 International Conference on Engineering Computation. doi:10.1109/icec.2009.73.
- [8] Singh, J., & Kaur, H. (2019). Plant Disease Detection Based on Region-Based Segmentation and KNN Classifier. *Lecture Notes in Computational Vision and Biomechanics*, 1667–1675. doi:10.1007/978-3-030-00665-5_154.
- [9] Phadikar, S., Sil, J., & Das, A. K. (2013). Rice diseases classification using feature selection and rule generation techniques. *Computers and Electronics in Agriculture*, 90, 76–85. doi:10.1016/j.compag.2012.11.001.
- [10] Munisami, T., Ramsum, M., Kishnah, S., & Pudaruth, S. (2015). Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers. *Procedia Computer Science*, 58, 740–747. doi:10.1016/j.procs.2015.08.095.
- [11] Chaudhary, A., Kolhe, S., & Kamal, R. (2016). An improved random forest classifier for multi-class classification. *Information Processing in Agriculture*, 3(4), 215–222. doi:10.1016/j.inpa.2016.08.002.
- [12] Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, 137, 52–58. doi:10.1016/j.compag.2017.03.016.
- [13] Islam, M., Anh Dinh, Wahid, K., & Bhowmik, P. (2017). Detection of potato diseases using image segmentation and multiclass support vector machine. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, Canada. doi:10.1109/ccece.2017.7946594.
- [14] Farid, D. Md., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4), 1937–1946. doi:10.1016/j.eswa.2013.08.089.
- [15] McCallum, A., & Nigam, K. (1998). A comparison of event models for naïve bayes text classification. *AAAI-98 workshop on learning for text categorization*, 752(1), 41-48, 26-31 July, 1998, Madison, United States.
- [16] Tan, J. W., Chang, S. W., Abdul-Kareem, S., Yap, H. J., & Yong, K. T. (2020). Deep Learning for Plant Species Classification Using Leaf Vein Morphometric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1), 82–90. doi:10.1109/TCBB.2018.2848653.
- [17] Zenkl, R., Timofte, R., Kirchgessner, N., Roth, L., Hund, A., Van Gool, L., Walter, A., & Aasen, H. (2022). Outdoor Plant Segmentation With Deep Learning for High-Throughput Field Phenotyping on a Diverse Wheat Dataset. *Frontiers in Plant Science*, 12(774068). doi:10.3389/fpls.2021.774068.

- [18] D.Pujari, J., Yakkundimath, R., & Byadgi, A. S. (2016). SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique. *International Journal of Interactive Multimedia and Artificial Intelligence*, 3(7), 6. doi:10.9781/ijimai.2016.371.
- [19] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. doi:10.1016/j.compag.2018.02.016.
- [20] Teke, M., Deveci, H. S., Haliloglu, O., Gurbuz, S. Z., & Sakarya, U. (2013). A short survey of hyperspectral remote sensing applications in agriculture. 2013 6th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey. doi:10.1109/rast.2013.6581194.
- [21] Saxena, L., & Armstrong, L. (2014). A survey of image processing techniques for agriculture. *Proceedings of Asian Federation for Information Technology in Agriculture*. 29 September- 2 October, 2014, Perth, Australia.
- [22] Chen, Y., Wang, S., & Ge, Y. (2022). A Survey on the Applications of Image Classification Based on Convolution Neural Network. 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China. doi:10.1109/ipec54454.2022.9777354.
- [23] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7. doi:10.3389/fpls.2016.01419.
- [24] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11–26. doi:10.1016/j.neucom.2016.12.038.
- [25] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. doi:10.48550/arXiv.1409.1556.
- [26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. doi:10.1145/3065386.
- [27] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298594.
- [28] Food and Agriculture Organization of the United Nations (FAO). (2009). The state of food and agriculture. Food and Agriculture Organization of the United Nations (FAO), Rome, Italy.
- [29] Tyagi, A. C. (2016). Towards a Second Green Revolution. *Irrigation and Drainage*, 65(4), 388–389. doi:10.1002/ird.2076.
- [30] Altalak, M., Uddin, M. A., Alajmi, A., & Rizg, A. (2022). Smart Agriculture Applications Using Deep Learning Technologies: A Survey. *Applied Sciences (Switzerland)*, 12(12), 5919. doi:10.3390/app12125919.
- [31] Deisy, C., & Francis, M. (2018). Image Segmentation for Feature Extraction. *Advances in Multimedia and Interactive Technologies*, 232–257, IGI Global, Hershey, United States. doi:10.4018/978-1-5225-5775-3.ch013.
- [32] Ahmad, A., Saraswat, D., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agricultural Technology*, 3, 100083. doi:10.1016/j.atech.2022.100083.
- [33] Bi, C., Wang, J., Duan, Y., Fu, B., Kang, J. R., & Shi, Y. (2022). MobileNet Based Apple Leaf Diseases Identification. *Mobile Networks and Applications*, 27(1), 172–180. doi:10.1007/s11036-020-01640-1.
- [34] Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2022). Deep learning techniques to classify agricultural crops through UAV imagery: a review. *Neural Computing and Applications*, 34(12), 9511–9536. doi:10.1007/s00521-022-07104-9.
- [35] Meshram, V., Patil, K., Meshram, V., Hanchate, D., & Ramkteke, S. D. (2021). Machine learning in agriculture domain: A state-of-art survey. *Artificial Intelligence in the Life Sciences*, 1, 100010. doi:10.1016/j.ailsci.2021.100010.
