

ANALYSIS OF CNN LEAF-BASED DISEASE CLASSIFICATION ON DIFFERENT PLANT SPECIES BY SEQUENTIAL&DENSENET LEARNING MODEL

Savitha S¹, Sreedevi N², Nithin Kumar M³, Mohammad Abdussamad⁴, Ashish MB⁵

¹⁻⁵CMR Institute of Technology, Bengaluru

Abstract- According to world statistics India stands the world's second-largest producer of crops like wheat, rice, dry fruits, numerous vegetables, and fruits. Being the world's fifth-largest producers and exporters of various crops, grains, pulses, and other cultivated products, which constitutes the main source from which a large portion of India's Economy comes. One of the major reasons for Crop failure in our country is due to diseases that affect the crops caused by various factors like environment, pathogen, leaf host, etc. Plant disease detection and prevention must be done at the early stages to save the crops. This paper presents a comparative analysis of the CNN leaf-based disease classification model on different plant species. In this paper, we propose a comparison of sequential models like the CNN and DenseNet for performing disease identification in plants efficiently based on learning approaches using a pre-trained convolutional neural network (CNN's). Sequence models are the machine learning models that input or output sequences of data and show the accuracy of the sequential model and Dense Net model in terms of predicting the leaf diseases.

Keywords- Machine Learning, Sequential Model, CNN- convolutional neural network, RNN-Recurrent neural network, DenseNet

1.Introduction

Diseases that affect plants have turned into a major issue as they can cause serious loss in the quality and the production of agricultural goods. In India, an estimated percentage loss of crop production is around 15-20% due to pests, weeds, and diseases. Many farmers lack the knowledge of adopting new and recent technologies in agriculture [1]. Various factors affect leaves like environment, pathogen, and leaf host. In [2], the author presents various plant diseases like viral, fungal, and bacterial. Bacterial diseases include symptoms like leaf spots, cankers, scabs, etc. The most commonly affected bacterial disease is leafspot [3] as in fig1. Viral plant diseases include symptoms like crinkled leaves, Mosaic pattern on the leaf, discoloration of leaves, plant stunting. The most common viral diseases are seen in [4] are spotted wilt virus in tomatoes, Cucumber, and Cauliflower mosaic virus as in fig2, Brome mosaic virus, Plum pox virus. Fungal diseases are most commonly seen in vegetables and the symptoms include rust leaf, stem rust, white mold, etc. Viral plant diseases include Fusarium rots, Powdery and Downy mildew, rust, Sclerotinia rots, Botrytis rots and Sclerotium rots [5].



Fig1: Bacterial Leaf spot.



Fig2: Cucumber Mosaic virus.



Fig3: Fungal plant disease in Raspberries, powdery mildew, and rust spores.

Identification of plant diseases in an automated fashion is an important topic of research as it may be advantageous in observing the growth of crops, and detect the symptoms of diseases immediately as they appear on plant leaves. Therefore, a swift, automated, cheap, and accurate method to identify plant disease cases is an essential need in the agricultural world.

Studies show that Machine learning and deep learning algorithms and networks can be implemented to achieve the problem. Various machine learning techniques that have been implemented in agricultural researches include popular models like Artificial and Deep Neural Networks such as ANNs, DL, and Support Machine Vectors. The main feature of a good machine learning model is accuracy. In [6] a model is proposed which uses a Machine learning model which is used for categorization and feature selection secondly it uses an image processing model for capturing images, subjecting the image to noise removal, segmentation, and feature extractions. The use of image processing in the detection and prevention of leaf disease can help assist the researchers in recognizing the regions of infected leaf and stem, measuring and finding the region and shape of the affected area, determining the discoloration or change in colour of the infected leaf or stem [8]. The classification accuracies and the model used in [7] of new indices for healthy and infected leaves are 86.5%, 85.2%, 91.6%, and 93.5% respectively with an average accuracy of 89.2 %.

In this paper, the dataset we use is a set of leaves that are taken from Indian crops via a Kaggle image dataset having around 8152 images for tomato, potato, and bell pepper leaves. Our model has been tested on five diseases that affect plants. The dual functionality of feature extractors and classifiers is used by these Convolutional Neural Networks. It was observed that there was a possibility to increase the efficiency of the network by incorporating the CNN network with a shallow classifier.

The contributions of the paper are: (1) The base paper uses only one plant specimen i.e., soyabean, we have used 3 plant species tomato, potato and bell pepper. (2) Many diseases have been taken into account and a better model with high prediction and accuracy was built with fine tuning and optimization. (3) DenseNet model was used with performance optimizers and an accuracy of 98% was achieved which predicts the diseases with higher accuracy.

