AgriDoc: Classification and Prediction of plant leaf diseases

Kushal H, K Swathi, Konanki Nithyusha, Lavanya A, Kalneedi srihari, Ravindranath R C*

Department of CSE, Presidency University, Bangalore.

* Corresponding author Kushal H.at Department of CSE, Presidency University, Bangalore;
e-mail: kushal.h1999@gmail.com, swathikonduru144@gmail.com, konankinithyusha@gmail.com, lavanyananda51@gmail.com, sriarikalneedi1999@gmail.com, ravindranath@presidencyuniversity.in

Abstract

With the growing population, there comes a great need to provide sufficient necessities for everyone. Here comes the question whether we have enough resources to provide necessities for everyone or not. It shows the importance of increasing agricultural production. There are a lot of reasons for the decrease in Agriculture production, one of the main factors is diseases/pests. Pests/diseases can damage the entire crop in a short time if not detected and diagnosed on time. Detection of crop diseases at an initial stage can help farmers diagnose the disease on time, hence increasing the productivity of the crop. This is possible with the implementation of advanced technologies like Deep Learning (DL) in the field of Agriculture. DL is being used in Agriculture for Crop Recommendation, Precision Agriculture, Disease detection, and Smart Irrigation etc. DL approach, precisely Convolution Neural Network (CNN) can be used to detect the leaf disease more precisely and accurately than humans. The proposed work uses various CNN architectures like AlexNet, MobileNet, ResNet50 and some CNN based models that are built from scratch for the detection and identification of leaf diseases of various crops. Once the classification is done, these architectures will then be compared based on their performance and accuracy. The best model will be chosen for deployment using Django framework to create a web application to make the model more readable and user friendly.

Keywords: CNN; Transfer Learning; Deployment; Deep Learning; Plant Disease;

1. Introduction

Agriculture plays a crucial role in the survival of living beings on earth. The agriculture sector contributes over 17% to the Indian Economy and provides employment to 70% of the households in India. With the growing population comes the great need to increase the productivity in Agriculture. In recent decades, all the sectors have been moving towards technology. There have been advancements in the Agricultural sector over the centuries (tractors and other equipment). Recently, advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), DL, and Internet of Things (IoT), etc., are being implemented in agriculture for different applications like Water management, precision agriculture, Crop-recommendation, yield prediction, disease detection, etc.

The overall productivity is affected by the pests and diseases on stem, leaf, and fruit. The leaf diseases are mainly caused by bacteria, fungi, viruses, and climate change. To manually detect the infection in the plants is difficult, leading to the crop's destruction by the time the farmers notice the symptoms of the disease. If the pest /disease is not recognized at the initial stage, it will be a considerable loss, especially for large-scale farmers. Recently, there has been impressive growth in the DL algorithms, which is a subset of ML, is everywhere from the seeding to harvesting the crop. It plays a significant role in increasing agricultural productivity. It is difficult for a farmer to detect the affected region and detect the disease. Hence, there is a great need for modern methods to classify and detect crop diseases at an early stage. Using DL methods, detecting and recognizing pests and diseases at an early stage is made easy using the suitable algorithm [1].

DL approaches are employed in this proposed work to detect and identify diseases in plant leaves at an early stage. To train the model, fourteen different species of plants were considered. The accuracy and precision of several CNN designs are compared and evaluated. The Django framework will be used to deploy the model with the best model.This paper is subdivided into several sections. Section II describes DL, particularly CNN and its layers. It also goes through all the technical aspects that are necessary to comprehend this work. The researchers' related work on this topic is described in Section III. The proposed work is described in Section IV. Section V contrasts the architectures built and the best model's deployment. The conclusion, as well as the future scope, are included in Section VI.

2. Deep Learning

DL is a subset of ML, an area of computer science concerned with developing algorithms capable of learning independently, based on experience, in the same way humans do. These algorithms, usually called Artificial Neural Networks (ANN), are inspired by the structure and functionality of the human brain. DL is also one of the trending fields of data science. DL is very useful in solving dimensionality problems. DL models are implemented using deep networks, which are neural networks with multiple hidden layers.

2.1 Convolutional Neural Network:

CNN is a type of DL network that is used for image classification and clustering, as well as to object detection [5]. Deep Convolution Neural Network (DCNN) gives better accuracy than other neural networks, so they are preferred more. Some of the applications of DCNN are identifying faces, tumors, street signs, video analysis, time series forecasting, Anomaly detection, Natural Language Processing (NLP) [2].

CNN consists of four types of layers: convolutional layers, pooling, activation function, and fully connected layer.

2.1.1. Convolution Layer:

This layer is used to extract the feature map, such as edges from the input image, by reducing it to a smaller size using a filter. All the information in the field of the input image is brought together into a single pixel. From the input image, a small size matrix is created. The output of the convolution layer will be a vector. There are different types of convolution layers for different purposes. However, a 2D convolution layer is used in the proposed work[3].

Equation for convolution layers:

$$\mathbf{Z= W^T . X+b} \tag{1}$$

2.1.2. Pooling Layer:

This layer follows the convolution layer and is used to determine the size of the convolution layer's output matrix. This layer can have filters of various sizes, though most people use 2x2 filters. This layer lowers the feature map's dimensionality by a factor of two. The two most frequent pooling functions are Average pooling and Maximum pooling.A brief illustration of pooling functions is shown in fig 1.

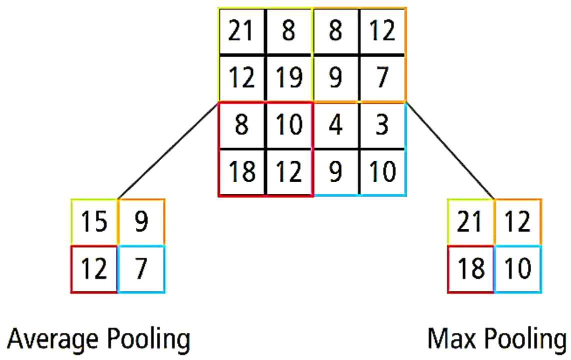

Fig 1. Pooling layers with 2x2 filters and stride 2.

Equations for max pooling and average pooling

- Max pooling: $P_{i,m} = \max_{n=1}^{G} \quad qi\ (m-1)\times s+n (2)$
- Average pooling: $P_{i,\,m} = r\sum_{n=1}^{G} \quad qi\ (m-1)\times s+n (3)$

Where G is pooling size, s is shift size, determines the overlap of adjoining pooling windows and r is a scale factor that can be learned

2.1.3. Activation Layer:

An activation function, which is also known as a transfer function, specifies how the weighted sum of a node's input is changed into output in a network layer. In most cases, all hidden layers use the same activation function. However, the output layer uses a different activation function. Rectified Linear Unit (ReLU), Sigmoid (Logistic), and Hyperbolic Tangent function are the three types of activation functions (Tanh). ReLU is widely used in CNN as the values less than zero are changed to zero, and the values greater than zero are changed.

