Tomato Leaf Disease Detection

A BTP Report by

Anirudh Jakhotia

Khushi Pathak

Vula Naveen Kumar

Roll No's:

S20190010007

S20190010091

S20190010192

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
SRI CITY

12 - 12 - 2022

Final Report

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRI CITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "Tomato Leaf Disease Detection" in the partial fulfillment of the requirements for the award of the degree of B.Tech and submitted in the Indian Institute of Information Technology Sri City, is an authentic record of my own work carried out during the time period from February 2022 to November 2022 under the supervision of Prof. Dr. R. Shathanaa, Indian Institute of Information Technology Sri City, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date

STUDENT NAME)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Prof. Dr. R. Shathanaa)

Abstract

In general, a plant becomes diseased when it is continuously disturbed by some causal agent that results in an abnormal physiological process that disrupts the plant's normal structure, growth, function, or other activities. In nature, plants may be affected by more than one or an imbalance between soil moisture and oxygen is often more susceptible to infection by a pathogen, and a plant infected by one pathogen is often prone to a disease-causing agent at a time. A plant that must contend with a nutrient deficiency soon by secondary pathogens. The combination of all disease-causing agents that affect a plant makes up the disease complex.

In this report, we designed an efficient framework for image classification using deep learning. More specifically, the framework is based on two different deep-learning architectures.

We have taken a publicly available dataset from Kaggle for our model building. We have started with a basic neural network approach to classify fruits based on their quality. The accuracy we were able to achieve with this network is 97.96% on dataset-1.

Finally, we built a web application using our deep learning model. The web application now detects and classifies the tomato plant in its 10 categories of diseases and accurately predicts the correct one.

Contents

Introduction

No life is possible without plants; although plants are essential for life, they face several challenges to grow as a variety of diseases hit them. The need for rapid recognition and diagnosis of diseases helps reduce the chances of damage to the ecosystem. High moisture and high-temperature favor disease development. The bacteria fill the water-conducting tissue of plants with slime by multiplying rapidly inside it. This results in affecting the vascular system of plants.

Sick leaves may need to be pruned. Different spraying may be required for different types of diseases. Sick products and health products must be separated at the stage of collecting products.

Farmers identify the disease by examining the plant and making judgments based on their past experiences. This method does not provide accurate results as different farmers may have different experiences and the method lacks scientific rigor as well. There are chances that the farmers might miss classifying a disease and a wrong treatment may cause more damage to the plant.

Likewise, domain experts' visit to the field is costly. This necessitates the need for an automated image-based disease detection and classification mechanism that can replace the domain expert. CNN's are well known for image-based classification problems. A distinguishing feature of CNN is the use of a convolution layer, which omits the need for matrix multiplication. The various layers in a typical CNN include convolution, activation, pooling, and classification. The purpose of the convolution layer is to reduce the dimension of input.

Literature Survey

Table - 1 - Literature Survey

Author and Year	Title	Methodology	Evaluation Parameters	Drawbacks
Paymode, Ananda S., and Vandana B. Malode (2022) - ScienceDirect- ISSN 2589-7217	Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG [4]	1. Dataset (PlantVillage) 2. Image-processing using Picture filtering, grey transformation, picture sharpening, and scaling ..) 3.Image augmentation (flipping, cropping, rotation, color transformation) 4.Transfer learning VGG	Accuracy (TP+TN)/(TP+FP+FN+TN) Accuracy for grapes crop : 1. Proposed VGG16 : 98.40% 2. Inception-ResNet-V2: 81.11% Accuracy for tomato crop: 1. Proposed VGG16 : 95.71% 2. Inception-ResNet-V2: 86.10%	1. Preparation of genuine datasets and applying to the deep learning models with multiple crops leaves images more than two is not done. 2. The use of Inception V3 and ResNet-based CNN models for much deeper analysis of crop images is an anticipation which is not done.

Pantazi, Xanthoula Eirini, Dimitrios Moshou, and Alexandra Tamourido (2019) - Computers and electronics in agriculture 156 (2019): 96-104.	Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers [3]	1. Image-segmentation (Separation of foreground-background) 2. Derivation of Histogram 3. Classification into target class and outliers. 4. A classifier for each class. 5. Nearest Support Vector Strategy.	Accuracy Accuracies for each of the 3 diseases and healthy plant: - 93% for powdery mildew - 83.3% for downy mildew infection - 100% for the healthy and black rot infected leaf samples - Total success rate of 95%.	The presented paper is only capable of identifying the afore-mentioned health conditions in plant species other than the already tested and detecting conditions and classifying them as new categories.
Morbekar, Achyut, Ashi Parihar, and Rashmi Jadhav - (2020) International Conference for Emerging Technology (INCET) (pp. 1-5). IEEE	Crop disease detection using YOLO [5]	1. Dataset (PlantVillage) 2.Image augmentation-GAN (Generative Adversarial Networks) technique. 3.Object detection using Bounding Box Prediction and Class Prediction 4.Feature Extraction. 5. Training	mAP(Mean Average Precision) is the average of AP. Average Precision (AP) is finding the area under the precision-recall curve. Intersection over Union (IoU) : IOU : Area of Overlap/ Area of Union	1. The dataset is restricted to major crops cultivated in India, skipping other types of crops.

Burhan, S. A., Minhas, S., Tariq, A., & Hassan, M. N. (2020, June) - 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) (pp. 1-5). IEEE.	Comparative Study Of Deep Learning Algorithms For Disease And Pest Detection In Rice Crops [6]	1. Data augmentation 2. Cross Validation 3. Train-Test Split 4.Models (Vgg16,Vgg19,ResNet50, ResNet50V2 and ResNet101V2)	Accuracy (TP+TN)/(TP+FP+FN+TN) Average accuracy (Avg of Each class)/Total no. of classes Accuracy of different models on artificial data: - ResNet50 - 75.00% - ResNet50V2 - 73.36% - Vgg19 - 74.84%	1. In K-mean clustering and other image processing algorithms, the features of images have to be extracted manually which can be a huge limitation. 2. These models were trained on artificial and clean dataset that was too superficial to match data obtained from real rice crops.
Zhang, Yang, Chenglong Song, and Dongwen Zhang(2020) - IEEE Access, 8, 56607-56614.	Deep Learning-Based Object Detection Improvement for Tomato Disease [7]	1.Faster RCNN 2.K-means clustering 3.ROI Pooling and classification regression 4. Feature extraction network	Accuracy of different models: RCNN-res101 - 98.54% RCNN-mobile - 97.31%	The datasets available for tomato plants are small which leads us to use methods that produce synthetic samples that are too superficial to match data obtained from real tomato plants.

Kulkarni, O. (2018, August)- Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-4). IEEE.	Crop Disease Detection Using Deep Learning [8]	1. Pre-processing the dataset. 2.Training the Convolutional Neural Network (CNN) model to identify the type of crop. 3. Training CNN model to detect the disease. 4. Validation of model through obtained results	Accuracies in the detection of crop type MobileNet - 99.62% InceptionV3 - 99.74% Accuracies in detection of crop disease : MobileNet - 99.04% InceptionV3 - 99.45%	1. The proposed methodology was only tested on five classes of crops and three types of crop diseases for each class. 2. Classification of images that are not captured in a controlled environment and images that have multiple orientation are not considered.
Kirti, Rajpal, N. (2020, December) - 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 976-979). IEEE.	Black Rot Disease Detection in Grape plant (Vitis vinifera) Using color Based Segmentation and Machine Learning . [9]	1. Image Pre-Processing: (Read RGB Image, HSV Conversion) 2. Feature Extraction : K-means clustering to segment brown area from green area. 3. Percentage of Brown Region w.r.to green region. 4.The classification is applied using supervised machine learning algorithm (SVM)	1. Confusion Matrix 2. Color Percentage Ratio= Brown (Diseased) Pixels/ Green(Healthy) Pixels 3.The accuracy is computed with the kernels - 93.3% with Linear Kernel, - 94.1% with RBF Kernel - 93.9% with Polynomial Kernel.	1. Image processing algorithms, the features of images have to be extracted manually which can be a huge limitation.

Li, Lili, Shujuan Zhang, and Bin Wang- IEEE Access 9 (2021): 56683-56698.	Plant Disease Detection and Classification by Deep Learning—A Review [10]	1.CNN 2.Data Augmentation (GANS) 3.Transfer Learning 4. Few Shot Learning(FSL) 5. Hyper-spectral imaging(HSL) with DL models	1. Accuracy 2. Detection time 3.Precision 4.Single Image segmentation speed 5.Testing time 6.Average Accuracy 7. Accuracy of different models: SVM with CNN model - 96.8% AlexNet-precursor + Cascade Network - 97.62%	1.Large datasets are required for the training of CNN's but for plant disease recognition, such large datasets are not available. 2.Poor robustness of most of the DL models, they show good detection effects on their dataset but not so effective with other datasets.

Methodology

We started with a deep neural network for the classification of healthy and diseased plants. We have built an n-layered neural network We trained our 2-layered neural network model with a kaggle tomato plant disease dataset which consists of 10 classes.

CNN

CNN uses special convolution and pooling processes and performs parameter sharing. This makes CNN models work universally on any device, making them universally attractive. In this way, we will be able to use our model in automation systems.

CNN-1 Approach:

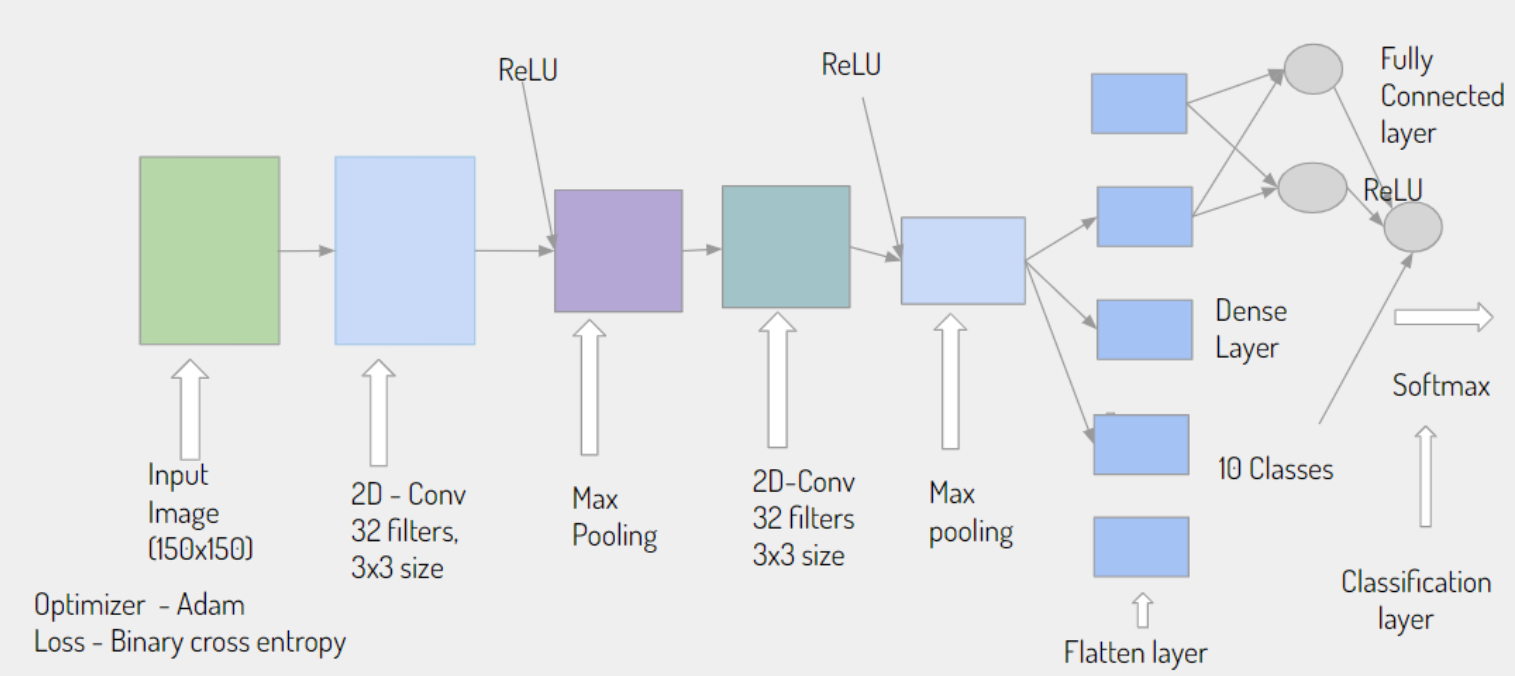

Figure - 1 - CNN-1 Model

We have then implemented a CNN model. The architecture consists of 2 layers. In each layer, there is a 2d-convolution layer followed by a max-pooling layer. In the end, we have used softmax-layer for classification.