2. Proposed Work.

Detection and classification by image processing can be outlined as shown below

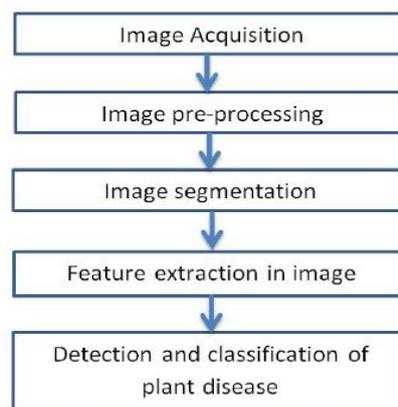


Fig 4 – A basic flow of the image processing

1) Image Acquisition

The images of the plant leaf are captured through the digital camera, smartphone, or a public repository. The image can be in a format like .bmp, .jpg, .gif as input to the model presented in this paper. This image is in RGB (Red, Green, and Blue) form. Color transformation structure for the RGB leaf image is created, and then, a device-independent color space transformation for the color transformation structure is applied.

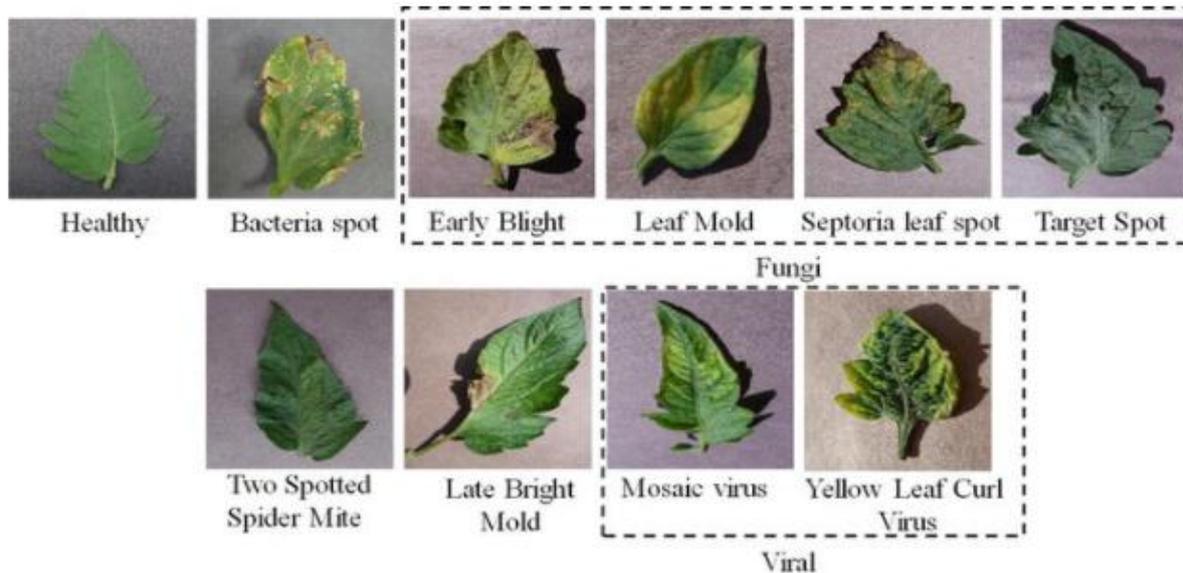


Fig5 – Sample images of healthy and diseased tomato leave taken from the datasets [13].

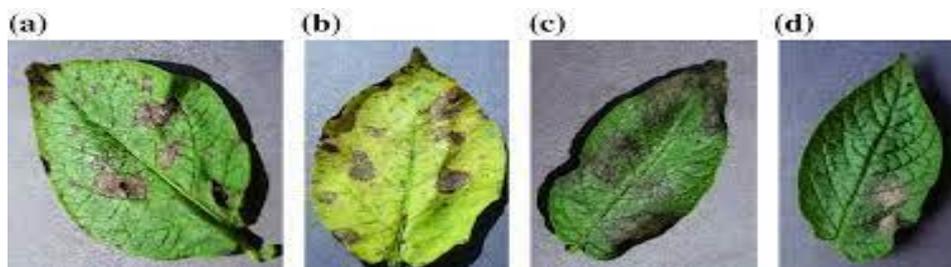


Fig6- Sample image of diseased potato leaves from datasets.

- 2) Image Preprocessing: The image obtained in real can be subjected to water droplets, dust, and spores as noise which acts as noise, and unwanted objects which must be removed before processing further during this stage it involves methods like image resizing, smoothing, increase/decreasing the contrast of the image enhancement, etc. In our work presented we resize the image to 257*257 pixels.
- 3) Image Segmentation: In this step, the image is either further divided or partitioned into multiple segments or grouped to various algorithms are used in this step like K-mean clustering, OTSU method, converting RGB to HSI or HSV.

Converting RGB to HSI (Hue, Saturation, Intensity) involves the following steps.

Let Red, Green, Blue (R,G,B) be the color coordinates with a value between 0 and 1 then the HSI value is calculated as.

- 1) The RGB image is read.
- 2) The RGB image is represented in the range[0,1].
- 3) HSI component is found.

$$\rho = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)^{1/2}]} \right\} \dots \dots \dots (1)$$

- 4) Calculate the H (Hue) value.

$$h = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \dots\dots\dots(2)$$

5) Calculate the S(Saturation) value.