Equations for activation function ReLU, Tanh, Sigmoid is as follows

ReLU: $\max \ (0, x)$ (4)

Tanh: $\tanh(x)$ (5)

Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$ (6)

2.1.4. Fully Connected Layer:

After finishing the convolution, pooling, and activation layers, the output from the preceding layer, the last obtained matrix, is sent into the fully connected layer as input. This layer is responsible for categorization and detection.

2.2. CNN Architectures
Many CNN architectures are developed to solve real-world problems. Each architecture has its significance. The architectures can also be built from scratch according to one's requirements and preferences.

These are some of the architectures of CNN:
1. LeNet-5
2. VGG
3. Architectural Design for CNNs
4. AlexNet
5. Inception and GoogleNet
6. Residual Neural Network(ResNet) etc.
7. MobileNet

In the proposed model, different architectures were used to build different models to compare the performance. These architectures include AlexNet, ResNet50, MobileNet and some models were built from scratch.

2.2.1. Model 1: Built from scratch model (6 layered)
This model is built from scratch and it has 6 layers that include 5 convolution layers the input dimension set as 64x64x3 where 64x64 is the size of the image and 3 is the number of channels that represents it as a coloured image. In the convolution layer, the padding parameter set is the same, the convolution layers include MaxPooling, dropout, Batch Normalization with axis set to 1 and activation function as ReLU and 1 fully connected layer that includes 1024 dense layers. The out layer consists of 38 class labelled followed by SoftMax as activation function. The overview of the proposed architecture is demonstrated in fig 2.

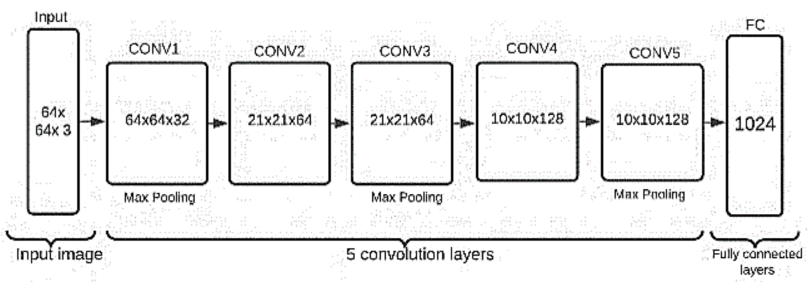

Fig2. Architecture of Model 1.

2.2.2. Model 2: Built from scratch model (5 layered)
This is built from scratch which has 5 layers that include 4 convolution layers that include MaxPooling, dropout, and activation function as ReLU and 1 fully connected layer that includes 512 dense layers, dropout, and activation function as ReLU as shown in fig 3. The output layer consists of 38 class labels as our dataset has 38 different categories followed by the SoftMax activation function. The input image has dimensions of 128 x 128 x 3 since the

image data was coloured.

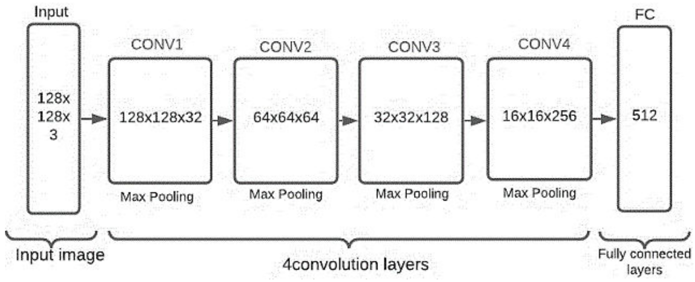

Fig3. Architecture of Model 2.

2.2.3. AlexNet:

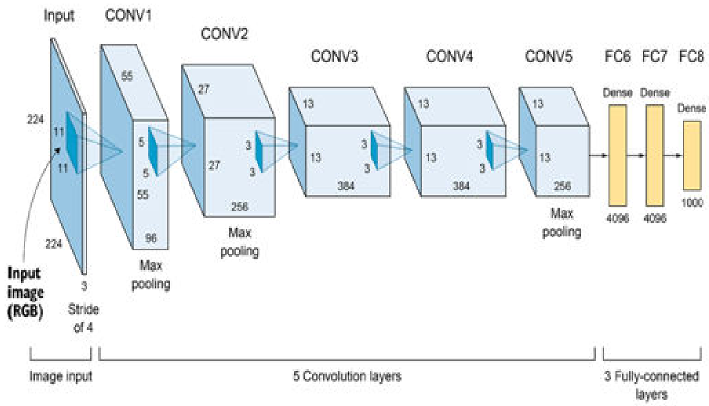

Fig 4. Internal working of AlexNet architecture.

AlexNet is a CNN model that correctly predicts the output using convolution and many pooling layers and the ReLU activation function, normalization layers, and dense layers. A ReLU activation function is coupled to every convolutional and fully connected layer. Five convolutional layers, three max-pooling layers, two fully linked layers, and a softmax layer make up this architecture, as shown in fig 4. Max pooling is used to reduce the feature map's dimensionality and obtain brighter pixels of the image. There are over 60 million parameters in this architecture.[4]

The image is initially scaled to 227*227 before being sent to the first convolutional layer, which comprises 96 kernels with a size of 11*11 applied. The edges from the given image are detected using these kernels, and the image is then forwarded to the second convolutional layer, in which 256 kernels are applied with 5*5 sizes. With every function in the kernel, the max-pooling layer with 3*3 pooling reduces the image dimensions.

This technique is done until you have a 3*3 input size with 256 kernels. In the end, fully connected layers are made up of 4096 neurons connected. Finally, the last layer has 1000 classes, which are further reduced to 38 classes based on the classes in our dataset. To accelerate the speed, the ReLU activation function, a nonlinear and non-saturating function, is used after all the convolution layers and the two fully connected layers have been applied. To deal with overfitting, a dropout function is added. It also improves network performance during the testing phase. To enhance the speed, stability and performance, batch normalization is used.

2.2.4. ResNet50

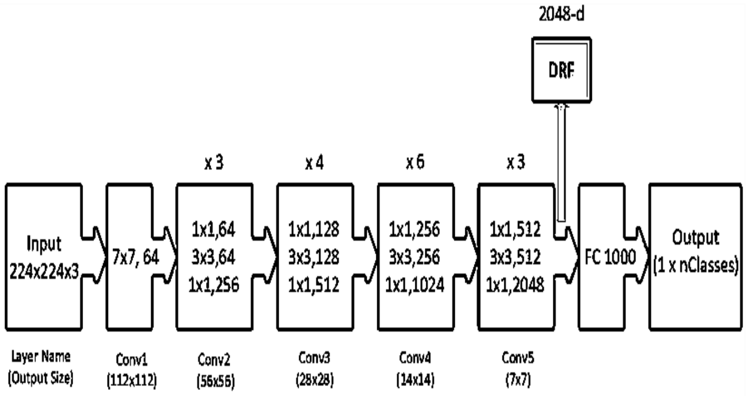

Fig 5. Overview of ResNet Architecture

Residual Networks called ResNet is a CNN which is used in most computer vision tasks. Before ResNet was applied only on image recognition however later it was used in non-computer vision tasks to achieve better accuracy. This was initially built on the ImageNet dataset. There are 1,000 different object classes in the ImageNet dataset, containing 1,281,167 training pictures, 100,000 test pictures, and 50,000 validation pictures.ResNet won the ImageNet challenge in 2015.

