TOMATO PLANT DISEASE DETECTOR

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

Sudhansh Sharma (171312)

Under the supervision of

Dr. Ekta Gandotra

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled "**TOMATO PLANT DISEASE DETECTOR**" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering and Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from **August 2020 to December 2020** under the supervision of **Dr. Ekta Gandotra, Assistant Professor (Senior Grade),** Computer Science and Engineering/Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sudhansh Sharma, 171312

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

Ekta

Dr. Ekta Gandotra

Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Dated: 14/05/2021

ACKNOWLEDGEMENT

Our thanks and appreciations also go to our fellow mates in developing the project "**TOMATO PLANT DISEASE DETECTOR**" and people who have willingly helped us out with their abilities. We are highly indebted to all the members of Computer Science Department, Jaypee University of Information Technology for their steerage and constant supervision yet providing necessary data related to the project and additionally for their support in finishing off the project. We would also like to express our gratitude towards **Dr. Ekta Gandotra (Department of Computer Science & Engineering and Information Technology)**, as our project Supervisor, for her kind co-operation and encouragement which helped us in completion of this project and for giving us such attention and time.

ABSTRACT

Computers are widely being used for mechanization and automation in various fields and have also found usage implementations in the field of agriculture. The identification of disease on the plant is an important key factor to prevent the loss of yield. The symptoms can be observed on various parts of the plant such as leaf, stem and fruit. Our work focuses on the leaf part of the plant. The leaf of the plant shows symptoms such as changing colour as well as spots on it. Presently, the leaf is observed by the naked eye and thus, it is a time consuming process. The aim of our work is to automate this process by identifying a diseased leaf from the leaf image on the basis of Inception v3 and ResNet50 model that is integrated on a Django Framework based web application. The method suggested by this takes an approach via clustering. The initial step here is to collect a dataset augment it and then, the images has to be resized so as to make the algorithm perform much faster. Later, the trained high performing model is integrated with a Django web app. Finally, using it the prediction is made and the output is produced.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter -1

INTRODUCTION

1.1 Introduction

The technique for converting an image into digital form and then later-on performing a series of operation on it to get an enhanced/improved image is known as Image Processing. Further this enhanced image is used for extracting out useful information from it. In the agricultural sector Image Processing has made an enormous progress and is being used on a large scale. Now the systems used to detect the plant pathologies can be automated. A central point that underpins the presence of life on earth begins from where it counts in the natural pecking order, the plants. Every one of these plant is inclined to different sicknesses since they are presented to the different states of nature. Due to these diseased plants the surrounding plants may also get diseased and may lead to destruction of the whole agricultural production. Tomato is a generally utilized harvest in a nutritious world with an extraordinary taste and wellbeing impact, assumes a significant part in horticultural creation and exchange far and wide. Given the significance of tomatoes, various procedures are utilized to amplify profitability and quality. Numerous illnesses and conditions can influence tomatoes during the developing season. The most well-known tomato sicknesses are target spot, spider mites, yellow leaf curl, bacterial spot, mosiac virus, late blight, early blight and septoria leaf spot.

Detection and Classification of diseases is an important task in the agricultural sector because Plant diseases are an important issue because they significantly reduce the quantity and quality of agricultural production. As the rural misfortunes are immense in light of these infections. Distinguishing and relieving these sicknesses at an exceptionally simple stage helps set aside a ton of cash and exertion.

There are various diseases that in the initial stage can be identified with the naked eyes, but far more number of diseases need some special learning to be recognized at the initial stages, for which an Machine Learning system is best suited. One more factor is in small cultivation fields and farms, farmers can manually identify the diseased plants but in larger fields due to the enormous number of the plants' sample it will be difficult to do it manually.

As an answer for this issue, I have created a system that uses deep learning to analyze, detect and classify any disease that might have affected a plant by taking an image of the leaf as an input for this mechanism.