Optimizer - Adam
Loss function - Binary cross entropy

CNN MODEL - 1 Summary

Layer1: 32 filters, filter size 3x3 , Max pooling, Activation relu, Layer2: 32 filters, filter size 3x3, Max pooling, Activation relu, Denser Layer 1: 128 nodes Denser Layer 2: 10 nodes Classification Layer: Softmax

CNN-2 Approach:

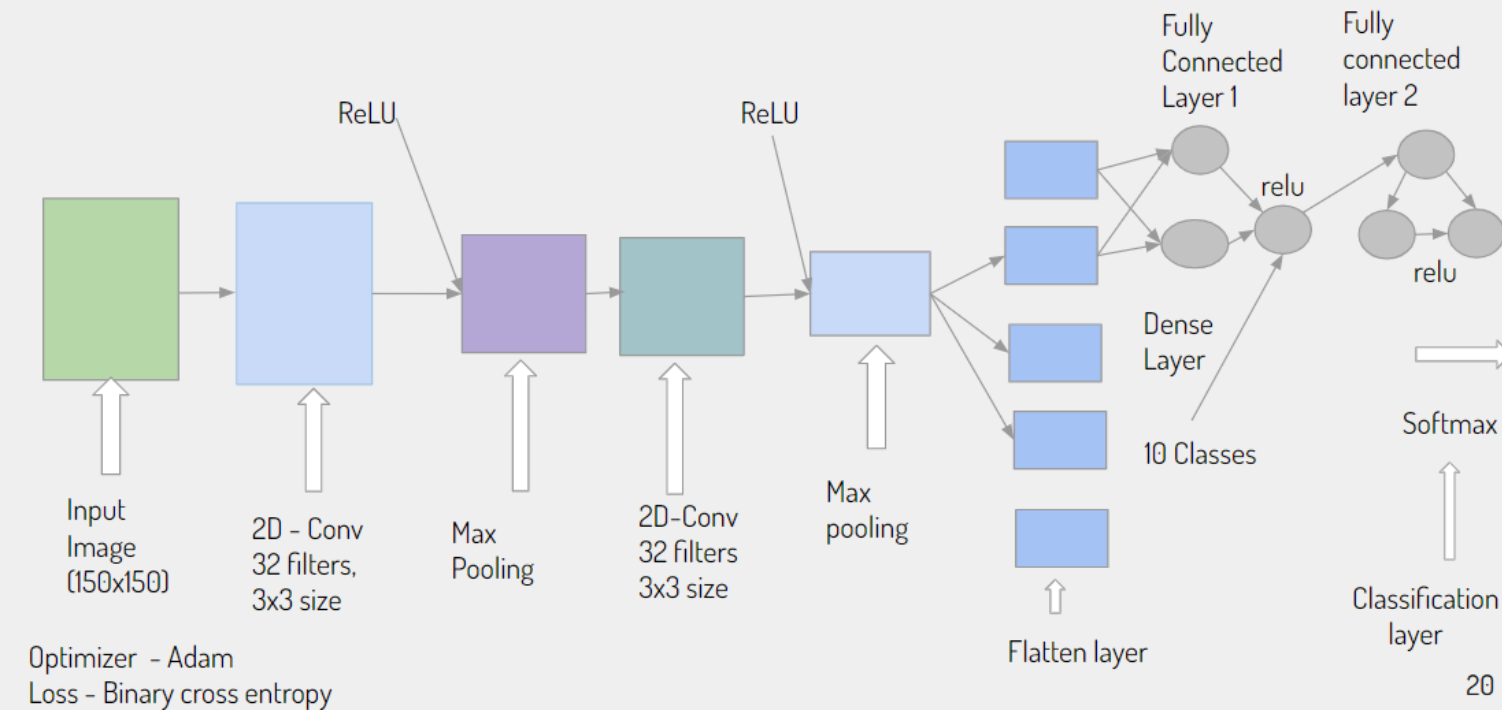

Figure - 2 - CNN-2 Model

After implementing CNN-1, we have observed there is a difference between train accuracy and validation accuracy. We have added a fully connected layer with batch size default. With this architecture, we are getting good results compared to the previous architecture.

Optimizer - Adam
Loss - Binary cross entropy
CNN MODEL - 2 Summary

Layer1: 32 filters, filter size 3x3, Max pooling, Activation relu, Layer2: 32 filters, filter size 3x3, Max pooling, Activation relu, Denser Layer 1: 128 nodes Denser Layer 2: 10 nodes Classification Layer: Softmax

CNN-3 Approach (Optimised):

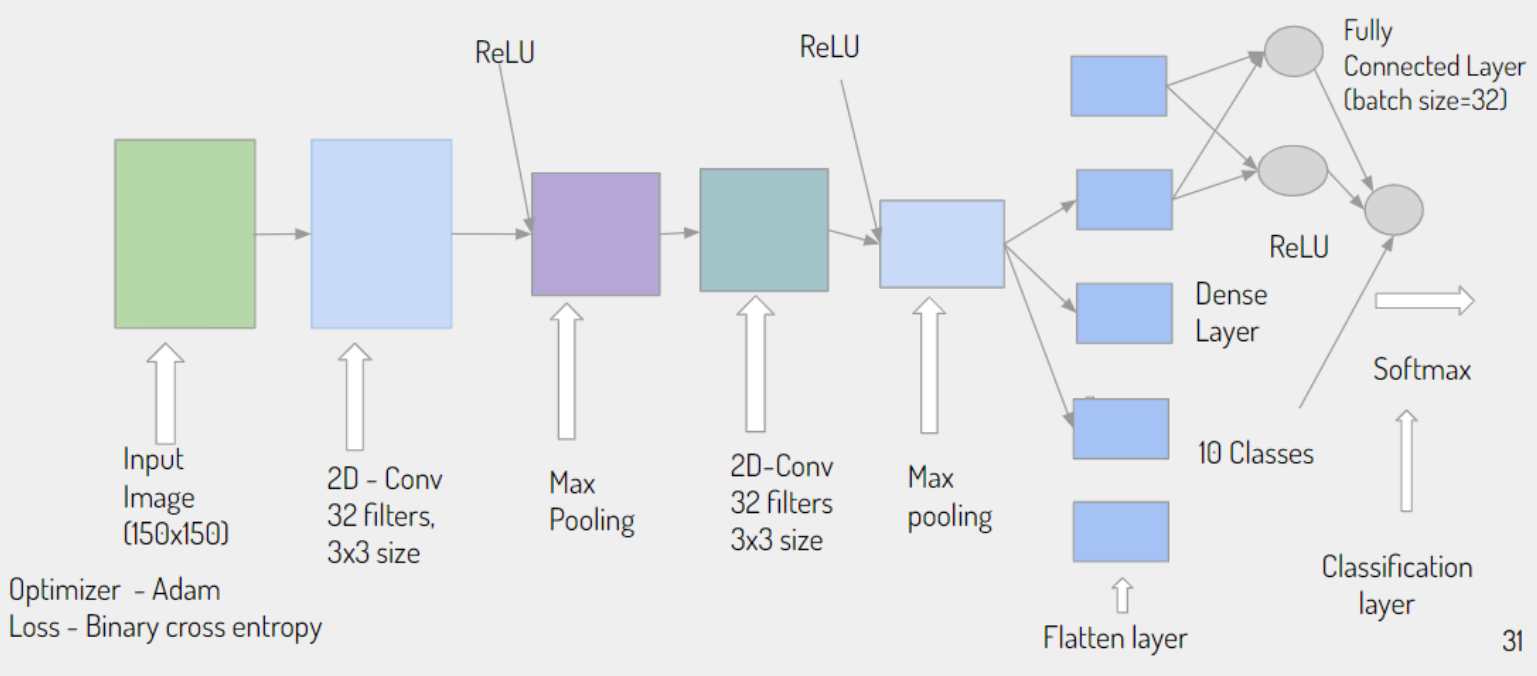

Figure - 3 - CNN-3 Model

After implementing CNN-2, we have again observed there is a difference between train accuracy and validation accuracy. We have removed the second fully connected layer which was added in CNN-2 and then made the existing fully-connected layer with batch size 64. We are now getting better results compared to the previous architecture.

Optimiser - Adam
Loss - Binary cross entropy
CNN MODEL - 3 Summary

Layer1: 32 filters, filter size 3x3 , Max pooling, Activation relu, Layer2: 32 filters, filter size 3x3, Max pooling, Activation relu, Denser Layer 1: 128 nodes Denser Layer 2: 10 nodes Classification Layer: Softmax

Inception-V3 Approach:

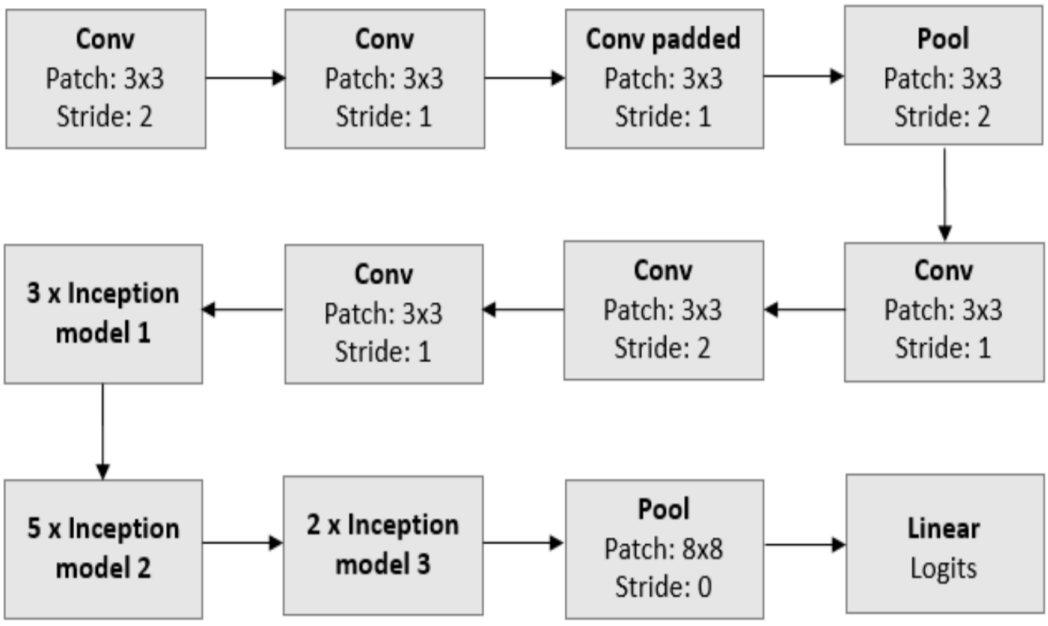

Figure - 4 - Inception-V3 Model - Credits - [2]

Inception v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. Just as ImageNet can be thought of as a database of classified visual objects, Inception helps the classification of objects in the world of computer vision.