The ResNet model gave better results as a solution for vanishing gradients problems. ResNet50 is one of the forms of ResNet. This has 50 layers that include 1 Maxpool layer, 1 Average Pool layer, and 48 Convolution layers. The ResNet architectureconsists of app ResNet architecture starting from ResNet18, ResNet34, ResNet50, ResNet101, ResNet152. ResNet50 achieved a great decrease in error from 20.47% at top-1 error to 5.25% at top-5 error on trained on ImageNet dataset. The summary of ResNet architecture is shown in fig 5.

2.2.5.MobileNet

MobileNet model is a light weight specially built for mobile application by TensorFlow. To achieve a lightweight neural network model MobileNet applies depth wise separable convolutions. This reduces the parameters numbers compared to traditional networks. The depth wise separable convolutions consist of two operations Depth wise convolution and pointwise convolution.

By applying these will help the model in reducing the number of parameters, In the below figure we can see that in depth wise separable convolutions, consists of depth wise convolution at first and followed by Point wise convolution. depth wise convolution is channel wise DK-DK spatial convolution its computational cost is $\mathbf{Df^2 * M * Dk^2}$ whereas pointwise convolution is a1x1 to convert the dimension, its computational cost is $M * \mathbf{N * Df^2}$**.** In traditional CNN we have a single 3x3 convolution layer, ReLU and Batch normalization. Instead of these in MobileNet we separate convolution into 1x1 Point wise convolution and 3x3 depth wise convolution. The overview of this architecture is given in fig 6.

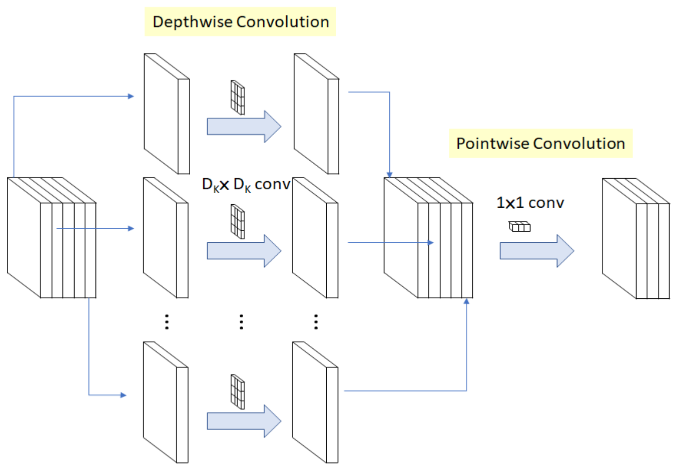

Fig 6. MobileNet Architecture Overview

3. Literature Survey

For this work, 25 papers were reviewed and analyzed to find some insights and know about the drawbacks of the existing works. Summary of each work is presented in this section.

In [6], Arias Bruhan et.al compared the outcome of different DL models that include vgg16, vgg19, Resnet50, Resnet50V2. These models were applied to artificial data as well as the real time data taken from the rice fields. On the artificial, Resnet50 has provided the best result with an accuracy of 75%. On the real-time data, RSNet101v2 has shown best performance with an accuracy of 86.79%. The proposed study showed that using DL models for the detection of pests & diseases in leaves is a useful yet revolutionary idea. This can be further developed into a mobile app that allows farmers to upload pictures of diseased plants for further diagnosis.

In [7], Omkar Kulkarni et.al used a DL model for the classification of leaf images into a healthy or diseased category based on the patterns. The objective of this model is to predict the disease based on textual similarity plant villa has been used to train the model. The CNN model was formulated to classify the species of crop & identify the diseases on images. It was observed that the inception model has performed better than MobileNet model in terms of both accuracy validation loss. This model with the smart plane applicant that provides user-friendly GUI.

In [8], Yang Zhang et.al proposed a model that detects healthy tomato leaves diseases. K means Clustering, Depth Residual Network methods were used for clustering and feature extraction processes. The experimental results have shown increased efficiency & higher detection accuracy than the original model. This could be further improved by including other characteristics such as fruits & stems to detect the diseases.

In[9], Janarthan et.al proposed a DL based architecture to detect the disease in citrus plants using sparse data. This proposed model is fast, lightweight, and accurate and allows farmers to use this architecture in their own mobile devices. In this, a patch-based classification network cluster prototype module & a simple neural network classifier has been used to detect the diseases precisely. The proposed model achieved 95.04% accuracy.

In [10], Yong Al etal. employedCNN to automatically detect or diagnose leaf diseases. The Resnet-v2 model was used to train the model in the beginning. The model had an overall detection accuracy of 86.1 percent. Following the training process, the WeChat applet for disease and pest recognition was designed and implemented. The actual test was carried out after this phase, which has shown promising results in identifying crop diseases.

In [11], Peng Jiang et al. suggested a DL strategy based on upgraded CNN for real-time detection of apple leaf diseases. By using rainbow concatenation and the GOOGLE NEST inception structure, this model is built using CNN. The suggested model (INAR-SSD) has a detection performance of 78 percent and a detection speed of 23.13 percent fps after being trained to detect five major apple illnesses. The model has been shown to be the most effective for detecting apple leaf diseases early and in real time.

In [12], Hammad Saleem, et al. investigated various DL architectures that are used alongside visualization approaches for the detection and classification of crop diseases, and several performance measures were applied to evaluate the strategies. This evaluation revealed various research gaps that may be filled to improve transparency in plant disease identification at the initial stages when symptoms are scarcely visible.

In [13], Ferentinos et al. created a CNN model for detecting and diagnosing plant illnesses by categorizing healthy and sick leaves. The model VGCNN had a 94.53 percent success rate in identifying the plant and its associated disease. This model could be improved to work in real-world farming situations.

In [14], Muhammad et al. executed difficult tasks to identify diseases in plant leaves. On 38 different classes of diseased and healthy leaves, single-shot multibox detector (SSD), Faster region-based convolutional neural network (RCNN), and religion-based fully convolutional networks (RCFN) meta-architectures were used. Using the tensor flow architecture for object detection. The SSD model that was trained using Adam optimizer had the highest mean average precision (map) of 73.07 percent during the experiment. After effectively identifying 26 various types of defective leaves and 12 types of healthy leaves in a single framework, the model's originality was proposed.