1.2 Motivation

Farming is the main area of the Indian Economy and the horticultural area gives the biggest work in provincial regions. The horticultural yield of India is the second most noteworthy on the planet after China. This area gives work to around 50% of the nation's labor force and records for around 17 to 18 percent of India's Gross Domestic Product. Numerous varieties of common items, grains, vegetables, flavors are created in India are furthermore conveyed to various countries, tea being a huge model. Thus growing quality creation has become crucial bit by bit. However, these harvests are powerless against different viral, bacterial and contagious illnesses and biological components. Because of the hefty reliance on the Farming Sector, it is essential to shield the yields from coming down with sicknesses. With our nation's profoundly unusual climate and different other natural and non-organic factors, the harvests are effectively defenseless to sicknesses which can genuinely influence the yield and nature of the harvests being created. Along these lines, making enormous financial misfortunes ranchers as well as the nation. Thusly, it is of most extreme critical to recognize sicknesses in yields as right on time as conceivable to keep it from spreading further. Illnesses in different yields can spread through abiotic just as biotic component. Along these lines, to guarantee appropriate yield and nature of the harvests, it is basic to give enough insurance, distinguishing proof and even treatment to the yields as and at the point when important. So to automate the task for identifying the diseased plants we developed a system which can identify the diseased plants simply without having any prior knowledge on the subject as well.

1.3 Problem Statement

Agriculture plays a major part of Indian Economy and our farmers are suffering due to diseased plants that lead to low agricultural production. Plant illnesses have transformed into a situation as it can cause huge decrease in both quality and amount of horticultural items. Programmed identification of plant infections is a fundamental examination theme as it might demonstrate benefits in checking enormous fields of harvests, and in this manner naturally identify the side effects of illnesses when they show up on plant leaves. The proposed framework is a product answer for programmed discovery and characterization of plant leaf sicknesses. Here we developed a web integrated trained model which takes the image as input and then results whether the image is healthy, if not then what is the disease. The plan for identifying the diseased comprises of four principle steps, initial a shading change structure for the input RGB image is created; at that point the green pixels are concealed and taken out utilizing explicit edge esteem followed by division measure, the surface insights are processed for the valuable portions, at last the removed highlights are gone through the classifier. In classifier the infection can be distinguished and answer for illness can be found.

1.4 Objective

As the monetary arrangement of India depends absolutely on horticultural creation, outrageous consideration of food fabricating is important. Transporters like an infection, organism and microorganism's reason defilement to verdure with misfortune in quality and measure of assembling. There is a huge amount of absence of ranchers in assembling. Subsequently appropriate consideration of plants is basic for the equivalent.

Checking of plants/crops and their organization from the earliest starting point time is most critical. An enormous degree of farmers in India sprinkles pesticides on cash harvests, vegetables or natural item plants. A significant part of the time it has been seen that overdose of pesticides is over 40%. Hereafter it hurts plants/crops similarly as to individuals. So there is a need to check for the illnesses in the plants. Different plant illnesses can straightforwardly be distinguished by unaided eye. One such normal descriptor is Leaf Spot. Leaf spots allude to the dull shaded, normally earthy however now and then dark, spots happening on the leaves of different plants. Greater parts of Leaf Spots are brought about by organisms or on the other hand microscopic organisms yet these might be brought about via air toxins or bugs. The subsequent spots contrast for different plant species and the specialist causing them. They additionally differ by size and color. To forestall the further spread of such sicknesses, it is smarter to distinguish them at an underlying stage and give legitimate treatment. The treatment can be as basic as eliminating the contaminated leaves to restrict the spread whenever identified early. Be that as it may, in the event that it is past the point of no return, at that point now and then an extremely intense proportion of supplanting the whole yield can likewise be required.

In little fields, the human labor force can be utilized to help identify and perceive different unhealthy yields. However, there is consistently a factor of human blunder which may end up being lethal in the event of enormous scope ranches. With the presentation and advancement in the field of Computer Vision, different new strategies are arising which can help in effectively distinguishing and perceiving ailing yields.

Our objective was to make a system which takes the input image of a tomato leaf and tells if it is diseased with what type of disease or is it healthy. So we have made two models one based on Inception_v3 and the other one on Resnet, these are integrated with Django based webpage, where we will input the image; which will give a message about the type of disease in our tomato plant leaf.

1.5 Methodology

1.5.1 Collection of Data:

The first step is to formulate a dataset. In our case, the dataset consisted of images of leaves of various plants of **Tomato**. These images can be taken using a Digital Camera or a Smartphone Camera.

We explored various datasets like the Plant Village dataset which is an enormous dataset consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease. This dataset was made in laboratory conditions so we explored a different dataset which was made on field which depicts the state of leaf in a natural state and contains 2,598 data points in total across 13 plant species and up to 17 classes of diseases, involving approximately 300 human hours of effort in annotating internet scraped images. We finally considered a dataset for only tomato leaves which was available on kaggle and was also made with the help of some other datasets which resulted in 16,012 images of tomato leaves divided in 10 categories.