Optimizer - Adam
Loss - Categorical cross entropy
Inception-V3 Summary

48 layered-architecture
Denser Layer: 10 nodes
Classification Layer: Softmax

VGG-16 Approach:

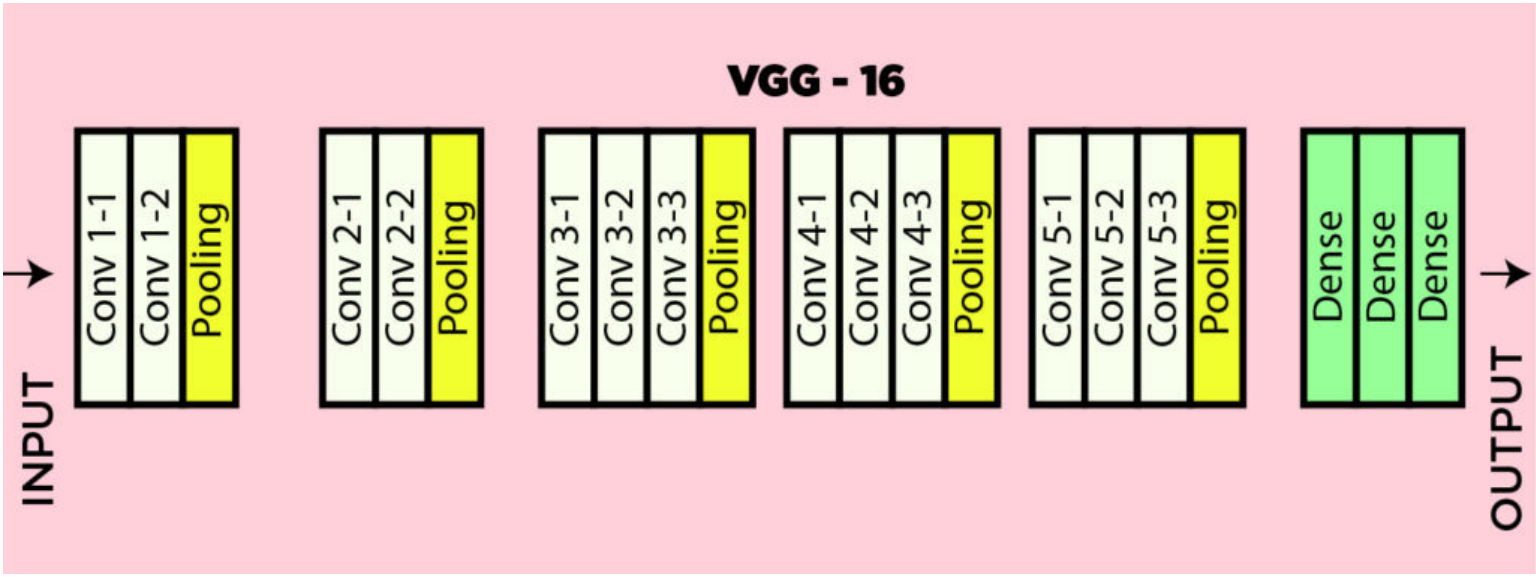

Figure - 5 - VGG-16 Model - Credits - [3]

The VGG16 architecture consists of twelve convolutional layers, some of which are followed by maximum pooling layers and then four fully connected layers, and finally a 1000 softmax classifier. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million parameters

Optimizer - Adam
Loss - Categorical cross entropy
VGG-16 Summary

16 layered-architecture
Denser Layer: 10 nodes
Classification Layer: Softmax

VGG-19 Approach:

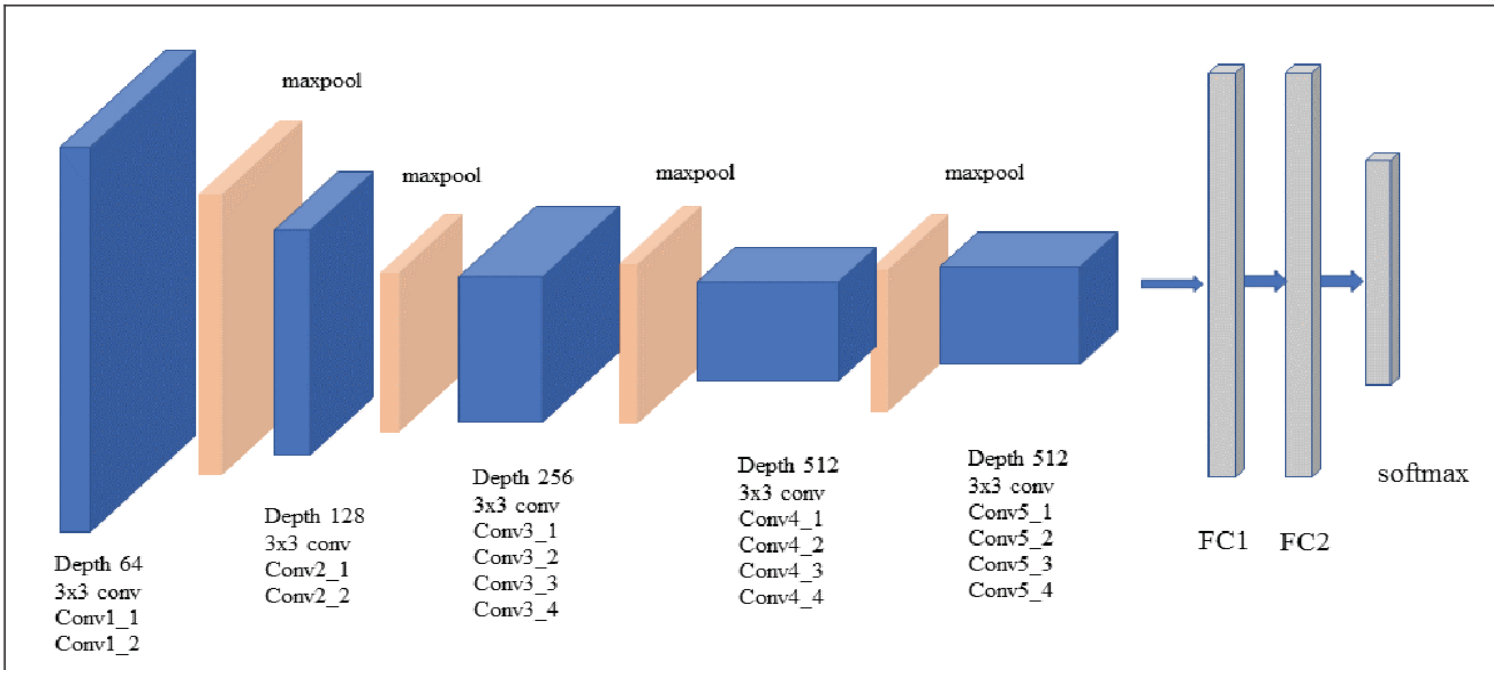

Figure - 6 - VGG-19 Model - Credits - [4]

VGG19 is a variant of the VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layers, 5 MaxPool layers, and 1 SoftMax layer). There are other variants of VGG like VGG11, VGG16, and others. VGG19 has 19.6 billion FLOPs. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

Optimizer - Adam
Loss - Categorical cross entropy
VGG-19 Summary

19 layered-architecture
Denser Layer: 10 nodes
Classification Layer: Softmax

Base Paper-1 [1] VGG Architecture:

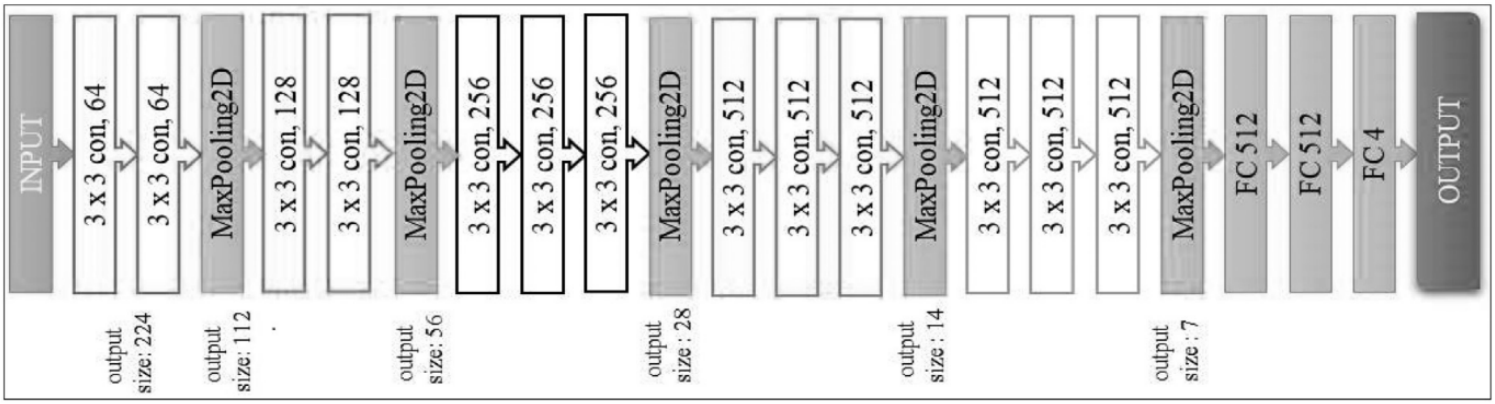

Figure - 7 - Base Paper-1 [1] VGG Architecture

This is a base paper-1 [1] architecture we have taken to compare our model. Using this architecture we have trained the model with our dataset.

Model Results:

CNN - 1:

Table - 2 - CNN-1 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -99.37 Testing Accuracy - 93.54	Training loss - 0.0057 Testing loss - 0.0738

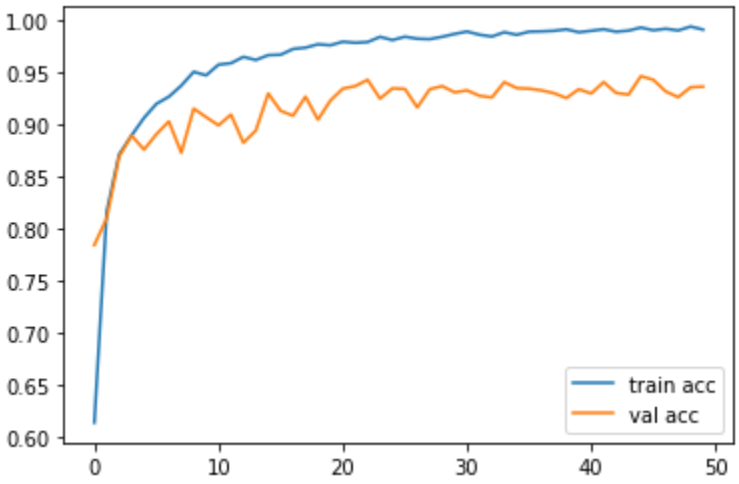

Figure - 8 - CNN - 1 - Training and Validation Accuracy

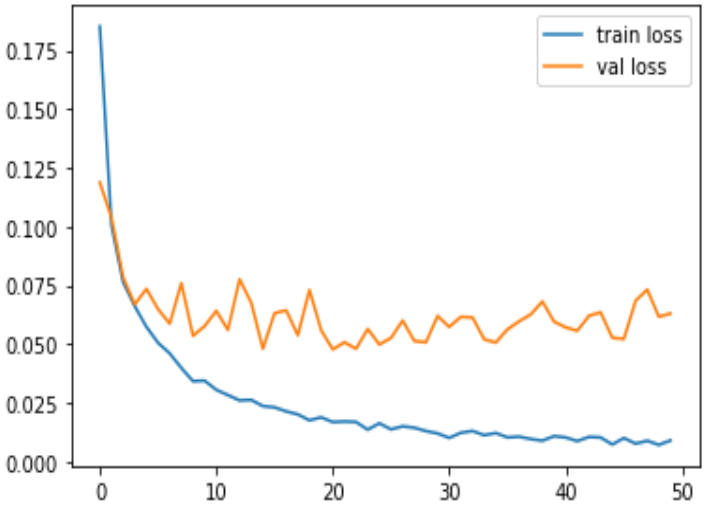

Figure - 9 - CNN - 1 - Training and validation Loss

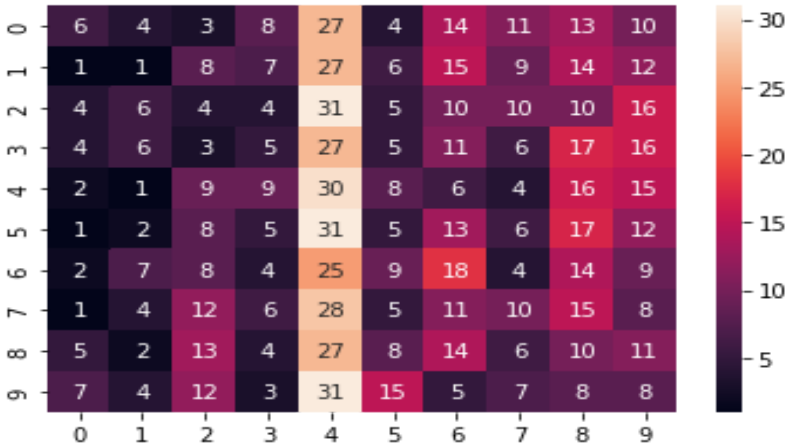

Figure - 10 - CNN - 1 - Confusion Matrix

This is the confusion matrix of the CNN-1 model. Diagonal elements represent correctly classified images and the rest of the elements in this matrix represent misclassified images.