In [15], Parul Sharma et al. explored potential techniques for automating the diagnosis of plant diseases using independent data. CNN models were trained on segmented visual data to achieve this. In terms of accuracy and data, the models FCNN and SCNN were evaluated. When compared to the F-CNN model, the SCNN model achieved 98.6% accuracy when trained on segmental pictures. As more data sets become accessible in the future, the proposed model will perform well in the early detection of diseases.

In [16], Faithpraise Fina et al. exhibited the identification and recognition of pests using a combination of the k-means algorithm, clustering, and the correspondence filter. To detect the data set, the data space is divided into Voronoi cells, which search for clusters and thereby separating the pests from their environment. To detect pests, attributes such as (leaf, stem) are retrieved, and pest identification is done using correspondence filter, which achieves rotational invariance up to an angle of 360*, demonstrating the algorithm's efficiency for pest detection and recognition in plants.

.

In [17], Games Selvaraj etal. presented a model that uses a DCNN to detect disease and plants in a banana leaf. Large pre-scanned leaf data sets of banana plants were extracted from hotspots in Africa and southern India to achieve this. This detection model was built withDCNN architectures that were trained via transfer learning. The RSNet50 and InceptionV2 models outperformed the mobile Net v1 model, according to the findings. The model has a 90% accuracy rate, indicating that it could be valuable in the early detection of illnesses and pests. This might be taken further by using mobile apps to assist banana producers all around the world.

In [18], Anupam Bapat et al. created a plant leaf disease detection model based on the CNN framework that recognizes 14 distinct species and 26 modules. This was shown to be accurate to the tune of 99.35 percent. The proposed model makes advantage of TAD pro genesis, which uses mobile phones to detect diseases all around the world. The obtained results open the door to a variety of options, such as the number of fertilizers to be sprayed while considering numerous elements such as soil contents, humidity, water, wind speed, solar radiation level, and much more to gain a thorough grasp on the plant.

In [19], Sachin B et al. focused primarily on the identification of plant disease using CNN. They used data from soybean plant dataset to identify disease in the soybean plant using transfer learning. They employed pre-trained CNNlike AlexNet and GoogleNet(CNNs). They have trained for 649 diseased soybean leaf image samples and 550 healthy soybean leaf image samples. They achieved an accuracy of 98.75 percent for the AlexNet CNN-based model and 96.25 percent for the GoogleNet CNN-based model following the training.

In [20], Chen et al. studied the detection of plant leaf disease using pre-trained models such as VGGNet on ImageNet and the Inspection model. Instead of starting the training from the beginning by randomly initializing the weights, they used transfer learning models on a huge dataset to accomplish so. They discovered that ImageNet outperformed other state-of-the-art approaches, achieving 91.83 percent validation accuracy and 92 percent accuracy in identifying the class of rice plant images.

In [21], Surampalli Ashok et al. used DL algorithms to detect early plant leaf disease. They used image processing approaches based on image segmentation, clustering, and open-source algorithms to identify disease in tomato plant leaves. OpenCV was used to test the proposed technique. The CNN technique was required to obtain hierarchical features in this model, which achieved 98% overall accuracy.

In [22], Muhammad Hammad Saleem et al. provided a comprehensive analysis of plant disease classification based on DL. They have accomplished this in two steps. 1 The best CNN was discovered by comparing well-known CNN models with updated and cascaded/hybrid versions of several of the DL models given in recent research. 2. It was intended that by employing several DL optimizers to train the best-obtained model, the best-obtained model's performance would improve. All of these DL models were trained on a plant village dataset, which contains 26 illnesses from 14 distinct plant species.All of these DL models were trained on a plant village dataset, which contains 26 diseases from 14 distinct plant species. The F1-score achieved was 0.9978, and the validation accuracy was 99.81 %.

In [23], Kaizhou et al. aimed on employing a CNN to detect and identify plant disease in order to prevent the loss of Ginkgo biloba in the future. They employed the VGGNet 16 and Inception V3 models for this. They train 1322 original photos from the lab and 2408 original images from the field after the data has been pre-processed. Using the VGG model, they were able to obtain 98.44 % in the lab and 92.19 % accuracy in the field. Using the Inception V3 model, 93.2 % accuracy was achieved in the field and 92.3 % accuracy was achieved in the lab.

In [24], Shruthi et.al conducted a survey for the detection of plant diseases caused by bacteria, viruses, fungi by using machine learning, precisely deep learning techniques. According to the survey, it was observed that Convolutional neural network has shown better performance compared to other techniques like Support Vector Machine (SVM) Classifier,K-Nearest Neighbors (KNN) Classifier, ANN Classifier, FUZZY Classifier (Fuzzy C-Means Classifier), DL. More algorithms, such as decision trees and Nave Bayes classifiers, have the potential to be utilized for disease detection in plants and to aid farmers in the automatic diagnosis of diseases in diverse crops.

In [25], PranaliK.Kosamkar et.al proposed a system that detects the disease and recommends the pesticide to be used accordingly. Image preprocessing and feature extraction techniques were applied on images followed by a CNN to classify the disease and recommend the remedy accordingly using TensorFlow technology. DL and Java web services are the technologies used for the implementation of the model. CNN with 3,4,5 layers are used for training the model and to interact between these systems with ease, the android application along with JWS was used. It was observed that the 5-layer CNN model achieved higher accuracy with 95.05% training accuracy and 89.67% validation accuracy.

In [26], Ashish et al. developed an automatic system based on CNN for detecting the diseases of potato leaves that uses the VGG19 pre-trained model for feature extraction. After extracting features from pre-trained models are passed into different classifiers such as SVM, KNN, and Logistic regression and it classifies the healthy and unhealthy leaves. Orange data mining tool is used for classification and feature extraction. They have used a plant village dataset that contains healthy leaves and unhealthy leaves such as Late Blight leaves & Early Blight leaves. Finally, they achieved 97.8% accuracy for the VGG19 model along with the Logistic regression classifier.

In [27], Oppenheim D et al. focused on image-based disease detection of plant leaf diseases using DL, precisely CNN model. They have used potato plant leaves to identify the tuber disease using the CNN architecture. Firstly, they have used cameras to capture the images of leaves, and then an image labeler application was used to mark and label the visual symptoms of diseases and the CNN algorithm is used for image classification along with that they have used data augmentation. Their accuracy for correct classification was in the range of 83% to 96% depending on the training data.

In [28], Mohamed F et al. focused on DL techniques such as feature extraction and transfer learning for the detection of leaf diseases of the plants taken from a dataset with 14 species including healthy and diseased categories for training the model. They did a comparison between three architectures as ResNet 50, Google Net, VGG-16 along with the two classifiers such as SVM and KNN and finally they obtained good accuracy as 97.92% for the VGG-16 model along with the SVM classifier.