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)] \dots\dots\dots (3)$$

6) Calculate the I (Intensity) value.

$$I = \frac{1}{3}(R + G + B) \dots\dots\dots (4)$$

Converting RGB to HSV (Hue, Saturation, Value) involves the following steps.

Let Red, Green, Blue (R,G,B) be the color coordinates with a value between 0 and 1. Let Max and Min be the largest and smallest of these values then the values of HSV are calculated as.

1) To change the range from 0....255 to 0...1 divide the values of R,G,B by 255 such that the values of R', G', B', SMAX, SMIN, θ , Hue, Saturation, Value are calculated as

$$\begin{aligned} R' &= R/255 \\ G' &= G/255 \\ B' &= B/255 \\ SMAX &= \text{MAX}(R', G', B') \\ SMIN &= \text{MIN}(R', G', B') \\ \theta &= SMAX - SMIN \end{aligned}$$

2) Calculate the H (Hue) value.

$$H = \begin{cases} 60^\circ \times \left(\frac{G' - B'}{\Delta} \% 6 \right) & , SMAX = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , SMAX = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , SMAX = B' \end{cases} \dots\dots\dots(5)$$

3) Calculate the S(Saturation) value.

$$s = \begin{cases} 0 & , SMAX = 0 \\ \frac{\Delta}{s_{MAX}} & , SMAX \neq 0 \end{cases} \dots\dots\dots(6)$$

4) Calculate the value.

$$V = SMAX \dots\dots\dots(7)$$

4. Feature Extraction and Classification of Image: Feature extraction and Classification are two basic parts of the CNN model [9,10] where several convolution layers are included in feature extraction followed by the models for activation function and max pooling. The classifier model includes fully connected layers.

3. Architecture of CNN and Sequential Models. The architecture of the CNN model [11] is as described below.

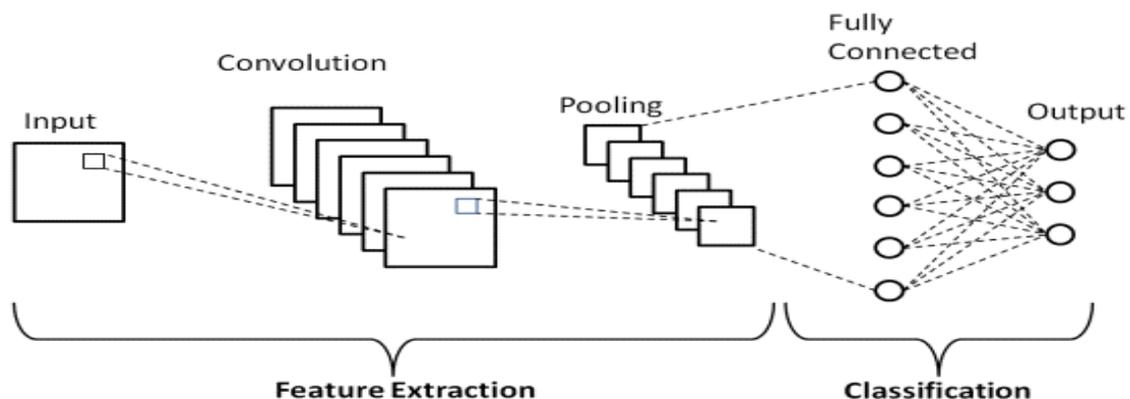


Figure 7 – Architecture of CNN model [11].

Convolution Layers

The convolutional, pooling, and fullyconnected (FC) layers are the three types of layers that are stacked to make up the CNN architecture apart from these layers other parameters like the dropout layer and the activation function are also included.

1. Convolutional Layer: In this layer, input is the images as this is the initial layer it is used to extract various characteristics from the input. Between the input image and a filter of size (MxM), The operation of convolutional neural is performed. The filter is slid over the input image secondly, the dot product is computed between the components of the input and the filter.

$$P_j^n = f \left(\sum_{i \in P_j} P_i^{n-1} * k_{ij}^n + Q_j^p \right) \dots\dots\dots (9)$$

The feature map is defined as output that gives information about the image like the corners and edges of the image. This output is later fed to the other respective layers to learn various other features of the input images.

2. Pooling Layer

Pooling Layer: This layer is used to reduce computational overhead by narrowing the size of the convolved feature map, which is achieved by reducing the interdependence between the CNN layers and operating independently on every feature map. Based on the technique used, there exist several types of Pooling operations.

$$s_i = \frac{\beta_i}{\sum_k \lambda R_j \alpha_k} \dots\dots\dots (10)$$

Where R_j denotes the pooling region j , F denotes feature map, and i is the element index inside region j . Stochastic St, is used in pooling operation for every feature map F , the stochastic (St) is expressed by

$$\lambda \beta_{xy}^{pk} = St(P, q, x, y) \varepsilon Q \left(\beta_{mn}^{Q-1} w(x, y) \right)_{xy}^{Pk} = st(m, n, x, y) \varepsilon P_3 \dots\dots\dots (11)$$

In the Max Pooling layer, it selects the maximum element which is taken from the feature map that is covered by the filter. The average of the elements in a predefined sized image section is calculated in Average Pooling. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

3. Fully Connected Layer: This layer comprises of the weights and biases along with the neurons and is used to connect the neurons of two different layers. They are the last few layers of the CNN architecture and placed before the output layer. This layer takes in flattened input images from previous layers which further undergoes few more FC layers thereafter, in which the operations of mathematical function occur. This step is immediately followed by the Classification process.