CNN -2:

Table - 3 - CNN-2 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training accuracy - 98.58 Testing accuracy - 88.80	Training loss - 0.0126 Testing loss - 0.1029

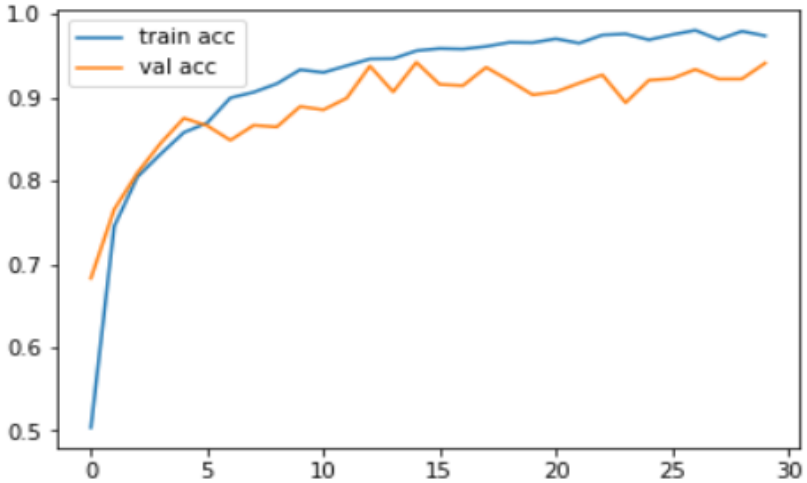

Figure - 11 - CNN - 2 - Training and Validation Accuracy

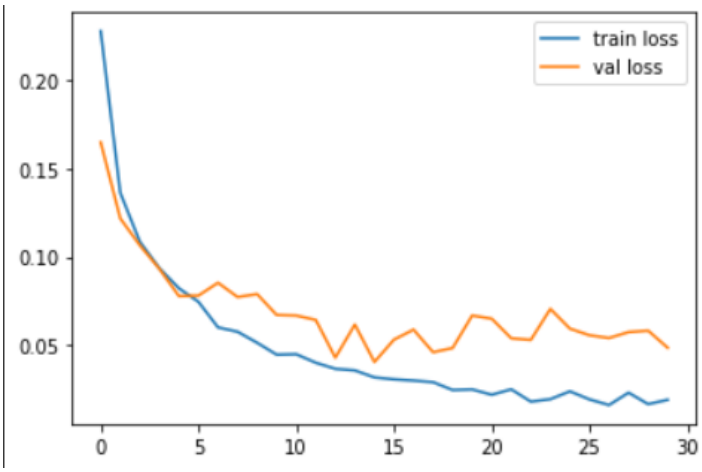

Figure - 12 - CNN - 2 - Training and Validation loss

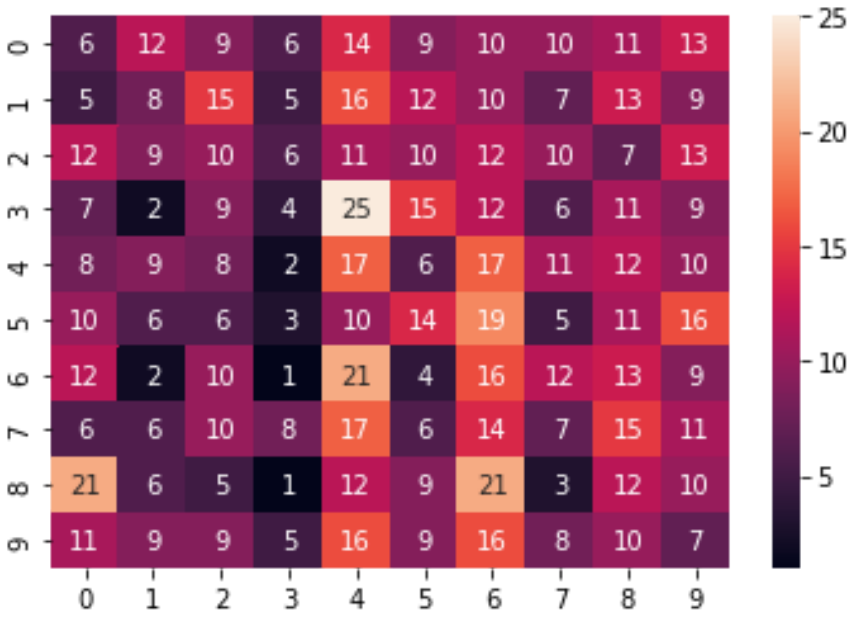

Figure - 13 - CNN - 2 - Confusion Matrix

The above graphs represents

1. Training vs Validation Accuracy of CNN 2 model
2. Training vs Validation Loss of CNN 2 model
3. Confusion Matrix of CNN 2 model

CNN -3:

Table - 4 - CNN-3 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -99.35 Testing Accuracy - 91.77	Training loss - 0.0059 Testing loss - 0.0859

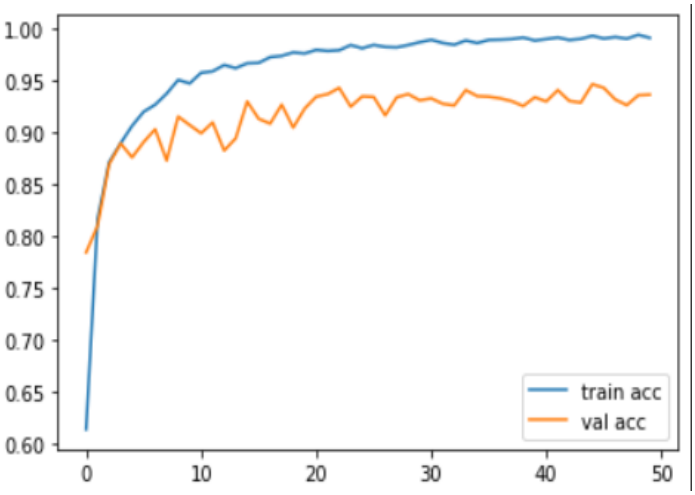

Figure - 14 - CNN - 3 - Training and Validation Accuracy

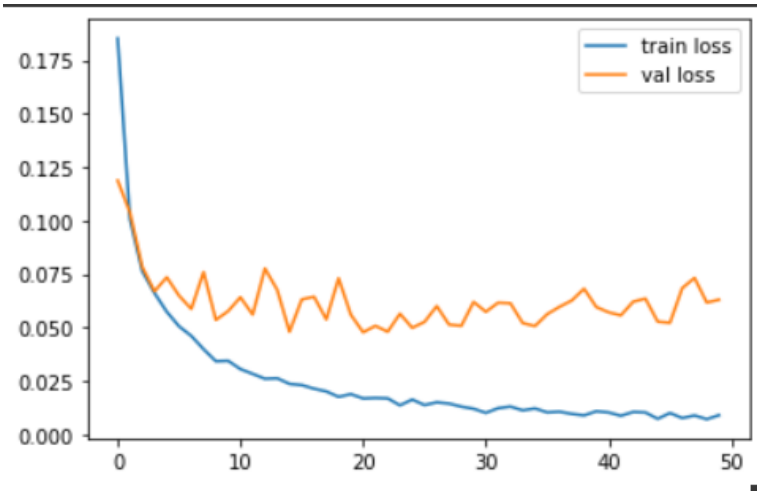

Figure - 15 - CNN - 3 - Training and Validation Loss

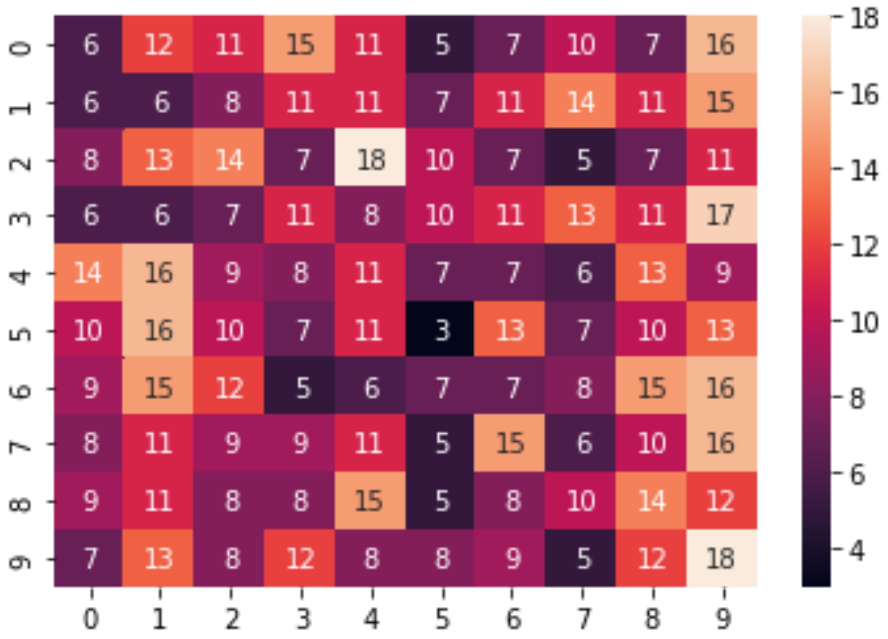

Figure - 16 - CNN - 3 - Confusion Matrix

The above graphs represents

4. Training vs Validation Accuracy of CNN-3 model
5. Training vs Validation Loss of CNN-3 model
6. Confusion Matrix of CNN-3 model

INCEPTION-V3:

Table - 5 - INCEPTION -V3 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -99.37 Testing Accuracy - 93.54	Training loss - 0.0057 Testing loss - 0.0738