Description	Number of Images
Tomato Bacterial Spot	2127
Tomato Early Blight	2400
Tomato Late Blight	2314
Tomato Leaf Mold	2352
Tomato Septoria Leaf Spot	2101
Tomato Spider Mites	2176
Tomato Target Spot	2284
Tomato Yellow Leaf Curl Virus	2451
Tomato Mosaic Virus	2238
Tomato Healthy	2407
TOTAL	22850

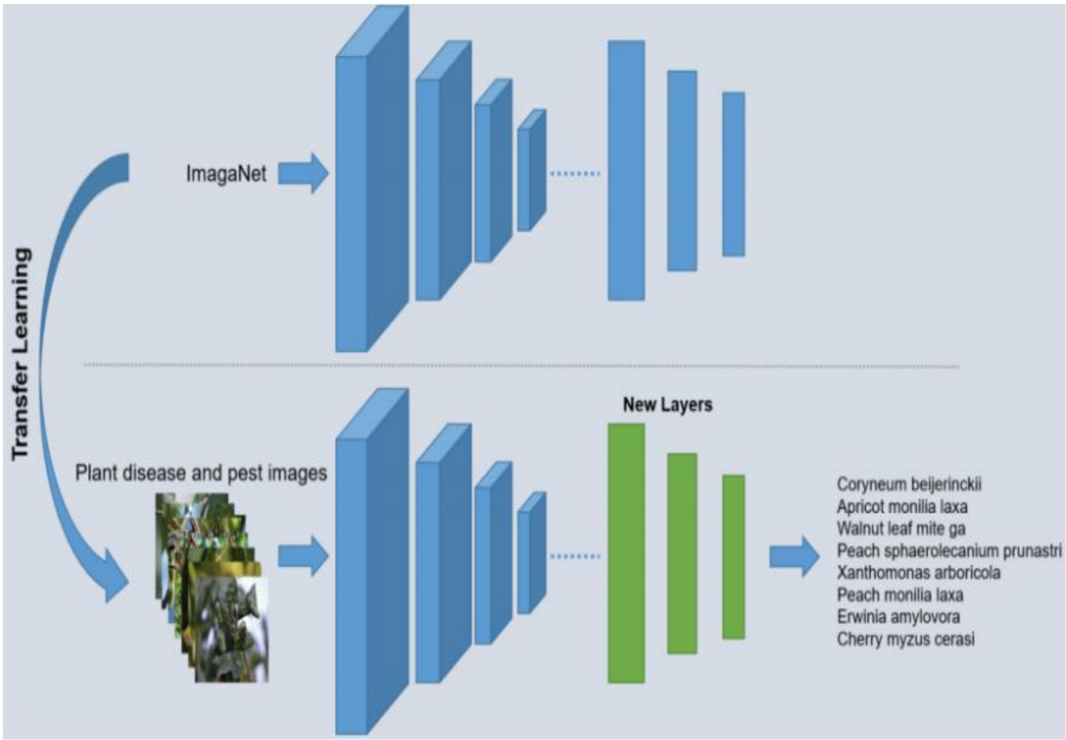
Table 1 Details of Dataset

1.5.2 Transfer Learning

Transfer learning has been highly successful for classification problems. The main advantage in using transfer learning is that instead of starting the learning process from scratch, the model starts from patterns that have been learned when solving a different problem which is similar in nature to the one being solved. This way the model leverages previous learning and avoids starting from scratch. In image classification, transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a similar problem to the one that we want to solve.

Transfer learning is a machine learning method that is reused as a starting point for solving a different problem by using knowledge obtained from a model developed in the solution of a problem. The current study fine-tuned this by using pretrained CNN models based on transfer learning. The reason for using pretrained CNN models is that they are faster and easier than training a CNN model with randomly initialized weights. In addition, the fine-tuning process is based on transferring new layers instead of the last three layers of the pretrained networks to our classification task.

Figure 1 Simple representation of transfer learning



1.5.3 Deep Feature Extraction:

Deep feature extraction is based on extracting features learned from a pretrained convolutional neural network. These features are used to train machine learning classifiers. In other words, this method is based on the extraction of deep features from the fully connected layer of pretrained networks. In this, effective deep features were extracted from a certain layer of deep learning models, which are **ResNet50** and **InceptionV3**.