4. Dropout: When all the features are connected to the FC Layer problem of overfitting can occur. When the models perform well on the trained dataset such that it can affect the model's performance when used on a new dataset during which overfitting occurs. To overcome this limitation during the training process a few neurons are excluded from the neural network which indeed reduces the size of the overall model. With a dropout of 0.3, 30% of the nodes are randomly dropped from the network.

5. Activation Functions: It is used to learn, approximate continuous and complex relationship between variables of the network. These functions decide which information of the model should proceed in the forward direction and which ones should not at the end of the network. They add non linearity to the model. The commonly used activation functions such as the ReLU, SoftMax, tanH, and the Sigmoid functions.

DenseNet Model Architecture:

The DenseNet architecture [12] is all about modifying this standard CNN architecture like so:

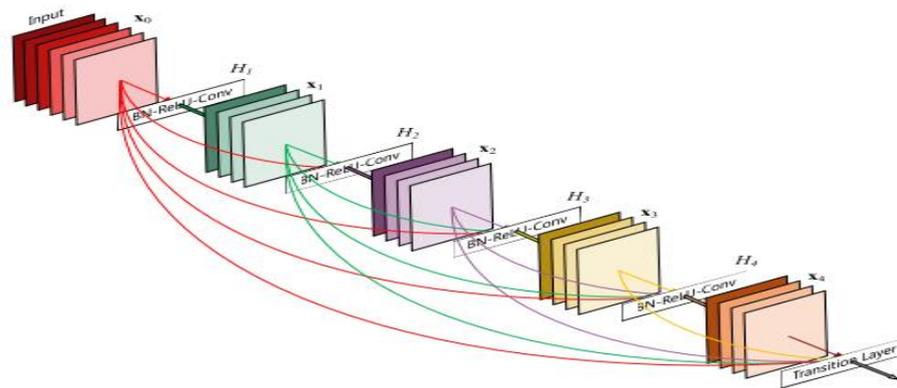


Figure 8 – DenseNet Model Architecture [12]

In a DenseNet architecture all the layers are interconnected, hence the name Densely Connected Convolutional Network. For L layers, there exist $L(L+1)/2$ direct connections and the feature maps of all the former layers are taken as inputs for the next layer. DenseNet essentially connects every layer to every other layer. This is the main idea that is extremely powerful. The input of a layer inside DenseNet is the concatenation of feature maps from previous layers.

4. Methodology Used in proposed work.

In this method, we take a sequential model of a lesser-known accuracy and optimize the enhancement of the model and make it easy to predict the disease at a much higher accuracy than previously established results, and finetune it for further detection. We use a CNN (Convolutional Neural Network) and the dataset is taken from the Kaggle website where around 8500 images are available for training and testing.

Pre-trained CNN models were used over a huge dataset taken from Kaggle. The batch size (mini) is taken as 64, the epochs were set to 50 with 73 iterations applied. The dataset was split into batches and the network coefficient was computed by applying gradient descent.

4.1 Performance measure.

The dataset is taken from Kaggle for the plant species of tomato, potato, and bell pepper leaf diseases.

Table of the sample sets.

Plant Species	Training Samples	Testing Samples
Tomato	3500	600
Potato	2152	645
Bell pepper	2500	750

4.2 Image pre-processing and labelling

In the feature extraction phase, to improve the accuracy of the models a set of processed and clean images are to be used to train and test the network classifier. A total of 8152 leaf sample images taken from the dataset, were pre-processed to dimensions of $256 \times 256 \times 32$ for the Sequential architecture model and $64 \times 64 \times 3$ for the DenseNet. These pre-processed images are then used to train the CNN. To amplify the accuracy of recognition parameters such as epoch, batch size, rate of learning, and optimizer techniques were modified.

4.3 Training of Sequential and DenseNet layers

1. Load the pre-trained network.
2. Perform a fresh recognition task by reconfiguring the last three layers.
3. Train the existing model with newly obtained data.