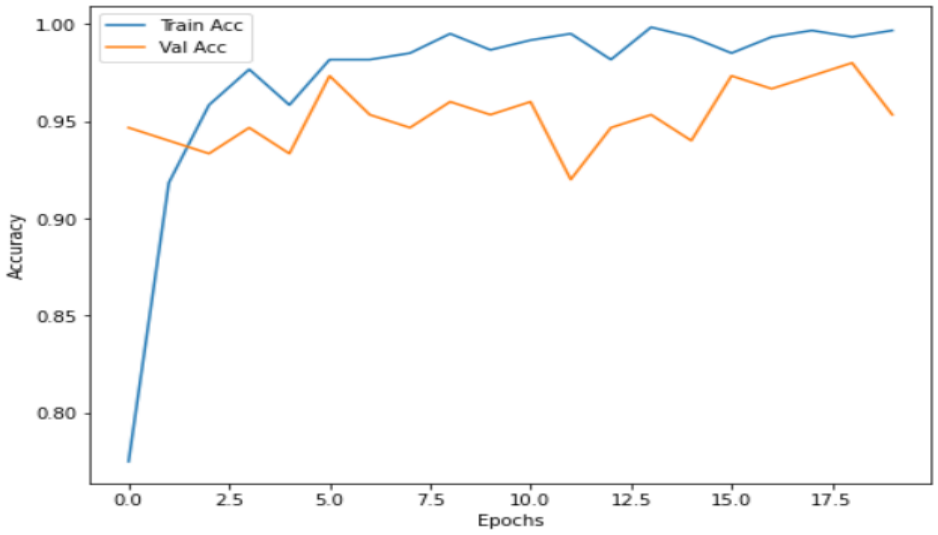

Figure - 17 - Inception-V3 - Training and Validation Accuracy

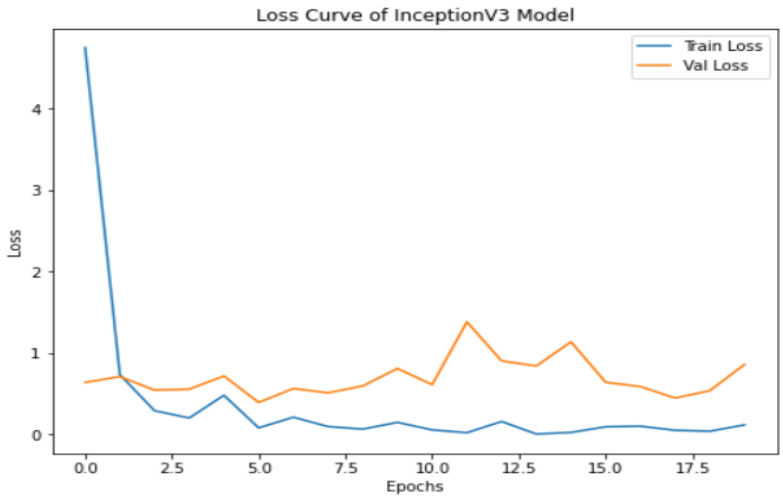

Figure - 18 -Inception-V3 - Training and Validation Loss

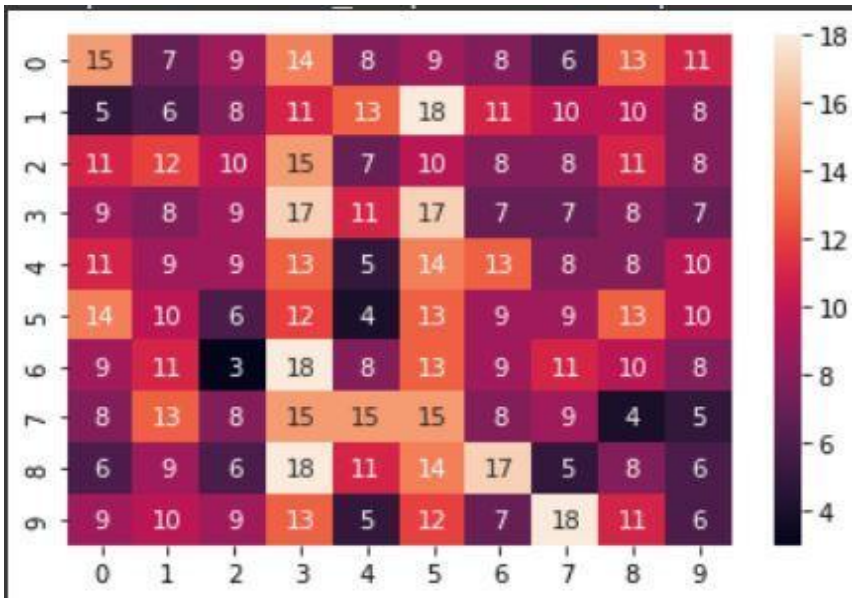

Figure - 19 - Inception-V3 - Confusion Matrix

The above graphs represents

7. Training vs Validation Accuracy of Inception-V3 model
8. Training vs Validation Loss of Inception-V3 model
9. Confusion Matrix of Inception-V3 model

VGG-16:

Table - 6 - VGG-16 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -99.37 Testing Accuracy - 94.17	Training loss - 0.0020 Testing loss - 0.2110

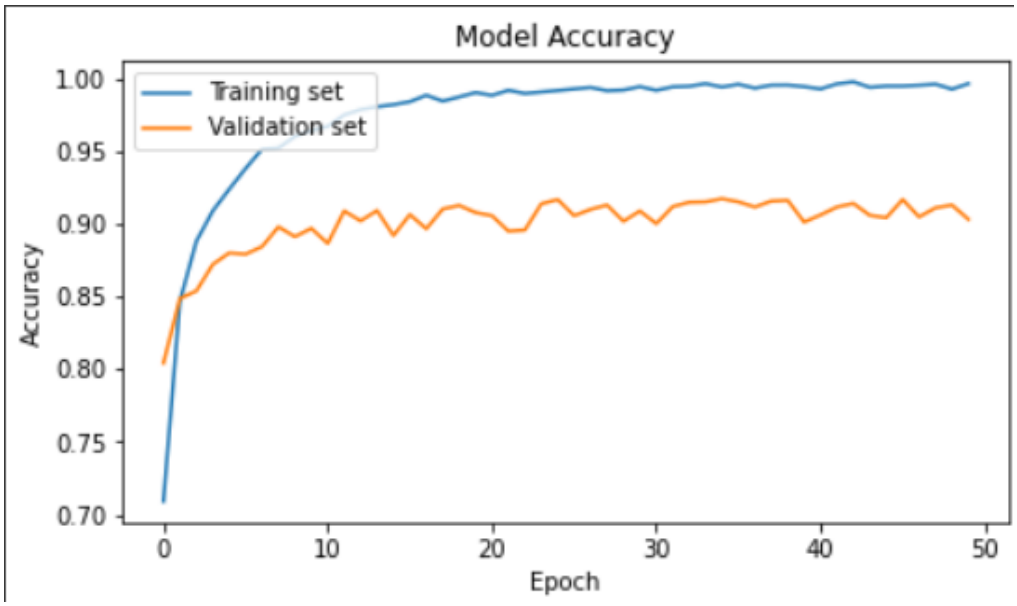

Figure - 20 - VGG - 16 - Training and Validation Accuracy

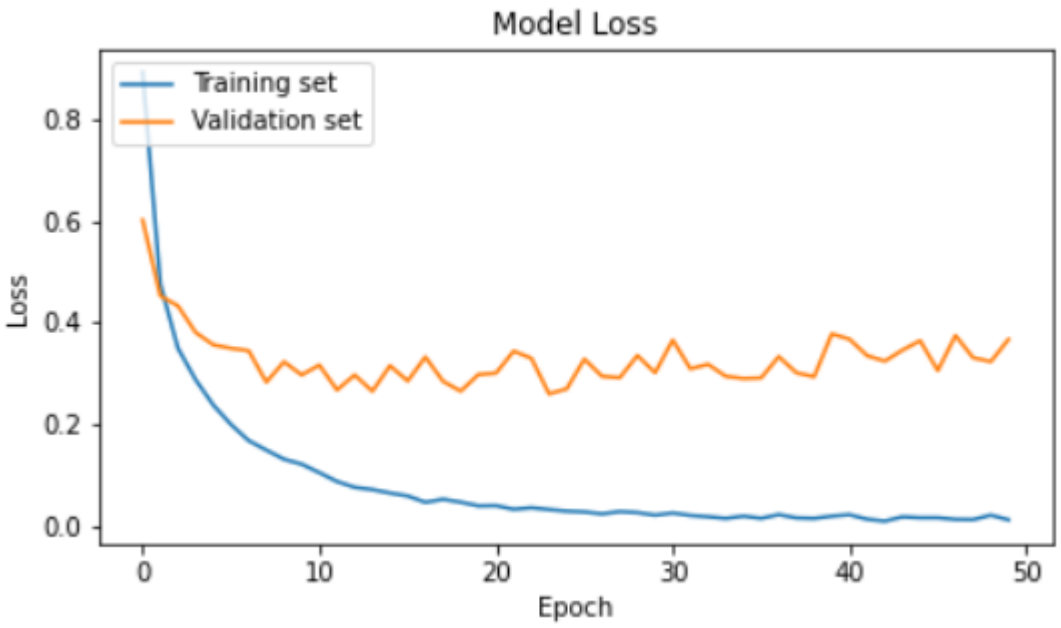

Figure - 21 - VGG - 16 - Training and Validation Loss

Figure - 22 - VGG - 16 - Confusion Matrix

The above graphs represent

10. Training vs Validation Accuracy of VGG-16 model
11. Training vs Validation Loss of VGG-16 model
12. Confusion Matrix of VGG-16 model

VGG-16 - Fine Tuning:

Table - 7 - VGG-16-Fine Tuning Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -100 Testing Accuracy - 98.82	Training loss - 0.0001 Testing loss - 0.0483