- [36] Chauhan, P., Mandoria, H. L., Negi, A., & Rajput, R. S. (2021). Plant Diseases Concept in Smart Agriculture Using Deep Learning. *Smart Agricultural Services Using Deep Learning, Big Data, and IoT*, 139–153, IGI Global, Hershey, United States. doi:10.4018/978-1-7998-5003-8.ch008.
- [37] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.308.
- [38] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-first AAAI conference on artificial intelligence*, 4-9 February, San Francisco, United States.
- [39] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint*. doi:10.48550/arXiv.1704.04861.
- [40] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2018.00474.

- [41] Wang, J., Bretz, M., Dewan, M. A. A., & Delavar, M. A. (2022). Machine learning in modelling land-use and land cover-change (LULCC): Current status, challenges and prospects. *Science of the Total Environment*, 153559. doi:10.1016/j.scitotenv.2022.153559.
- [42] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27–48. doi:10.1016/j.neucom.2015.09.116.
- [43] Kaya, Y., & Gürsoy, E. (2023). A MobileNet-based CNN model with a novel fine-tuning mechanism for COVID-19 infection detection. *Soft Computing*, 1-15. doi:10.1007/s00500-022-07798-y.
- [44] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318. doi:10.1016/j.compag.2018.01.009.
- [45] Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence*, 31(4), 299–315. doi:10.1080/08839514.2017.1315516.
- [46] Fuentes, A., Yoon, S., Kim, S., & Park, D. (2017). A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors*, 17(9), 2022. doi:10.3390/s17092022.
- [47] Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. *Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband*, 6-10 March, 2017, Stuttgart, Germany.
- [48] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience*, 2016. doi:10.1155/2016/3289801.
- [49] Atabay, H. A. (2017). Deep Residual Learning for Tomato Plant Leaf Disease Identification. *Journal of Theoretical & Applied Information Technology*, 95(24).
- [50] Guo, Y., Zhang, J., Yin, C., Hu, X., Zou, Y., Xue, Z., & Wang, W. (2020). Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming. *Discrete Dynamics in Nature and Society*, 2020. doi:10.1155/2020/2479172.
- [51] Al-Tarawneh, M. S. (2013). An empirical investigation of olive leave spot disease using auto-cropping segmentation and fuzzy C-means classification. *World Applied Sciences Journal*, 23(9), 1207–1211. doi:10.5829/idosi.wasj.2013.23.09.1000.
- [52] Entuni, C. J., Afendi Zulcaffle, T. M., & Kipli, K. (2020). Severity Estimation of Plant Leaf Diseases Using Segmentation Method. *Applied Science and Engineering Progress*, 14(1). doi:10.14416/j.asep.2020.11.004.
- [53] Chouhan, S. S., Singh, U. P., & Jain, S. (2020). Applications of Computer Vision in Plant Pathology: A Survey. *Archives of Computational Methods in Engineering*, 27(2), 611–632. doi:10.1007/s11831-019-09324-0.
- [54] Singh, V., & Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, 4(1), 41–49. doi:10.1016/j.inpa.2016.10.005.
- [55] Sengupta, D. (2019). Deep learning architectures for automated image segmentation. *EngrXiv (Engineering Archive)*, 1-70. doi:10.31224/osf.io/qt27a.
- [56] Sharma, P., Berwal, Y. P. S., & Ghai, W. (2020). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Information Processing in Agriculture*, 7(4), 566–574. doi:10.1016/j.inpa.2019.11.001.
- [57] Geetharamani, G., & Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers & Electrical Engineering*, 76, 323-338. doi:10.1016/j.compeleceng.2019.04.011.
- [58] Bhange, M., & Hingoliwala, H. A. (2015). Smart Farming: Pomegranate Disease Detection Using Image Processing. *Procedia Computer Science*, 58, 280–288. doi:10.1016/j.procs.2015.08.022.
- [59] Haque, I., Alim, M., Alam, M., Nawshin, S., Noori, S. R. H., & Habib, M. T. (2022). Analysis of Recognition Performance of Plant Leaf Diseases Based on Machine Vision Techniques. *Journal of Human, Earth, and Future*, 3(1), 129-137. doi:10.28991/HEF-2022-03-01-09.
- [60] Pinki, F. T., Khatun, N., & Islam, S. M. (2020). Visual Features based Paddy Leaf Disease Recognition, its Severity Detection and Remedy Prediction using K-means Clustering and AdaBoost. *Journal of Image Processing & Pattern Recognition Progress*, 7(3), 41-52.
- [61] Poonguzhali, R., & Vijayabhanu, A. (2019). Crop condition assessment using machine learning. *International Journal of Recent Technology and Engineering*, 7(6), 897–900.
- [62] Kumar, A., & Tiwari, A. (2019). A Comparative Study of Otsu Thresholding and K-means Algorithm of Image Segmentation. *International Journal of Engineering and Technical Research (IJETR)*, 9(5). doi:10.31873/ijetr.9.5.2019.62.
- [63] Kumari, Ch. U., Jeevan Prasad, S., & Mounika, G. (2019). Leaf Disease Detection: Feature Extraction with K-means clustering and Classification with ANN. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). doi:10.1109/iccmc.2019.8819750.