In [29], SrdjanSladojevic et al. proposed a CNN model for the classification and detection of plant leaf diseases. They have chosen a dataset with 13 types of diseases and also healthy leaves to train the model. Fine-tuning and data augmentation techniques were used along with DCCN models. Finally, they obtained a good accuracy of 96.3%. In the future, they want to use drones to capture images of a wide range of land to detect the disease of leaves.

In [30], Md. Tariqul Islam et al. came up with a CNN model for the detection of plant leaf diseases. Image processing techniques were used for disease detection in images. CNN model was used for training the model and the dataset consists of both healthy and diseased leaves including strawberry, grapes, and potato leaves. Finally, they obtained an accuracy of 94.29%. In future, they want to implement the open multimedia system and to implement the software that automatically detects disease and gives the required solution.

4. Proposed System-AgriDoc

This section contains methodology of the proposed model along with architecture. Also, the dataset used is thoroughly discussed in this section. The model Agri-Doc is used to classify and predict the disease of the plant by taking plant leaf image as the input. The model is divided into various steps to get the best accuracy for predicting the disease.

4.1. Dataset Discussion

The Plant Village Dataset from Kaggle was used in the proposed model. As indicated in table 1, this dataset contains 54305 images, which include images from 38 different categories pertaining to ten different types of crops. Colored photos were used because, as compared to grey scale images, as they provide more accuracy. Images were considered from various angles and conditions, some of which are shown below. Training and testing datasets are created from the images. When the model is trained with more images, the accuracy improves. For training and testing multiple CNN models, the images are partitioned into 80 percent -20 percent and 90 percent -10 percent, as shown in Table1.

4.2.Proposed Methodology

There are four steps involved in the proposed work that includes Data Acquisition, Data pre-processing, Classification and Detection using CNN models, comparison of the models, deployment of the best model.

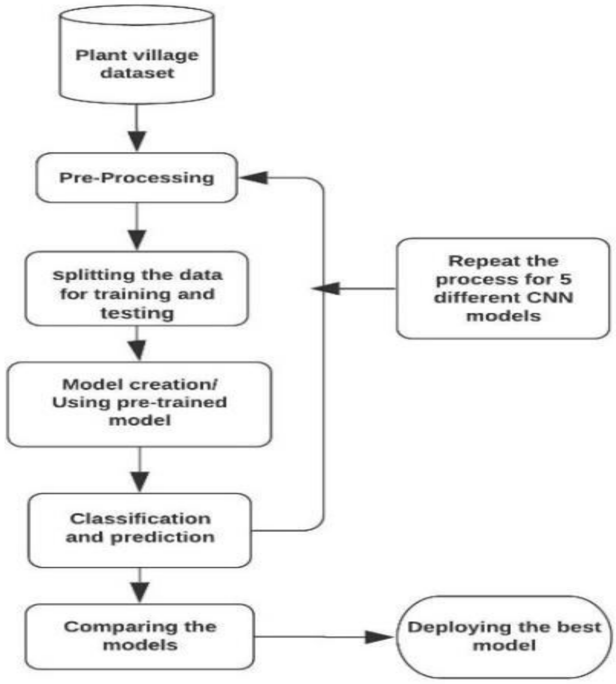

Fig 7. Workflow of AgriDoc Architecture

4.2.1. Data Acquisition:

Datasets of three different plants(potato, tomato, and pepper) are to be extracted and structured to meet the criteria. The dataset contains 54305 images having 38 different categories belonging to 10 different crops. The images are converted to colour, Gray scale and segmented images, but coloured images were more preferred as they give better accuracy.

4.2.2. Data pre-processing:

Initially when the images were obtained, the background of images may vary and may have non uniform lighting that affects the accuracy of the model. Hence, pre-processing is highly important for the removal of noise and image segmentation which helps in achieving higher accuracy for the proposed CNN model. The varying background issue can be dealt by segmenting only the relevant part of the image that is required for the prediction. After segmenting, the images are converted into grey scale images first and then all the images are given the black background, therefore highlighting the relevant part of the image.

4.2.3. Classification and Prediction using CNN

To detect the diseases in a plant's leaf, CNN are used, because CNN, which is an evolution of ANN, performs better on image data. Convolution, Pooling, flattening and fully connected layers are the main functions of CNN, which are used to detect the edges of image patterns and reduce the image size. Two different CNN architectures are used. Moreover, the model is trained using TensorFlow and Keras(TensorFlow high-level API for training DL models).

S.NO	Species	Diseases	Total Images	Train_test split(90%-10%)		Train test split (80% - 20%)	
				No of training images (90%)	No of testing images(10%)	No of training images (80%)	No of testing images (20%)
1.	**Tomato**	Tomato Yellow Leaf Curl Virus	5357	4821	536	4286	1071
2.		Tomato Target Spot	1404	1264	140	1123	281
3.		Tomato Septoria Leaf Spot	1771	1594	177	1417	354
4.		Tomato Healthy	1591	1432	159	1273	318
5.		Tomato Leaf Mold	952	857	95	762	190
6.		Tomato Late Blight	1909	1718	191	1527	382
7.		Tomato Mosaic Virus	373	336	37	298	75
8.		Tomato Early Blight	1000	900	100	800	200
9.		Tomato Two Spotted Spider Mite	1676	1508	168	1341	335
10.		Tomato Bacterial Spot	2127	1914	213	1702	425
11.	**Squash**	Squash Powdery Mildew	1835	1652	183	1468	367
12.	**Blueberry**	Blueberry Healthy	1502	1352	150	1202	300
13.	**Raspberry**	Raspberry Healthy	371	334	37	297	74
14.	**Cherry**	Cherry Healthy	854	769	85	683	171
15.		Cherry Powdery Mildew	1052	947	105	842	210
16.	**Corn**	Corn Common Rust	1192	1073	119	954	238
17.		Corn Healthy	1162	1046	116	930	232
18.		Corn Gray Leaf Spot	513	462	51	410	103

19.		Corn Northern Leaf Blight	985	887	98	788	197
20.	**Peach**	Peach Healthy	360	324	36	288	72
21.		Peach Bacterial Spot	2297	2067	230	1838	459
22.	**Orange**	Orange Huanglongbing	5507	4956	551	4406	1101
23.	**Grape**	Grape Leaf Blight	1076	968	108	861	215
24.		Grape Healthy	423	381	42	338	85
25.		Grape Black Measles (Esca)	1383	1245	138	1106	277
26.		Grape Black Rot	1180	1062	118	944	236
27.	**Pepper Bell**	Pepper Bell Healthy	1478	1330	148	1182	296
28.		Pepper Bell Bacterial Spot	997	897	100	798	199
29.	**Soybean**	Soybean Healthy	5090	4581	509	4072	1018
30.	**Potato**	Potato Early Blight	1000	900	100	800	200
31.		Potato Late Blight	1000	900	100	800	200
32.		Potato Healthy	152	137	15	122	30
33.	**Strawberry**	Strawberry Leaf Scorch	1109	998	111	887	222
34.		Strawberry Healthy	456	410	46	365	91
35.	**Apple**	Apple Cedar Rust	275	248	27	220	55
36.		Apple Scab	630	567	63	504	126
37.		Apple Healthy	1645	1481	164	1316	329
38.		Apple Black Rot	621	559	62	497	124
Total Count			**54305**	**48877**	**5428**	**43447**	**10858**

Table1: Number of classes and species used in the proposed model.