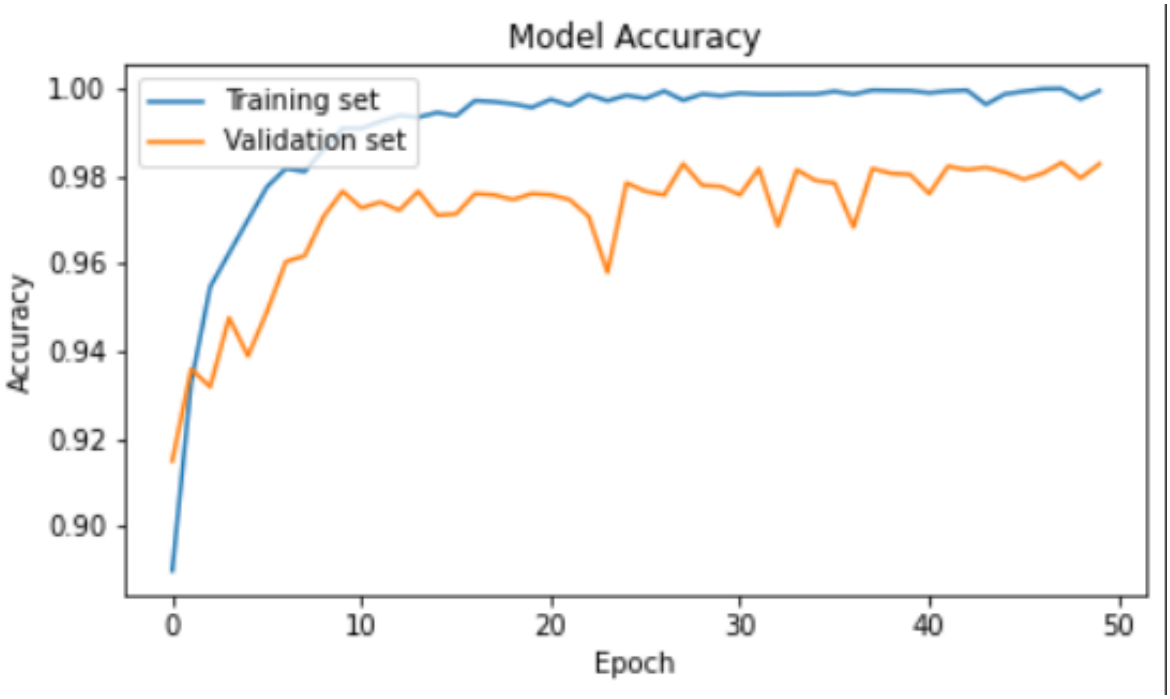

Figure - 23 - VGG-16-Fine Tuning - Training and Validation Accuracy

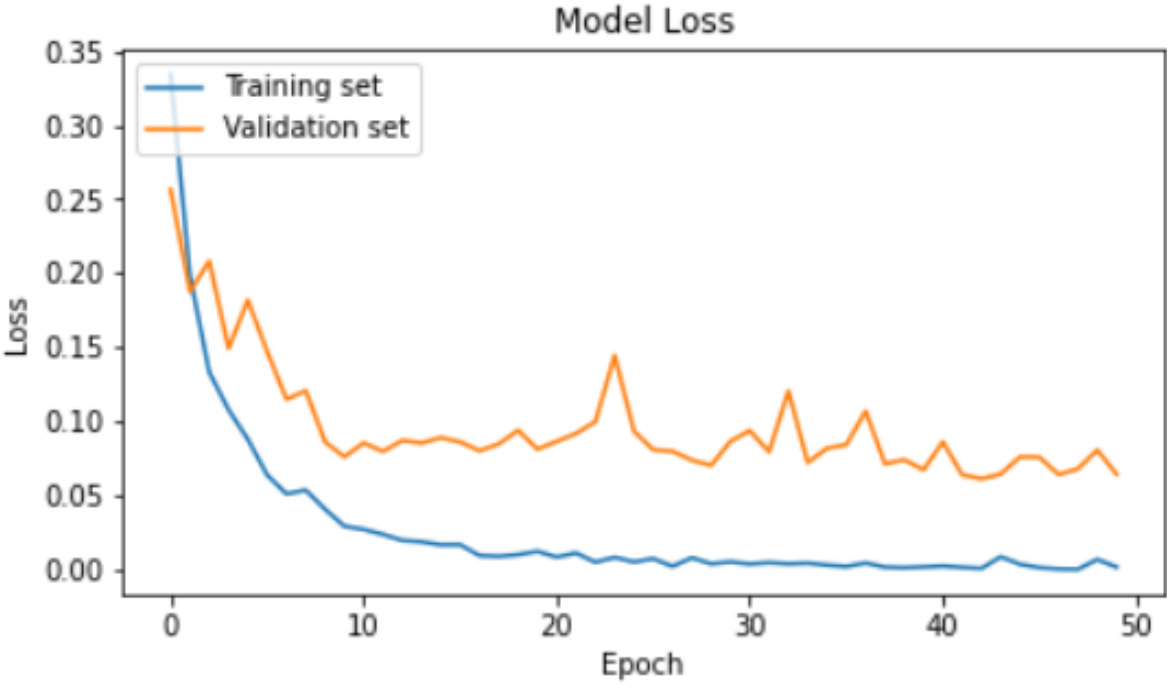

Figure - 24 - VGG-16-Fine Tuning Training and Validation Loss

The above graphs represent

13. Training vs Validation Accuracy of VGG-16 Fine-tuned model
14. Training vs Validation Loss of VGG-16 Fine-tuned model

VGG-19:

Table - 8 - VGG-19 Results

Dataset Name	Total Training Images	Total Testing Images
Tomato leaf disease detection (Total 18,345)	14,678 (80%)	4,585

Dataset Name	Accuracy	Loss
Tomato leaf disease detection	Training Accuracy -98.79 Testing Accuracy - 88.37	Training loss - 0.0372 Testing loss - 0.04795

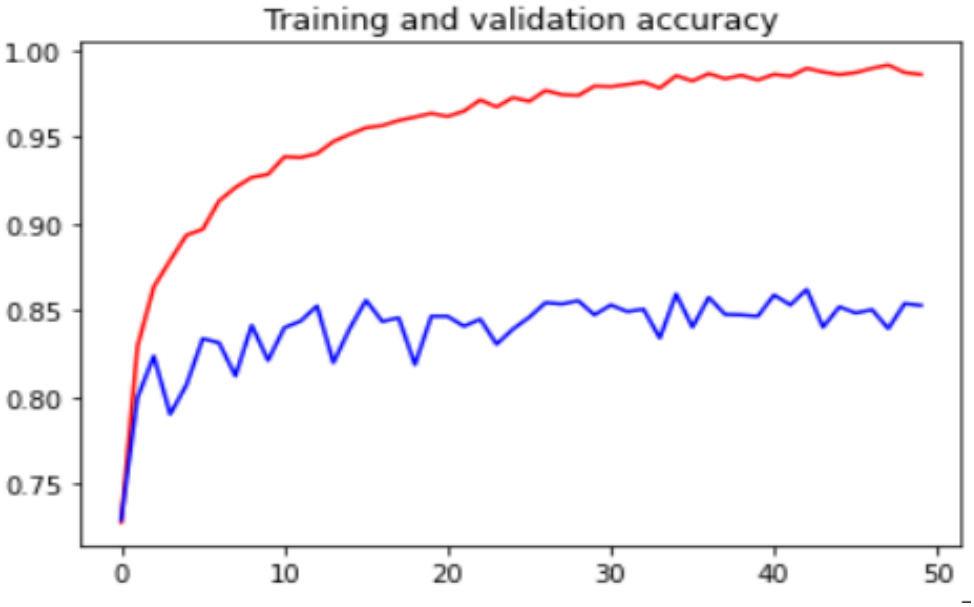

Figure - 25 - VGG-19 - Training and Validation Accuracy

Figure - 26 - VGG-19 - Training and Validation Loss

The above graphs represents

1. Training vs Validation Accuracy of VGG19 model
2. Training vs Validation Loss of VGG19 model

Base Paper - I [1]:

Table - 9 - Base Paper- I [1] Results

Dataset Name	Total Training Images	Total Testing Images
Plant Village Dataset (Total 16,000)	12,800 (80%)	3,200

Dataset Name	Accuracy	Loss
Plant Village Dataset	Training Accuracy - 95.71 Testing Accuracy - 94.32	Training Loss - 0.1643 Testing Loss - 0.2627

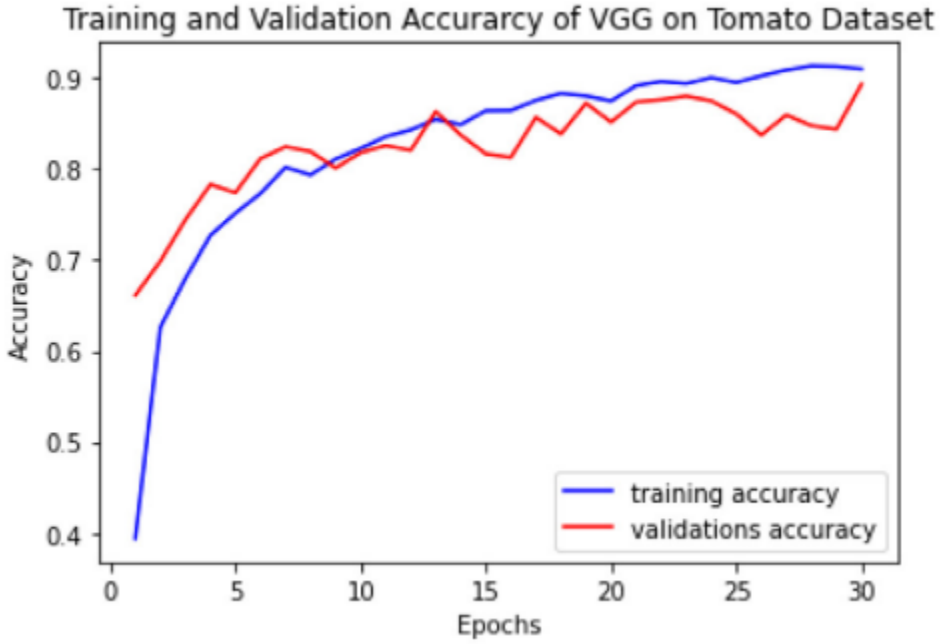

Figure - 27 - BasePaper [1] - Training and validation Accuracy

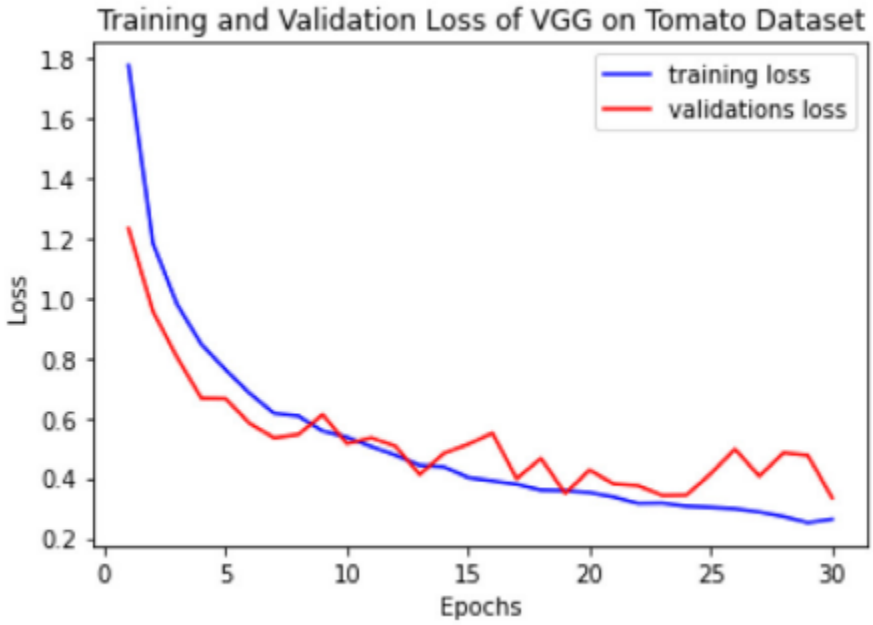

Figure - 28 - BasePaper [1] - Training and validation Loss

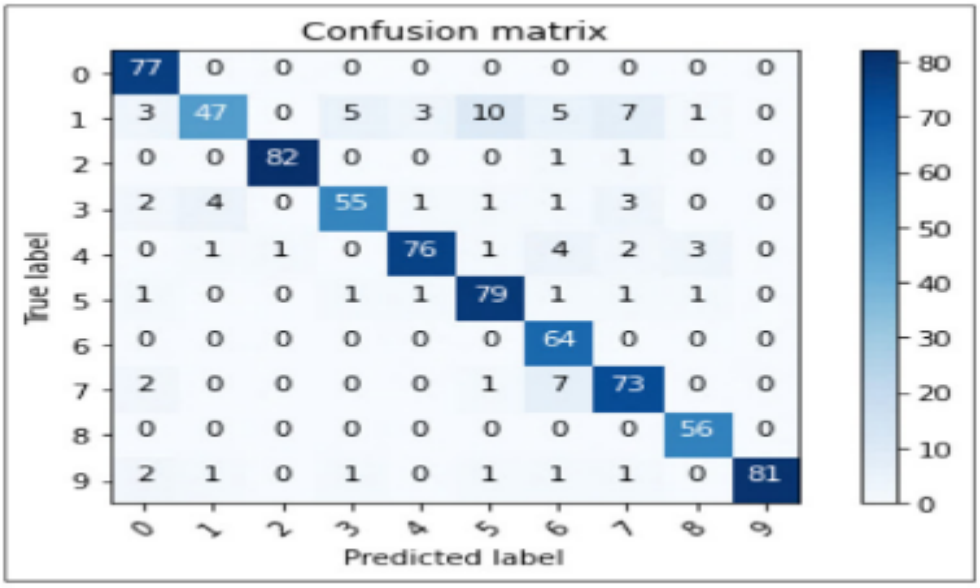

Figure - 29 - BasePaper [1] - Confusion Matrix

The above graphs represent

3. Training vs Validation Accuracy of BasePaper [1] model
4. Training vs Validation Loss of BasePaper [1] model
5. Confusion Matrix of BasePaper [1] model

Model Comparisons

1) Our VGG16 vs Base Paper -1 [1] - VGG16 - Accuracy:

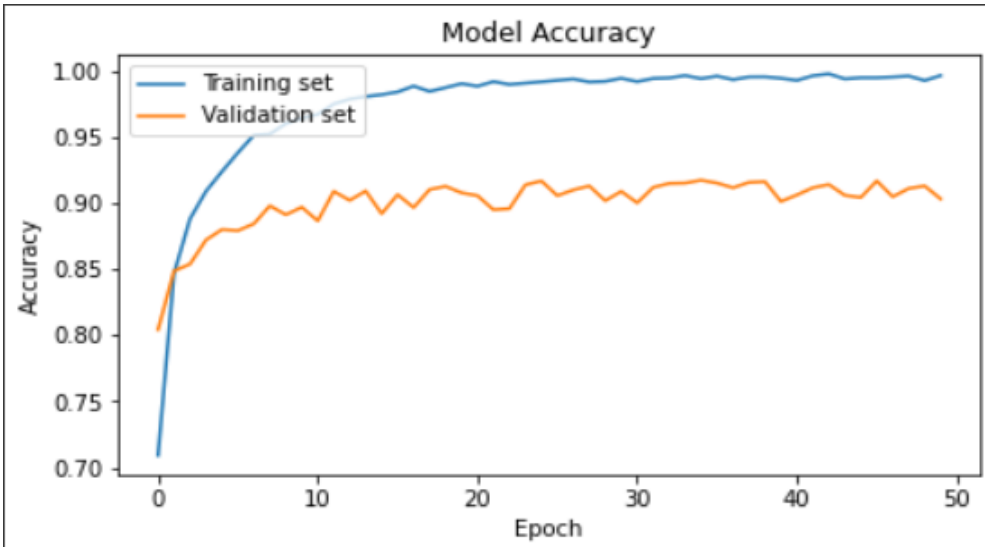

Figure - 30 - Our VGG16 - Accuracy

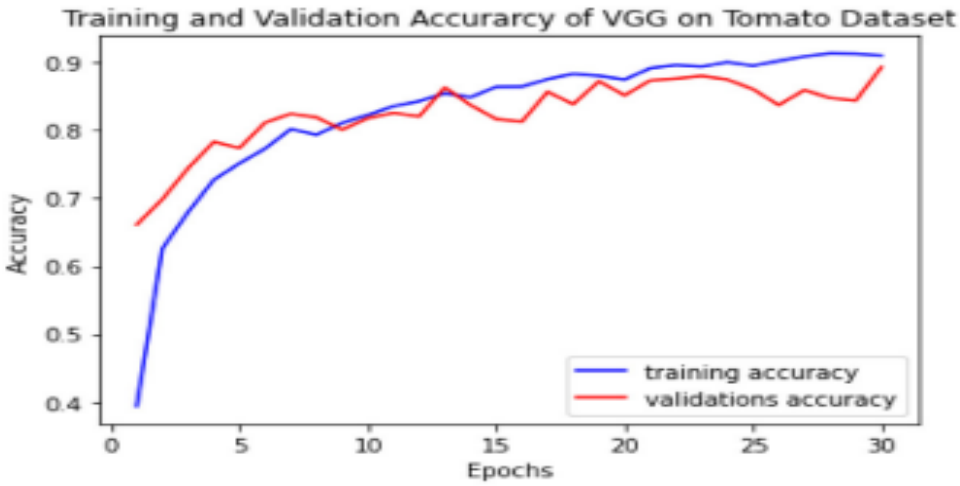

Figure - 31 - Base Paper- 1 [1] - VGG16 - Accuracy

2) Our VGG16 vs Base Paper -1 [1] - VGG16 - Loss:

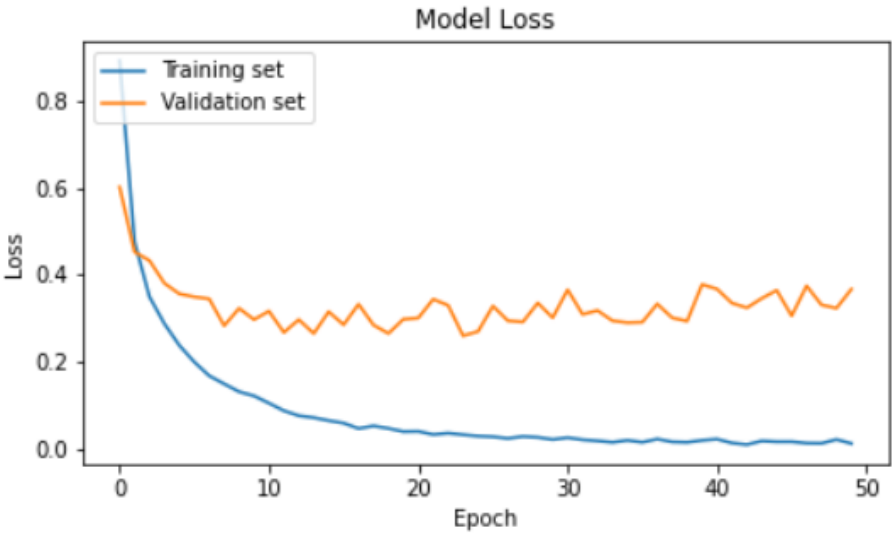

Figure - 32 - Our VGG16 - Loss

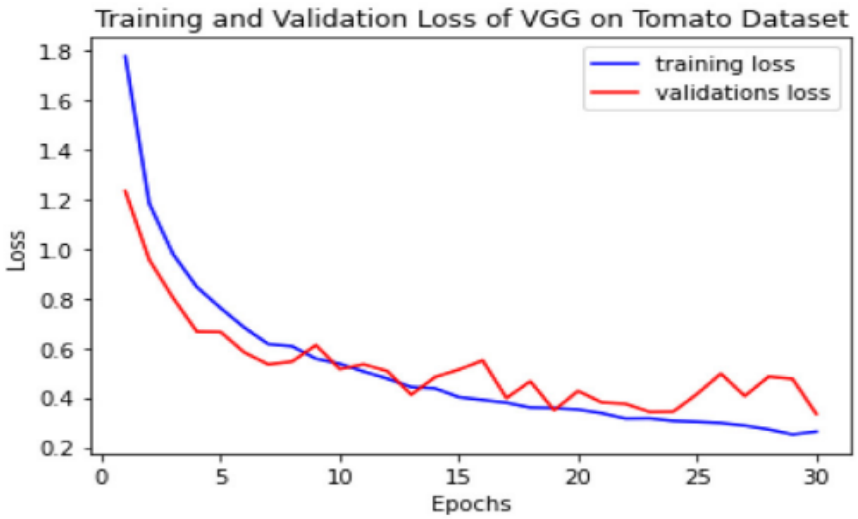

Figure - 33 - Base Paper- 1 [1] - VGG16 - Loss

3) Our Initial VGG16 vs VGG16 - Fine -Tuning Accuracy:

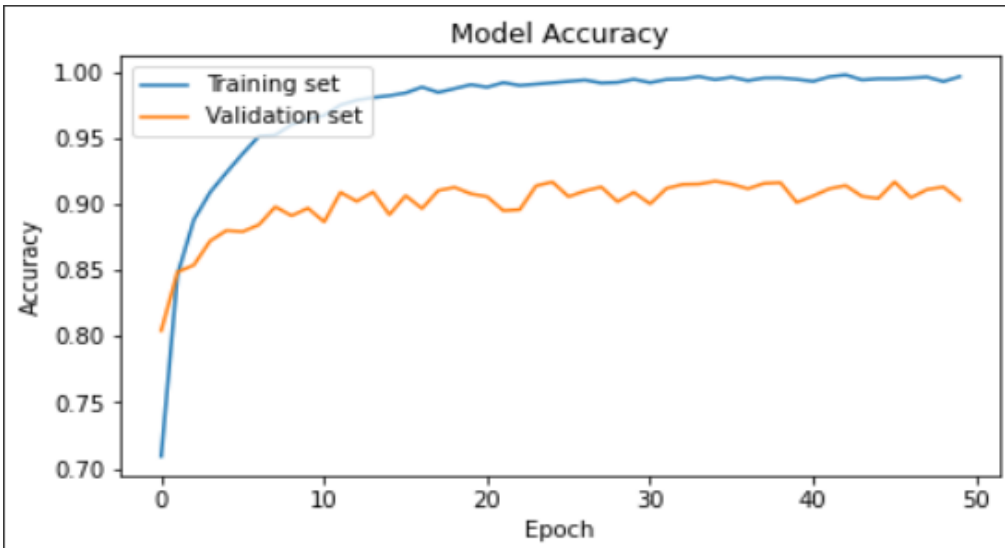

Figure - 34 - Our VGG16 - Accuracy

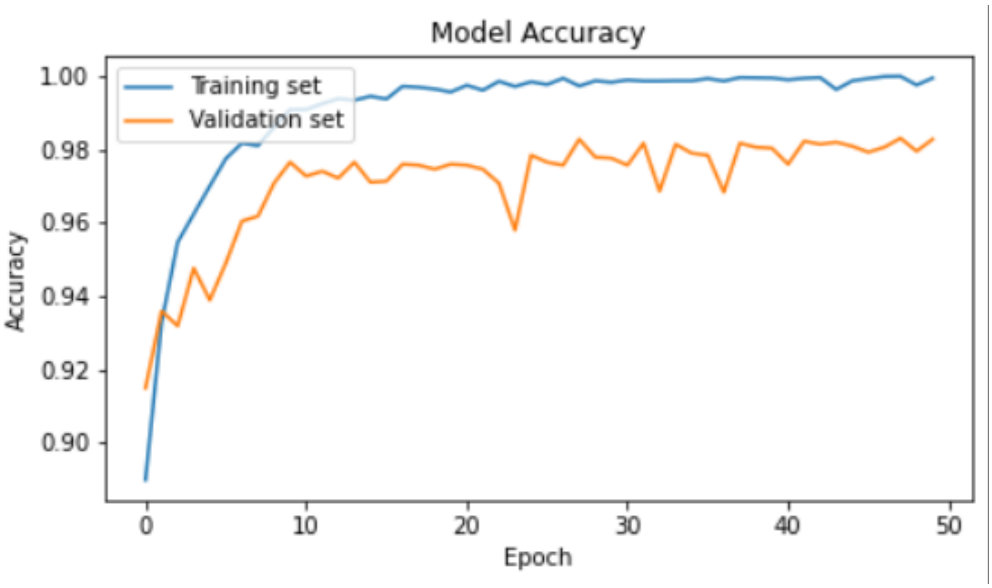

Figure - 35 - VGG16 - Fine -Tuning - Accuracy

4) Our Initial VGG16 vs VGG16 - Fine -Tuning Loss:

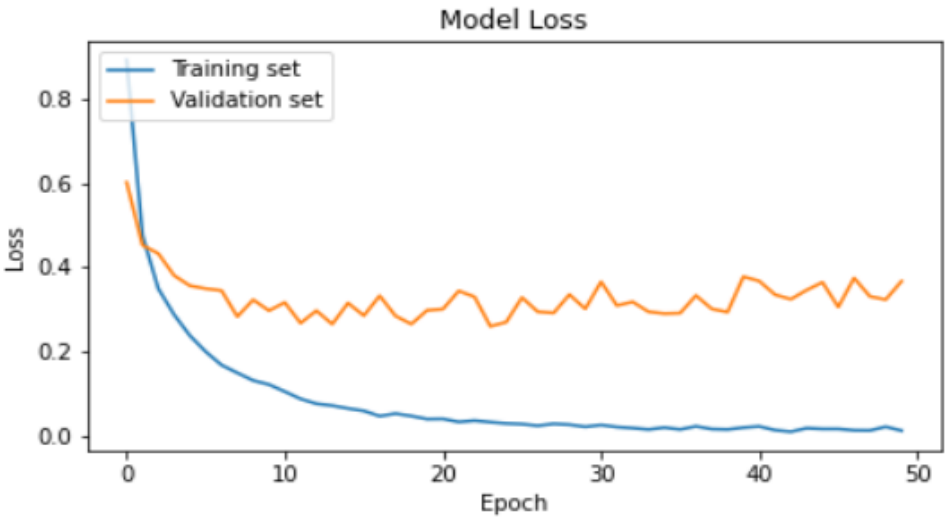

Figure - 36 - Our VGG16 - Loss

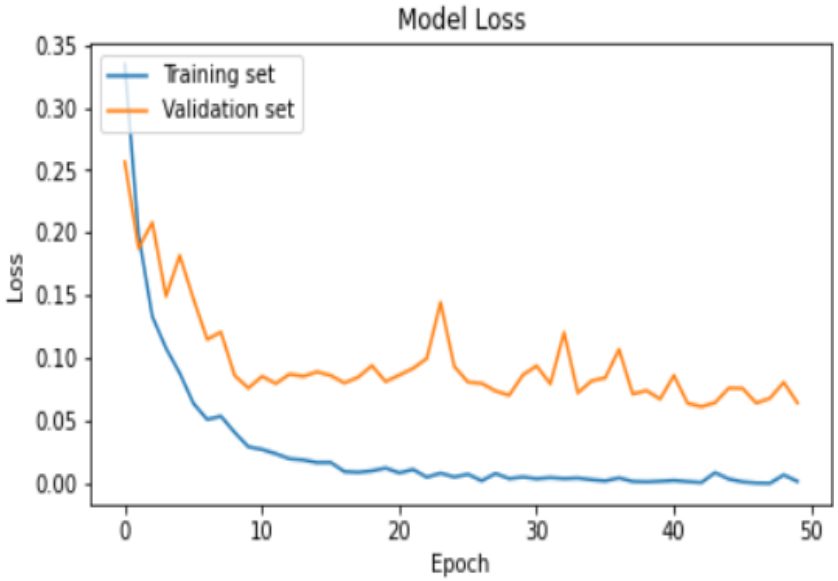

Figure - 37 - VGG16 - Fine -Tuning - Loss

All Results

Table - 10 - All Results

Model	Dataset	Network	Architecture	Results
CNN1	kaggle	CNN	2 Conv Layers, 2 Max pooling, 1 Flatten Layer, 1 Dense Layer, 1 Fully Connected Layer with batch default size	Testing accuracy - 93.54%
CNN 2	kaggle	CNN	2 Conv Layers, 2 Max pooling, 1 Flatten Layer, 1 Dense Layer, 1 Batch Normalization, 2 Fully Connected Layer with batch default size	Testing accuracy - 88.80%
CNN 3	kaggle	CNN	2 Conv Layers, 2 Max pooling, 1 Flatten Layer, 1 Dense Layer, 1 Fully Connected Layer with batch size 64	Testing accuracy - 91.77%
INCEPTION -V3	kaggle	D-CNN	48 layered deep architecture	Testing accuracy - 93.54%
VGG-16	kaggle	D-CNN	16 layered deep architecture	Testing Accuracy - 94.17%
VGG16 - Fine Tuning	kaggle	D-CNN	16 layered deep architecture	Testing Accuracy - 97.96%
VGG19	kaggle	D-CNN	19 layered deep architecture	Testing Accuracy - 87.33%

Web Application Features

Home Page:

Figure - 38 - Home Page

This is the User Interface of the web-application, we have built using streamlit api which is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time.

Image Upload:

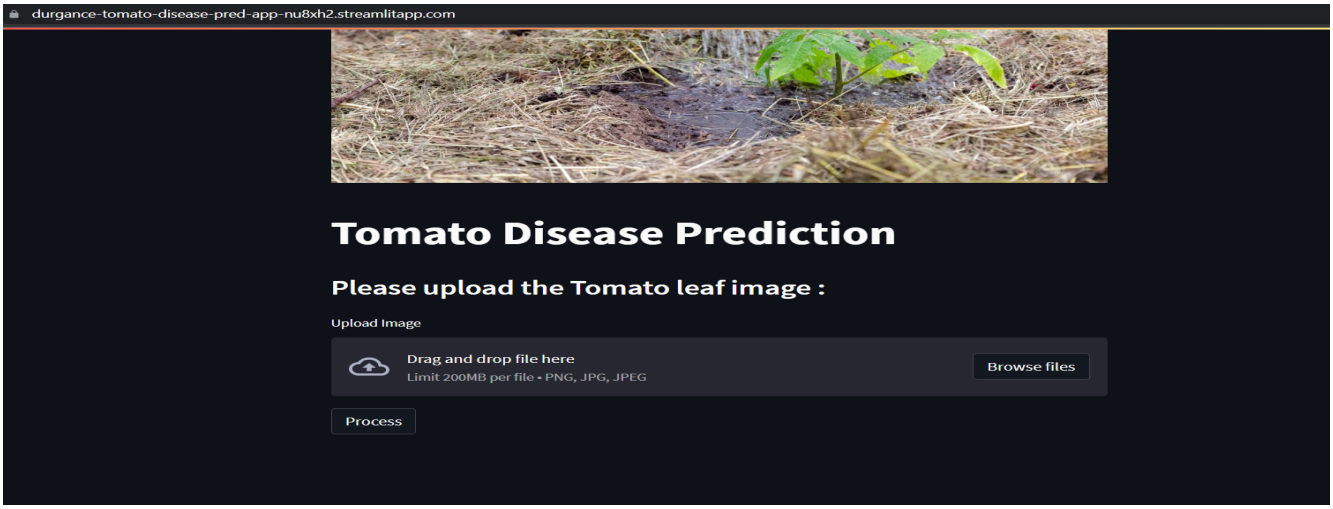

Figure - 39 - Image Upload

The above image shows the upload feature of our application. Users need to upload a leaf image and after processing in the backend they can get the quality of the leaf whether it's healthy or infected.

Process Button :

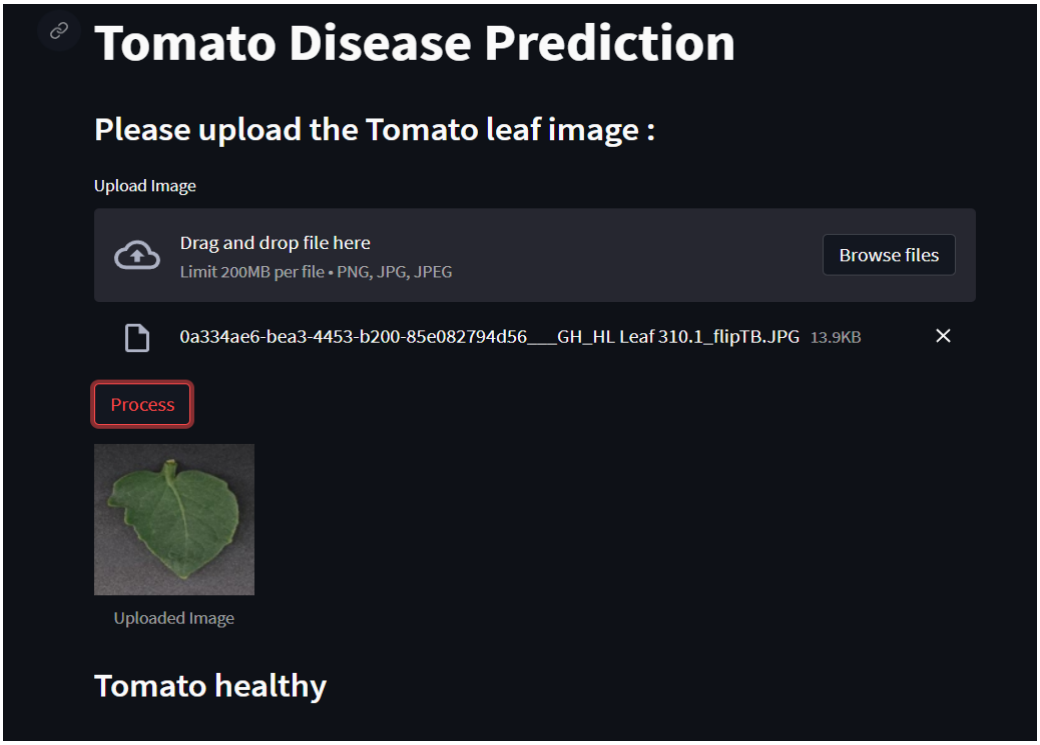

Figure - 40 - Process Button

Here, we can upload the sample images from the testing dataset and click process to classify a plant leaf as diseased with the disease category or as an healthy leaf.

Moroever, I have now uploaded a healthy image of the leaf and we can see that the website classified the healthy leaf correctly as healthy.

Output / Result:

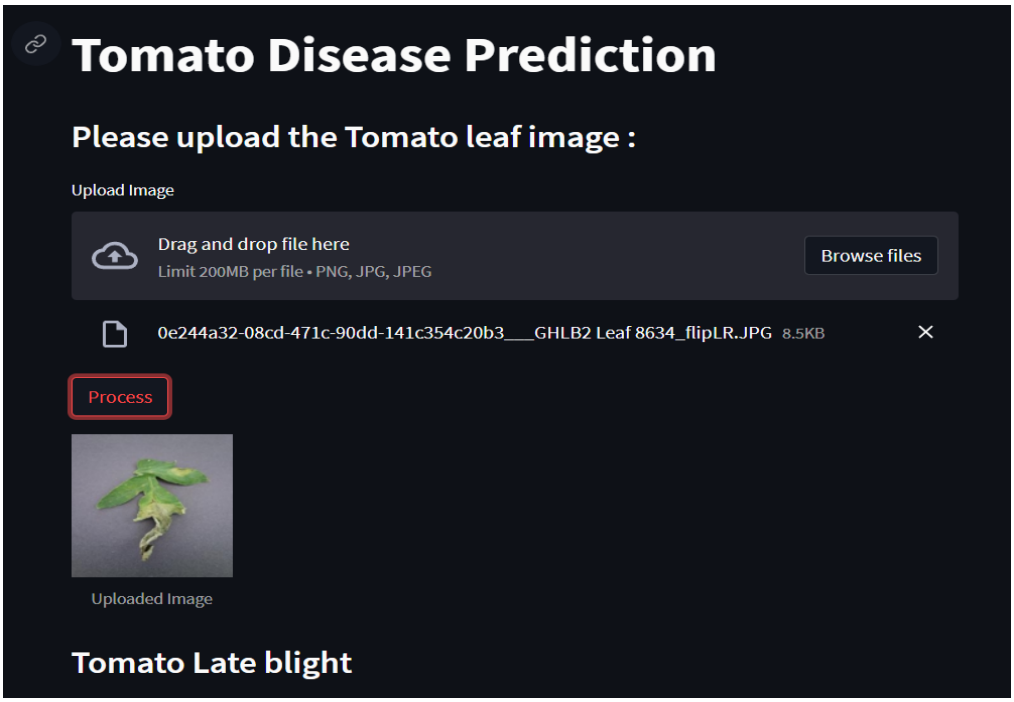

Figure - 41 - Output / Result

The above image shows how the page updates after uploading an image. The uploaded image along with the result about the respective leaf will be displayed. Here, a diseased leaf is uploaded and the website classified it correctly with the disease name.

Conclusion

Plants are a very essential part of any living being's life so, it’s very important to have a healthy plant and majorly leaves as they are used by plants to process their food. Now narrowing down the topic, let’s discuss the plants that we need i.e. crops and fruits. So, it’s difficult for farmers to keep track of all plants in the field and then manually separate healthy and diseased plants from them. It also takes a lot of manual work and time.

In this report, we showed all our model architectures starting from Convolutional Neural Network (CNN) to Inception-V3, VGG16, and VGG19.

We have taken a publicly available dataset from kaggle for our model building. We built 3 convolution, neural network models, inception-V3, VGG-16 and also fine tuned our VGG-16 to classify plants whether they are healthy or diseased, we also worked on the VGG-19 model. The accuracy we were able to achieve with this network is 98% on the dataset.

Finally, we showed the main features of our application i.e. giving a picture of a leaf as input and getting the name of the disease by which its being infected and how it works. Therefor, this can be really helpful for farmers to monitor the quality of crops.

List of Figures

List of Tables

List of Abbreviations and Symbols

- CF - Confusion Matrix
- TP - True positives
- FP - False positives
- TN - True negatives
- FN - False negatives
- CA - Classification Accuracy
- CNN - Convolutional Neural Network
- DNN - Deep Neural Network
- ReLU - Rectified Linear Unit
- ELU - Exponential Linear Unit

Our Code Implementations

CNN - 1:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/CNN1

CNN - 2:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/CNN2

CNN - 3:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/CNN3

VGG-16:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/VGG16

VGG-16 Fine-Tuning:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/VGG16-Fine-Tuning

VGG-19:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Codes/VGG19

Web Application URL:
https://durgance-tomato-disease-pred-app-nu8xh2.streamlitapp.com/

Web Application Code:
https://github.com/anirudhjak06/Crop-Disease-Detection/tree/main/Tomato-Web-App/Web-App%20using%20streamlit%20and%20VGG-19%20Model

Acknowledgements

1. Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural networks VGG. Artificial Intelligence in Agriculture. Link

2. Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018, May). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In 2018 IEEE. International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE. Link

3. VGG16 | D-CNN Model | GeeksforGeeks Link

4. Extract Features, Visualize Filters and Feature Maps in VGG16 and VGG19 CNN Models | Towards Data Science Link