4. Evaluate the performance outcome.

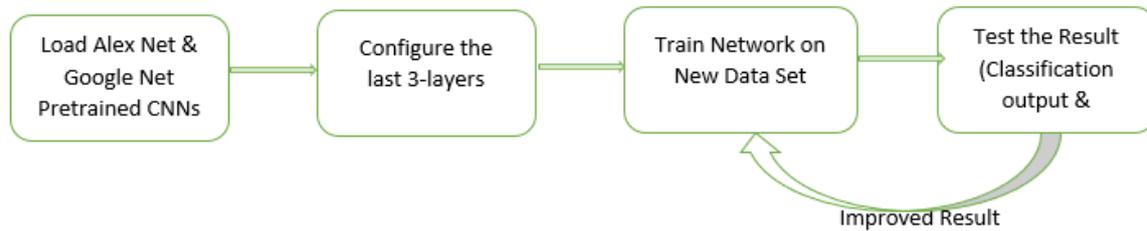


Figure 9 – Training of sequence and DenseNet layers

5. Experimental Results and Discussion

Performance Improvement Discussion

Techniques for accuracy performance improvement with data redesigning:

1. Increase image resolution (progressive resizing) from 128 x 128 x 3 to 256 x 256 x 3 or even higher size.
2. Random image rotations – change orientation of the image

Techniques for accuracy performance improvement with model optimization:

1. Fine tuning the model with subset data - Dropping few data samples for some of the overly sampled data classes.
2. Class weights - Used to train highly imbalanced (biased) database, class weights will give equal importance to all the classes during training

Techniques for accuracy performance improvement with hyper-parameter tuning:

1. Learning Rate (LR) optimization - Starting with a base LR and subsequently decreasing it for next epoch.
2. Batch Size - Usually try with max batch size your GPU can handle, depends on the memory of the GPU.
3. Increase model capacity - Increasing model depth (more number of layers) and width (number of filters in each convolution layer)

Sometime in real world scenario it's difficult to rely on single model. Different models can have different properties depending on number of layers it had, number of parameters it is trained on, etc. A particular model may not be able to predict best on particular set/type of images. In such cases we can combine multiple models to make a best prediction.

5.1 Sequential Model

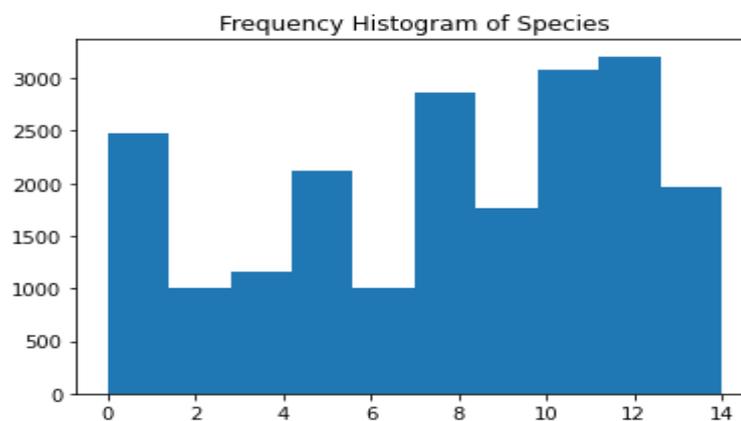


Figure 10 – Frequency histogram of species

Prediction using the Sequential Model:

Accuracy of the Sequential Model: 89%

Training and Validation Loss of Sequential Model:

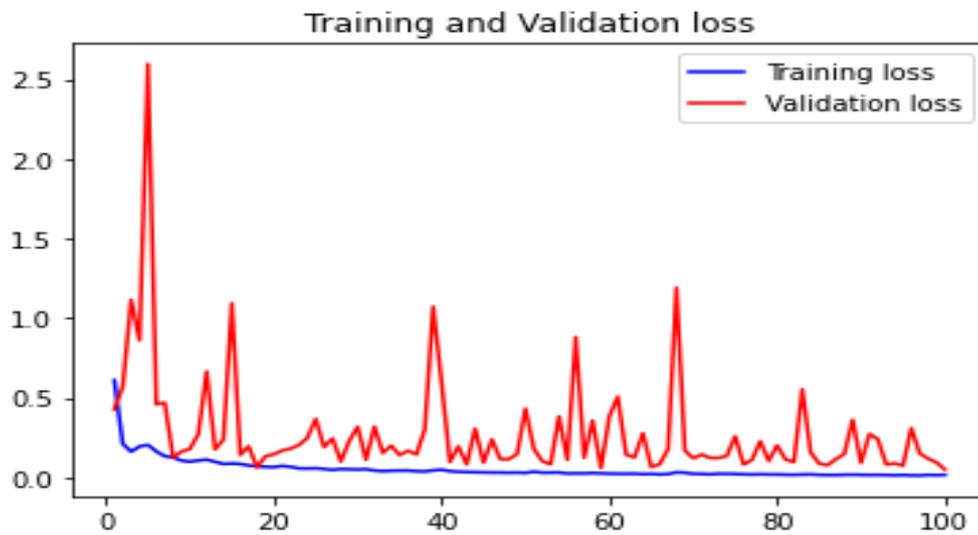


Figure 11-Training and Validation Loss

Training and Validation Accuracy

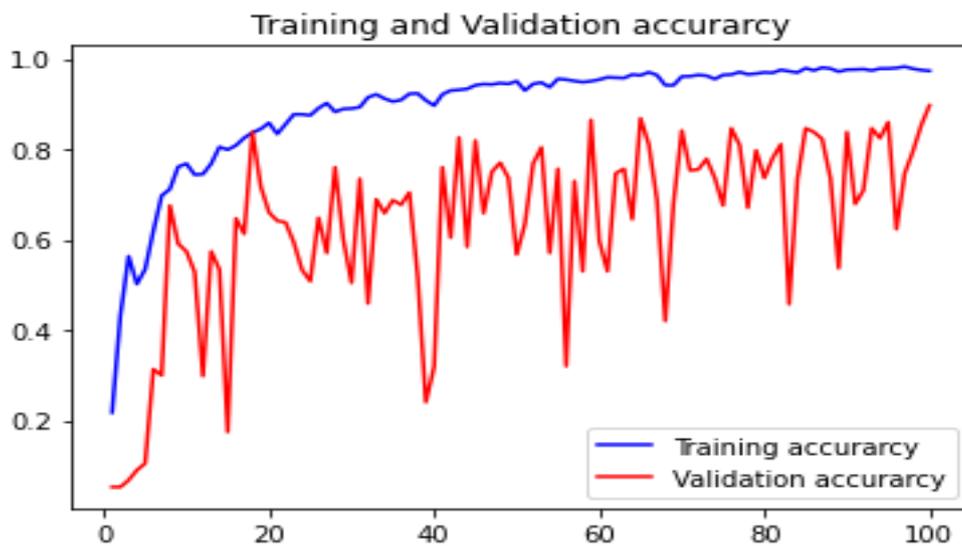


Figure 12 - Training and Validation Accuracy

Confusion Matrix of the Sequential Model :

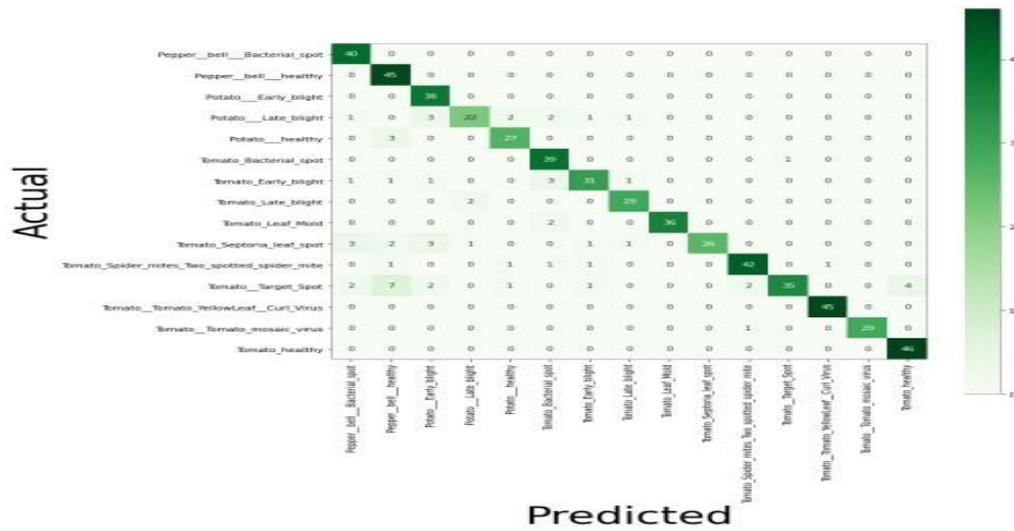


Figure 13 – Confusion Matrix of the Sequential Model

5.2DenseNet Model

Frequencies of species:

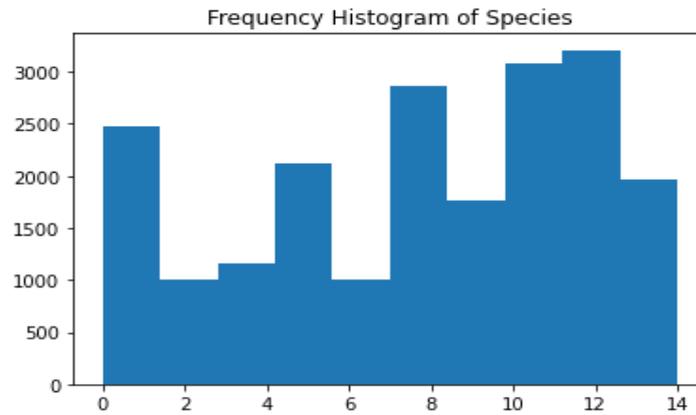


Figure 14 – Frequency histogram of species

Accuracy of the DenseNet Model: 98%

DenseNet Training and Validation Loss

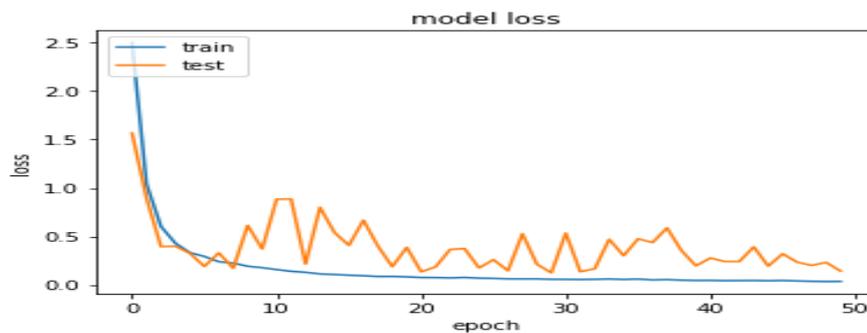


Figure 15– Training and validation loss of the DenseNet Model

DenseNet Training and Validation Accuracy

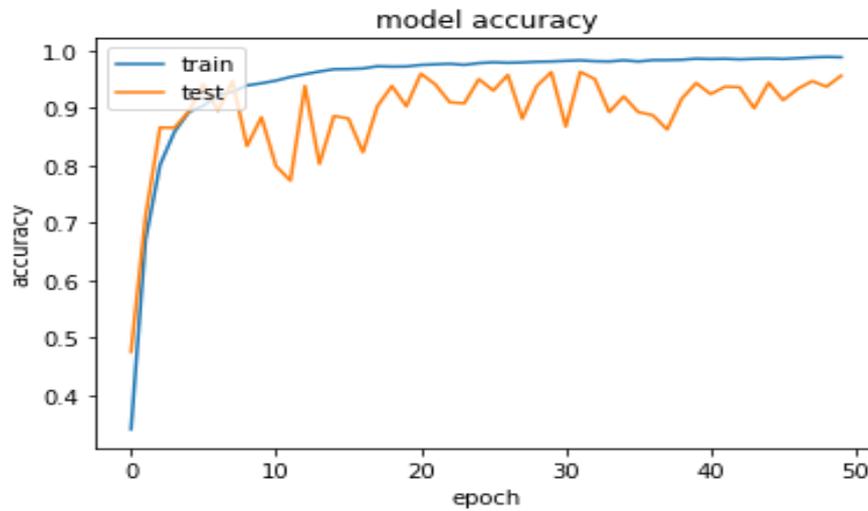


Figure 16 – Training and Validation Accuracy of the Sequential Model

Confusion Matrix for DenseNet Model:

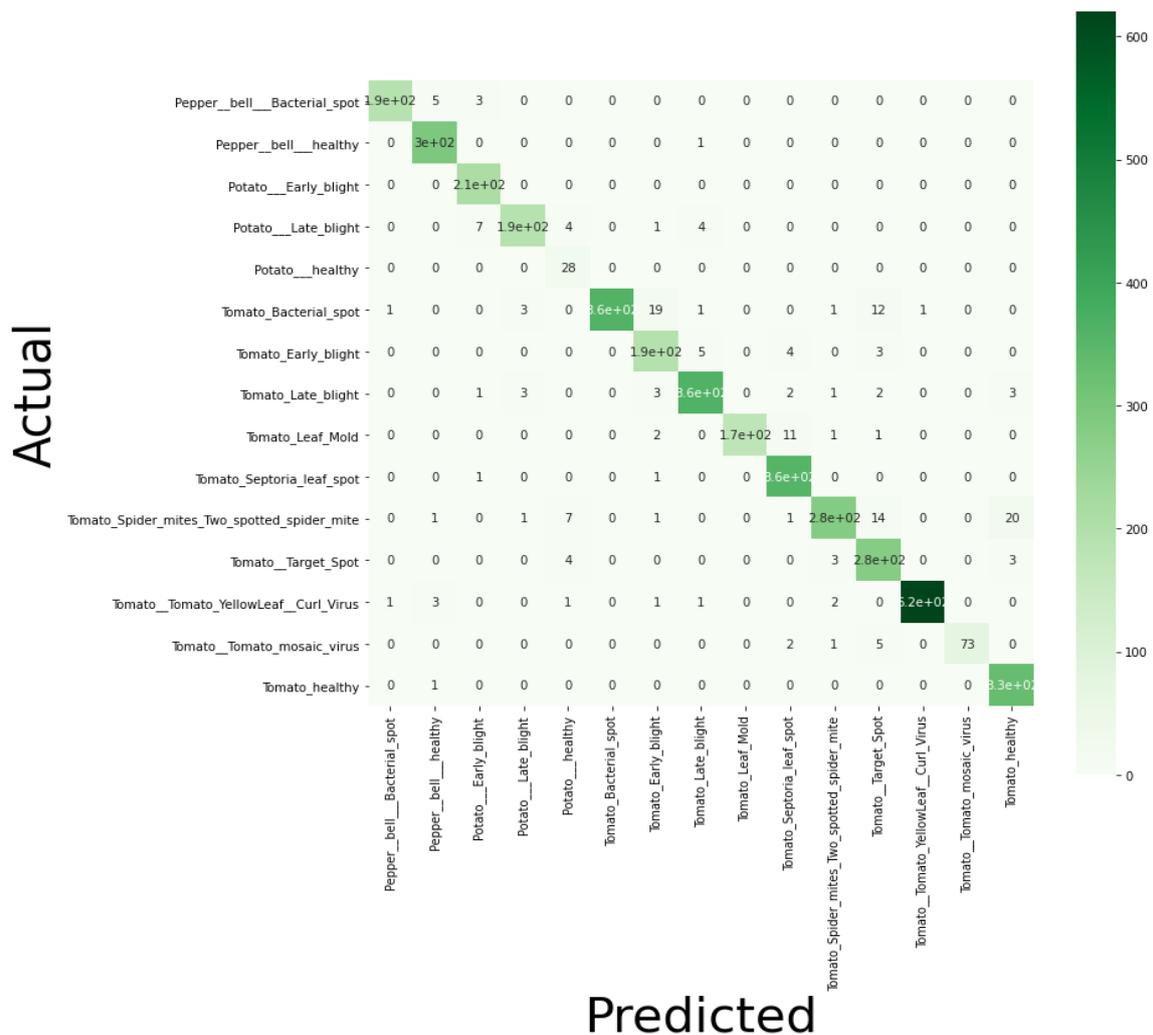


Figure 17 – Confusion matrix of the DenseNet Model.

6. Conclusion and Future scope

In this study, we propose a CNN model which utilizes sequential learning and DenseNet network to predict the disease of the plant and determine the accuracy of these models. Using pre-trained models, we fine-tune and increase the accuracy of the classification and optimize the performance of the models. The classification accuracies of those pre-trained models were 81% and 93%. Our proposed model correctly classifies the diseases and has an accuracy of 89% and 95% as shown in the results section. For the future, we have taken the work to develop another CNN model with a significant accuracy with better predictions and classifications and develop a simple Web-Based UI to predict the disease.

7. References:

- [1]. Balakrishna, K., & Rao, M. (2019). Tomato plant leaves disease classification using KNN and PNN. *International Journal of Computer Vision and Image Processing (IJCVIP)*, 9(1), 51-63.
- [2]. Hardikkumar S. Jayswal, Jitendra P. Chaudhari. Plant Leaf Disease Detection and Classification using Conventional Machine Learning and Deep Learning. *International Journal on Emerging Technologies* 11(3): 1094-1102(2020).
- [3]. Saradhambal, G., Dhivya, R., Latha, S., Rajesh, R., 2018. Plant disease detection and its solution using image classification. *International Journal of Pure and Applied Mathematics*, 119(14), pp.879-884.
- [4] Top 10 plant viruses in molecular plant pathology, <https://doi.org/10.1111/j.1364-3703.2011.00752.x>, OCT 2011.
- [5] Sarah D. Williams, Michael J. Boehm and Thomas K Mitchell, "Fungal and Fungal like plant diseases" *Agriculture and Natural resources*, Feb 2019.
- [6] Sachin B. Jadhav, Vishwanath R. Udupi, Sanjay B. Patil. "Convolutional neural networks for leaf image-based plant disease classification" *IAES International Journal of Artificial Intelligence*, Vol. 8, No. 4, December 2019, pp. 328-341.
- [7] Prajapati, H. B., Shah, J. P., & Dabhi, V. K. (2017). Detection and classification of rice plant diseases. *Intelligent Decision Technologies*, 11(3), 357- 373.
- [8] Saradhambal.G, Dhivya.R, Latha.S, R. Rajesh, PLANT DISEASE DETECTION AND ITS SOLUTION USING IMAGE CLASSIFICATION. *International Journal of Pure and Applied Mathematics*, Volume 119 No. 14 2018, 879-884.
- [9] **Emma Harte**, Plant Disease Detection using CNN, DOI:10.13140/RG.2.2.36485.99048, Sep 2020.
- [10] Adnan Mushtaq Ali Karol, Drushti Gulhane, Tejal Chandiwade, Plant Disease Detection using CNN & Remedy, *IJRAEEIE*, Vol. 8, Issue 3, March 2019.
- [11] The architecture of CNN, <https://www.upgrad.com/blog/basic-cnn-architecture/>
- [12] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, Densely Connected Convolutional Networks, *CVPR* 2017.
- [13] Muhammad E.h Chowdhury, Tawsifur Rahman, Amith Khandakar, Nabil Ibtehaz, Tomato Leaf diseases detection using Deep learning techniques. Open access, April 30, 2021.
- [14] Rubini, P.E., Kavitha, P, Deep learning model for early prediction of plant disease, *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, 2021, pp. 1104-1107.
- [15] Rubini, P.E., Kavitha. P, A comprehensive literature survey for deep learning approaches to agricultural applications, *World Review of Science, Technology and Sustainable Development*, 2021, 17(2-3), pp. 279-293.