4.2.4. Comparison of CNN models

Different CNN based models were built in this work. Some were built from scratch, and some were pre trained models like resnet50, MobileNet and AlexNet. The performance of these models is compared based on their accuracy, preciseness, time, and space efficiency etc. The best performing model will be chosen for deployment to make it easier for farmers to use. The comparison of models is represented in table 2 and is discussed in further sections.

Evaluation criteria	Model 1	Model 2	Model 3 (AlexNet)	Model 4 (ResNet 50)	Model 5 (MobileNet)
Image Size	64 x 64	128 x128	227x227	128x128	224x224
Data split	80%, 20%	90%, 10%	90%, 10%	90%, 10%	80%, 20%
Accuracy	92.6%	93.2%	94.4%	27.8%	86.5%
Validation accuracy	92.8%	90.2%	92.4%	36.9%	82.63%
Epoch	13	24	35	60	10
Time per epoch	10 min	37.15 min	67.3 min	48.1 min	8.75min
Time taken	10*13= 2.1 hrs	37.15*24= 14.8 hrs	35*67.3 = 39.25 hrs	60*48.1= 48.1 hrs	10*87.5= 1.45 hrs
No of layers	6	5	8	50	28

Table2: Performance Evaluation of CNN models.

4.2.5. Deployment

The model is deployed using the Django framework. After acquiring the best result from the trained model. This model is saved and the saved model which contains weights of the trained model is used for deployment to detect the disease in a plant using the Django framework.

5. Implementation

Model 01:

.

Layer	Feature Map	Size	Kernel Size	Activation
Convolution	32	64x64x32	3x3	ReLu
Pool/Max	32	21x21x32	3x3	ReLu
Convolution	64	21x21x64	3x3	ReLu
Convolution	64	21x21x64	3x3	ReLu
Pool/Max	64	10x10x64	2x2	ReLu
Convolution	128	10x10x128	3x3	ReLu
Convolution	128	10x10x128	3x3	ReLu

Pool/Max	128	5x5x128	2x2	ReLu
FC	--	1024	--	SoftMax

Table3. Tabular representation of Model 1parameters

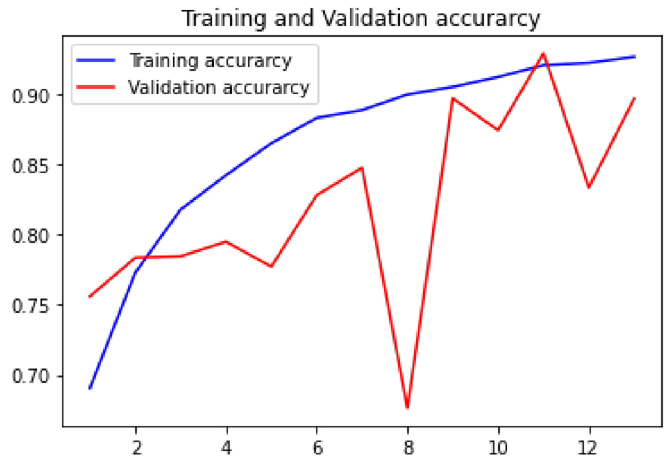

Fig 8. Training and Validation accuracy

The graph in fig 8 represents the fluctuation of training and valuation accuracy. We can see that the training accuracy increases evenly whereas there were huge variations in validation accuracy throughout the graph.

Model 2:

Layer	Feature Map	Size	Kernel Size	Activation
Convolution	32	128x128x32	3x3	relu
Pool/Max	32	64x64x32	2x2	relu
Convolution	64	64x64x64	3x3	relu
Pool/Max	64	32x32x64	2x2	relu
Convolution	128	32x32x128	3x3	relu
Pool/Max	128	16x16x128	2x2	relu
Convolution	256	16x16x256	3x3	relu
Pool/Max	256	8x8x256	2x2	relu
FC	--	512	--	Softmax

Table 4: Tabular representation of CNN model 2 parameters.

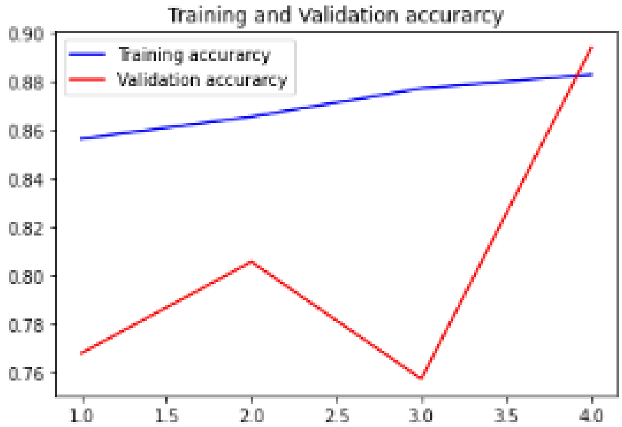

Fig 9. Training and validation accuracy of Model 2.

The graph in fig 9 represents the fluctuation of training and valuation accuracy. We can see the trainingaccuracyincreases evenly whereas validation accuracy low in the beginning but after more epochs it achieved more accuracy than train accuracy.

Model 3

Layer	Feature Map	Size	Kernel Size	Stride	Activation
Convolution	96	55x55x 96	11x11	4	relu
Pool/Max	96	27x27x 96	3x3	2	relu
Convolution	256	27x27x 256	5x5	1	relu
Pool/Max	256	13x13x 256	3x3	2	relu
Convolution	384	13x13x 384	3x3	1	relu
Convolution	384	13x13x 384	3x3	1	relu
Convolution	256	13x13x 256	3x3	1	relu
Pool/Max	256	6x6x 256	3x3	2	relu
FC1	--	4096	--	--	relu
FC2	--	4096	--	--	relu
FC3	--	1000	--	--	Softmax

Table 5. Representation of AlexNet architecture parameters.