1.5.4 Data Augmentation:

The images are resized to 256×256 pixels. I used data augmentation techniques like shearing, zooming, flipping and brightness change to increase the dataset size to more than double the original dataset size. The image rotation degree was set to be randomly generated from 0 to 45. The number of normal samples in the dataset was increased from 1,583 to 4,266 by performing the image augmentation techniques. In this manner, the number of samples for each class was equalized. This equal distribution makes it possible to use all of the data instead of selecting random data during the training process. It is expected that this situation increases the accuracy of the training and positively affects the classification results.

1.5.5 Summary of the Architecture:

The following steps summarize the proposed deep feature extraction:

- Acquire plant images
- Resize plant image according to deep networks using bilinear interpolation. For instance, color images sized 224 × 224 are used in ResNet50 and Inception V3, respectively.
- Features are extracted using the fully connected layers of the deep learning models
- Classification is performed using the deep features with classifiers

The following steps summarize the transfer learning:

- Acquire plant images.
- Resize plant image according to deep networks using bilinear interpolation.
- An additional last layer is added to the pretrained Inception v3 and ResNet model , and a classification output layer in order to adopt the pretrained CNN networks to solve the problem.
- Classification is performed using the newly created deep model

We split the data-set into three sets — train, validation and test sets.

We tried with pre trained models like Inception v3 and ResNet 50. The last layer is used for the classification with softmax as the activation function.

The loss function used is binary cross-entropy and trained the model for 15 epochs with a batch size of 16. Multi class log loss is chosen as the evaluation metric. Activation function used was Relu throughout except for the last layer where it was Sigmoid as this is a binary classification problem.

We used 30 percent dropouts to reduce over fitting in between the layers and batch normalization to reduce internal covariate shift.

1.6 Technologies and Frameworks Required

1.6.1 Google Colaboratory:

Google Colaboratory is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs. It does not matter which computer you have, what its configuration is, and how ancient it might be. You can still use Google Colab! All you need is a Google account and a web browser. And here's the cherry on top – you get access to GPUs like Tesla K80 and even a TPU, for free! Training models, especially deep learning ones, takes numerous hours on a CPU. We've all faced this issue on our local machines. GPUs and TPUs, on the other hand, can train these models in a matter of minutes or seconds. Colab gives us 12 hours of continuous execution time. After that, the whole virtual machine is cleared and we have to start again. We can run multiple CPU, GPU, and TPU instances simultaneously, but our resources are shared between these instances. It will cost you A LOT to buy a GPU or TPU from the market. Why not save that money and use Google Colab.

1.6.2 Django Web Framework:

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. This is the most popular framework available in python. It follows the MVT or Model-View-Template pattern. It is closely related to other MVC frameworks like Ruby on Rails and Laravel. In the MVC framework, the view and model parts are controlled by the Controller but in Django, the tasks of a controller are handled implicitly by the framework itself.

Django helps you develop software that are complete, versatile, secure, scalable, maintainable, portable and secure.

One of the rarest and most desirable skills in tech is the ability to combine machine learning and data science skills with practical web development. Django lets you create a number of applications under a single project. The application has all the functionalities to work independently. The app is considered as a package which you can reuse in other projects without making any major changes. This is the greatest advantage of using Django for building web applications.

1.7 Organisation

In **Chapter 1**, we have discussed the importance of agriculture in India, how it contributes towards the economy of India. The main focus is on the diseases of the plants and how it can be treated using different techniques of image processing.

In **Chapter 2**, we have discussed the research papers we have alluded to so as to improve comprehension of our task. The papers fundamentally center around strategies utilized in AI and different inquiries about did in this field.

In **Chapter 3**, we have talked about the strategies in detail and furthermore the methodologies used to foresee the result and viability of our outcome. In brief we discussed the system development.

In **Chapter 4**, we have talked about the strategies for performance analysis in details and compared the performances of Inception v3 and ResNet models. Usage and the consequences of the yields have been examined.

In **Chapter 5**, we have given the ends that have been gotten from this examination also, the future extent of this project.

Chapter -2

LITERATURE SURVEY

Literature survey has been conducted pertaining to tomato leaf diseases and significant works have been done by the researchers. Some of these works are presented here.