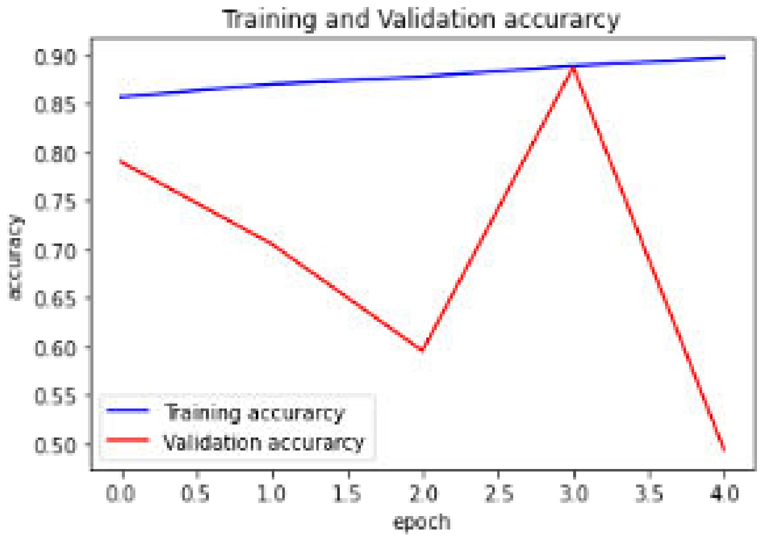

Fig 10. Training and Validation accuracy of AlexNet.

The graphin fig 10 represents the fluctuation of training and valuation accuracy. We can see the training accuracy increases evenly whereas validation accuracy low in the beginning and reaches accuracy as training accuracy then decreases. However, we have saved weights where accuracy was best in whole training.

Model 4

Layer's name	**Output size**	**ResNet 50**
Conv1	112x112	7x7, 64, stride 2 3x3 max pooling, stride 2
Conv2.x	56x56	$\begin{bmatrix} 1\times 1 & 64 \\ 3\times 3 & 64 \\ 1\times 1 & 256 \end{bmatrix} \times 3$
Conv3.x	28x28	$\begin{bmatrix} 1\times 1 & 128 \\ 3\times 3 & 128 \\ 1\times 1 & 512 \end{bmatrix} \times 4$
Conv4.x	14x14	$\begin{bmatrix} 1\times 1 & 256 \\ 3\times 3 & 256 \\ 1\times 1 & 1024 \end{bmatrix} \times 6$
Conv5.x	7x7	$\begin{bmatrix} 1\times 1 & 512 \\ 3\times 3 & 512 \\ 1\times 1 & 2048 \end{bmatrix} \times 3$
	1x1	Average pool, 1000-d fc

Table 6. Representation of ResNet architecture parameters.

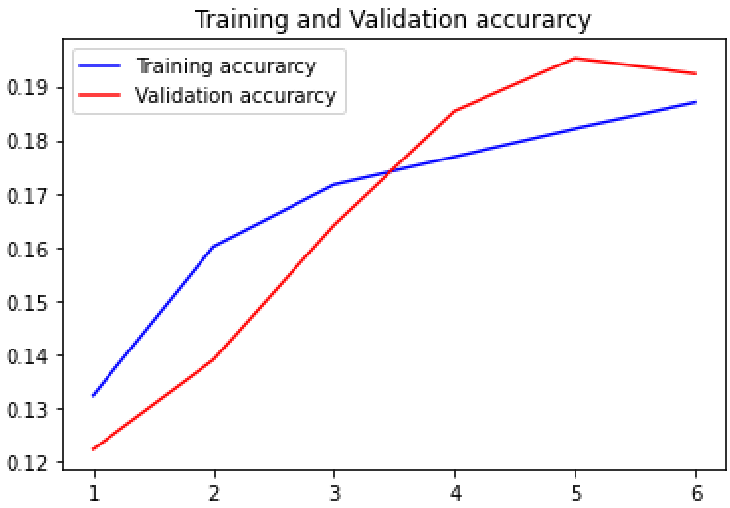

Fig 11. Training and Validation accuracy of ResNet architecture.

The graphin fig 11 represents the fluctuation of training and valuation accuracy. We can see that training accuracy increases throughout the graph whereas validation accuracy was low at the beginning and after 3 epochs validation accuracy outraced the training accuracy and became the highest by the end of the graph.

6. Model Evaluation and Deployment

6.1. Comparison

The performance of the proposed CNN models is evaluated and compared to choose the best one. CNN architectures like MobileNet, AlexNet, ResNet50 were considered along with some models that are built from scratch. Different parameters are chosen for comparison as shown below.

6.2.Deployment:

Django is one of the most popular frameworks to create your own web applications hassle-free. It is a high-level python-based web framework that is helpful in creating complex database driven websites in a less complex way. It has many features to simplify the process of web development. It is simple, time-effective, secure, up-to-date, backward compatible and suits any kind of project.

Fig 12. Home Page of Web Application

In our work, the Django framework was used for deployment because of its ease of use and best features. Once the convolution neural network models are evaluated based on their performance, the best model with the highest accuracy was chosen to further deploy the model into a web application as shown below.

This web page contains buttons to choose files, upload and to clear the input as shown in fig 12. The types of crops and their diseases that are covered are displayed in table form in the web page so that the users could search for the availability of the crop they want to check the disease for. Colour images are preferred for accurate prediction.

Fig 13. Prediction Page of Web Application

Once the image is uploaded, the user will be directed to the prediction page as shown in fig 13.
This page displays the uploaded image, crop species and the disease type. Once the user gets the result and if he wants to do the same for another image, there is a button 'Make new prediction' in the prediction page which lets the user go back to the main page to make new predictions.

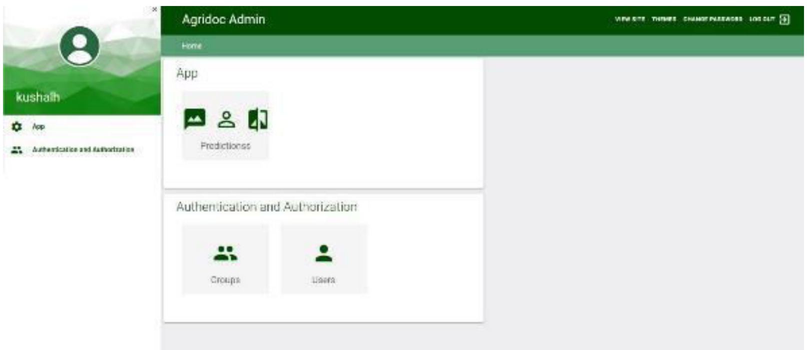

Fig 14.Admin Page of Web Application

The application also has an admin page which is linked to a database that stores the past predictions as shown in fig 15. The admin can clear the predictions too when necessary. The stored data can be used to count the wrong predictions which helps in improving the application further. The idea is to get feedback from the user after getting the prediction. The glimpse of the admin page is shown in fig 14 and fig 15.

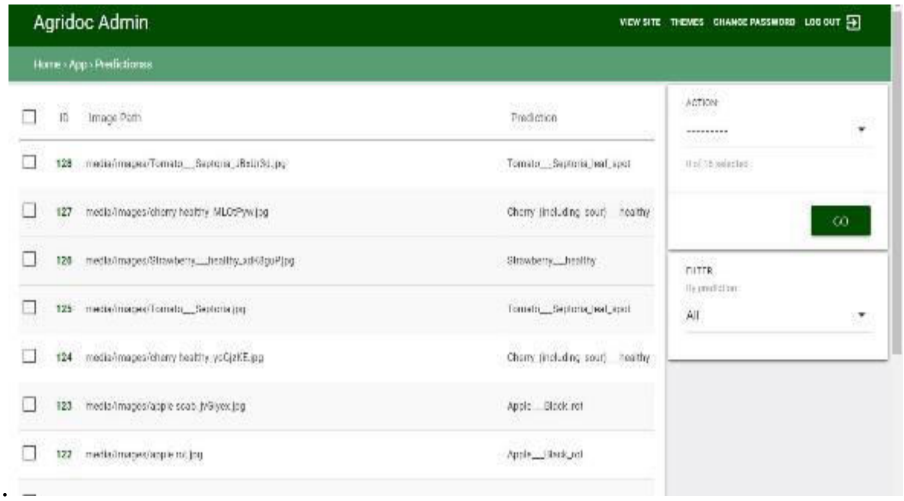

Fig 15. Overview of Database page

CONCLUSION AND FUTURE SCOPE

Agriculture playsa key role in our survival as well as in the economy of the country because it is the main source of food in any country. To provide food to everyone, it is necessary to increase productivity in Agriculture. Crop diseases are one of the reasons for the decline in Agricultural productivity. It is difficult to detect the pests and diseases in plant leaves at the initial stage manually. But with the help of CNN, crop disease detection and classification are made easy with accurate prediction. The proposed work AGRIDOC is meant for detection and identification of leaf diseases. Five CNN models were built for the plant disease detection. These models include AlexNet, MobileNet, ResNet and other two models are built from scratch. AlexNet achieved 92% accuracy, the built from scratch models achieved 90% and 92% accuracy, whereas MobileNet achieved 82% accuracy and ResNet achieved 37% accuracy. The model with 92% accuracy was deployed using Django framework to make the model more readable and user-friendly. Hence, the proposed model AGRIDOC can be used for accurate detection and identification of leaf diseases with 92% accuracy to help the farmer make suitable decisions to enhance the productivity in Agriculture.

AGRIDOC can be further developed by training a model for more epochs to give higher accuracy. It can be further developed into a mobile application to make it easy to use. It would be even better to use real time images for the prediction. Different CNN models can be used to check the accuracy and precision of the prediction. Also, more crops can be covered for the prediction of the diseases. There is a possibility of recommending a remedy to the disease using organic methods in the prediction page. With better resources and high computational power, it is possible to achieve better results than the state-of-the-art CNN models.

References:

[1] Kosamkar, Pranali& Kulkarni, V.Y. & Mantri, Krushna &Rudrawar, Shubham &Salmpuria, Shubhan& GadKosamkar, Pranali K., et al. "Leaf disease detection and recommendation of pesticides using convolution neural network." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[2] Patel, Pruthvi P., and Dineshkumar B. Vaghela. "Crop Diseases and Pests Detection Using Convolutional Neural Network." 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2019.

[3] Hari, S. Santhana, et al. "Detection of plant disease by leaf image using convolutional neural network." 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, 2019.

[4] Madhulatha, G., and O. Ramadevi. "Recognition of Plant Diseases using Convolutional Neural Network." 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2020.

[5] Kosamkar, Pranali K., et al. "Leaf disease detection and recommendation of pesticides using convolution neural network." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[6] Burhan, SyedaAiras, et al. "Comparative Study Of Deep Learning Algorithms For Disease And Pest Detection In Rice Crops." 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE, 2020.

[7] Kulkarni, Omkar. "Crop disease detection using deep learning." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[8] Zhang, Yang, Chenglong Song, and Dongwen Zhang. "Deep learning-based object detection improvement for tomato disease." IEEE Access 8 (2020): 56607-56614.

[9] Janarthan, Sivasubramaniam, et al. "Deep metric learning based citrus disease classification with sparse data." IEEE Access 8 (2020): 162588-162600.

[10] Ai, Yong, et al. "Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments." IEEE Access 8 (2020): 171686-171693.

[11] Jiang, Peng, et al. "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks." IEEE Access 7 (2019): 59069-59080.

[12] Saleem, Muhammad Hammad, Johan Potgieter, and Khalid Mahmood Arif. "Plant disease detection and classification by deep learning." Plants 8.11 (2019): 468.

[13] Ferentinos, Konstantinos P. "Deep learning models for plant disease detection and diagnosis." Computers and Electronics in Agriculture 145 (2018): 311-318.

[14] Saleem, Muhammad Hammad, et al. "Image-Based Plant Disease Identification by Deep Learning Meta-Architectures." Plants 9.11 (2020): 1451.

[15] Sharma, Parul, Yash Paul Singh Berwal, and WiqasGhai. "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation." Information Processing in Agriculture 7.4 (2020): 566-574.

[16] Fina, Faithpraise, et al. "Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters." International Journal of Advanced Biotechnology and Research 4.2 (2013): 189-199.

[17] Selvaraj, Michael Gomez, et al. "AI-powered banana diseases and pest detection." Plant Methods 15.1 (2019): 1-11.

[18] Bapat, A., S. Sabut, and K. Vizhi. "Plant leaf disease detection using deep learning." Int. J. Adv. Sci. Technol 29.6 (2020): 3599-3605.

[19]. J Jadhav, Sachin B., Vishwanath R. Udupi, and Sanjay B. Patil. "Identification of plant diseases using convolutional neural networks." International Journal of Information Technology (2020): 1-10.

[20] Chen, Junde, et al. "Using deep transfer learning for image-based plant disease identification." Computers and Electronics in Agriculture 173 (2020): 105393.

[21] Ashok, Surampalli, et al. "Tomato Leaf Disease Detection Using Deep Learning Techniques." 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.

[22] Saleem, Muhammad Hammad, Johan Potgieter, and Khalid Mahmood Arif. "Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers." Plants 9.10 (2020): 1319.

[23] Li, Kaizhou, et al. "Using deep learning for Image-Based different degrees of ginkgo leaf disease classification." Information 11.2 (2020): 95.

[24] Shruthi, U., V. Nagaveni, and B. K. Raghavendra. "A review on machine learning classification techniques for plant disease detection." 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). IEEE, 2019.

[25] Kosamkar, Pranali K., et al. "Leaf disease detection and recommendation of pesticides using convolution neural network." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[26] Tiwari, Divyansh, et al. "Potato leaf diseases detection using deep learning." 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2020.
[27] Oppenheim, Dor, et al. "Using deep learning for image-based potato tuber disease detection." Phytopathology 109.6 (2019): 1083-1087.

[28] Mohameth, Faye, ChenBingcai, and Kane AmathSada. "Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village." Journal of Computer and Communications 8.6 (2020): 10-22.

[29]Sladojevic, Srdjan, et al. "Deep neural networks based recognition of plant diseases by leaf image classification." Computational intelligence and neuroscience 2016 (2016).

[30] Md. Tariqul Islam, 2020, Plant Disease Detection using CNN Model and Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 10 (October 2020)