2.1 Detection of plant leaf diseases using image segmentation and soft computing techniques

2.1.1 Author: Vijai Singh, A.K. Mishra

2.1.2 Publication: Elsevier, October 2016

2.1.3 Summary: In this paper, the authors presented an algorithm for image segmentation which can be utilized for discovery and classification of leaf disease.In their proposed procedure Image procurement is done with the assistance of digital cameras. To expel contortion from the picture ,preprocessing is done on the picture and afterward a smoothing filter is applied . To evacuate the healthy green pixels a threshold is registered and the pixels having intensity less than the pre-figured edge esteem at that point zero is allocated to that pixel.Finally,useful sections are acquired to characterize leaf diseases.This technique can distinguish the plant malady at introductory stage and ideal outcomes shows proficiency of calculation in acknowledgment and characterization of leaf sickness.

2.2 An Individual Tomato Leaf Disease Identification Using Leaf Skeletons and KNN Classification

2.2.1 Author: N. Krithika, Dr. A. Grace Selvarani 7

2.2.2 Publication: ICIIECS Conference, 2017

2.2.3 Summary: In this paper they proposed grape leaf disease identification by using KNN classification. In the initial step images are acquired and enhanced on the basis of color and smoothness and after that leaf skeleton structure was used to identify the grape leaf . Before extracting the features noise from the image was removed by using denoising technique. Proposed method mainly recognizes the luminance and linear qualities of the leaf skeleton. KNN was utilized for leaf illness grouping. Preferred position of proposed strategy is to decrease the acknowledgment time and computational intricacy.

2.3 A new automatic method for disease symptom segmentation in digital photographs of plant leaves

2.3.1 Author: Jayme Garcia, Arnal Barbedo

2.3.2 Publication: Springer, July 2017

2.3.3 Summary: In this paper authors proposed methods to utilize Color channel manipulation and Boolean activities applied on binary masks making it progressively easier and vigorous. Algorithms was more precise and not exclusively were the error rates lower in all disease classes however the proposed algorithm was reliably hearty estimating from all side effect categories. This show that arithmetic manipulation algorithm is fit for recognizing a wide assortment of side effects, being quicker and nearly as exact as manual segmentation strategies.

2.4 Symptom based automated detection of tomato diseases using color histogram and textural descriptors

2.4.1 Author: H. Ali, M.I. Lali, M.Z. Nawaz, M. Sharif, B.A. Saleem

2.4.2 Publication: Elsevier, 2017

2.4.3 Summary: In this paper authors presented an approach for identification and classification of disease in citrus plants. They applied Delta E , LBP, RGB and HSV segmentation methods as descriptors. Their algorithm collects features of color image from histogram of segmented image, histogram for every channel is calculated and feature set is formed by concatenating features in a single array. Classification methods were used for disease level classification. Where Bagged tree ensemble classification performance was best and LBP features had powerful image level classification.

2.5 KNN-Based Image Segmentation For Deficiency Diagnosis.

2.5.1 Author: B. M. Sanchez Rangel, M. A. Aceves Fernandez, J. C. Murillo, J. C. P. Ortega, J. M. Ramos Arreguın

2.5.2 Publication: IEEE, 2016

2.5.3 Summary: In the paper authors presented an approach to diagnose & classify grapevine leaves having potassium deficiency. Segmentation method based totally on KNN (K-Nearest Neighbors) calculates Euclidean distance between points in the area representing a specific shade and based on distances, differentiate among colourations. Histogram methods work with grayscale intensity snapshots and may not distinguish among colorations. Thus KNN proved to have higher effects specifically with much less controlled environment situations.

2.6 Detection of plant leaf disease using image processing approach

2.6.1 Author: Sushil R. Kamlapurkar

2.6.2 Publication: IJSR, 2016

2.6.3 Summary: In this paper the author suggested to reduce the influence made by the background by transformation of image into HSI colour space and further increasing contrast of image by analyzing the histogram of intensity to get the threshold. Author used Gabor filtering method for feature extraction which are important for colour and morphology of the leaf spots.Major axis,minor axis , eccentricity are the features extracted from image.And finally these features were used by classifier to classify the disease.

2.7 Plant Diseases Detection Using Image Processing Techniques.

2.7.1 Author: Shivani K. Tichkule, Prof. Dhanashri. H. Gawali

2.7.2 Publication: IEEE, 2016

2.7.3 Summary: In this paper authors give different strategies of image processing to recognize illnesses on plants. In India different infections are seen on leaves and stems of plants because of different vermins. Gabor filters were used on leaves which made SVM to achieve efficient disease classification. BPNN gives extraction with a complex background. For fruit disease like Apple Rot, Grape-Black Rot, Downy Mildew image segmentation was done using K-means clustering and feature extraction by using SURF algorithm recognizing infections. ANN for pattern matching which then is used for classifying diseases.

2.8 A clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images.

2.8.1 Author: Xuebing Bai ,Xinxing Li ,Zetian Fu ,Xiongjie Lv ,Lingxian Zhang

2.8.2 Publication: Elsevier, 2017

2.8.3 Summary: In this paper authors gave the methods to improve detection of leaf spot on cucumber leaf against a complex background. Improvements to isolate the diseased part of leaf include three a run of ‘marked-watershed algorithm’ by which diseased portion of leaf was extracted from non-target elements. Neighborhood grayscale information overcomes under use of image pixel spatial information by FCM which refines the capacity of noise filtering . The gray distance between pixels and clustering center is reclassified as ||Xj 2 - Vi 2 || which endowed a significant change to fuzzy membership in one cycle and can accomplish convergence in fewer cycles bringing about decreased running time and improved viability of the calculation.

2.9 Detection of Leaf Diseases and Classification using Digital Image Processing.

2.9.1 Author: R.Meena Prakash, G.P.Saraswathy, G.Ramalakshmi, K.H.Mangaleswari, T.Kaviya

2.9.2 Publication: IEEE, 2017

2.9.3 Summary: In this paper various image processing techniques were used with the objective to apply image analysis and classification of leaf disease. Images were acquired and then altered to 256X256 pixels. Enhancement of leaf image was done by converting RGB to L*a*b color space. K-means clustering algorithm performs segmentation by limiting the sum squares of distances between the image intensities and cluster centroids . Gray-Level Co-Occurrence Matrix (GLCM) computes spatial relationships among pixels to make characterization of textures of images. For disease classification Support Vector Machine (SVM) kernel-based supervised learning algorithm was utilized.

Chapter -3

System Development

3.1 Dataset

We explored through various datasets and finalized a part of the Plant Village Dataset for our Project. This database is divided into two datasets for tomato leaf images according to different image sources. The tomato leaf images of the first dataset are selected from the PlantVillage database with ten categories (nine disease categories and one health). Each image is composed of a single leaf and a single background, for a total of 16,012 images. Afterwards, this database is divided into five subsets of 5-fold cross-validation.

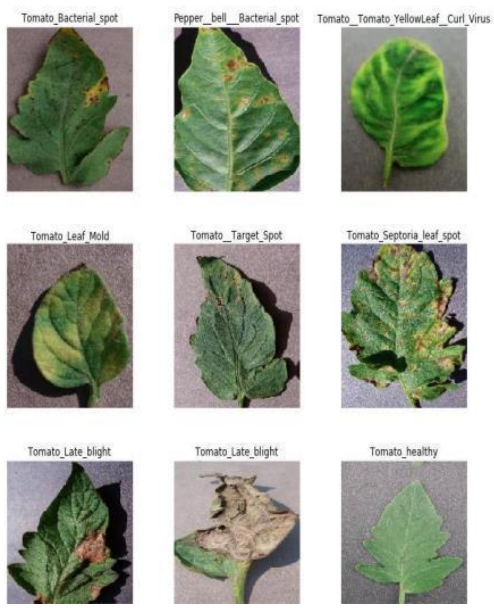


Figure 2 Sample of Dataset

The detailed categories of the first dataset are:

1. Bacterial spot
2. Early blight
3. Healthy
4. Late blight
5. Leaf Mold

6. Septoria leaf spot
7. Target Spot
8. Tomato mosaic virus
9. Tomato yellow leaf curl virus
10. Two-spotted spider mite

The size of the picture is different, and we unified the image size to 224 * 224. Then we use data augmentation method to increase the number of pictures, including clockwise rotation with 90 degrees, 180 degrees, and 270 degrees; horizontal mirroring, vertical mirroring, reducing image brightness and increasing image brightness, etc.

The dataset was given directly to the Inception v3 and ResNet which trained on them and developed a model which predicts whether a tomato leaf is healthy or not.

3.2 Inception v3

For the purpose of feature extraction, Inception v3 architecture is used. Inception is developed by google and many other researchers together. The building blocks of inception v3 are convolutions, max pooling, concats, dropouts, fully connected layers and average pooling. Batchnorm is also used throughout the model and applied to activation inputs. Feature extraction helps model to clearly distinguish between all the characteristics of the image and understand them for further interpretation.

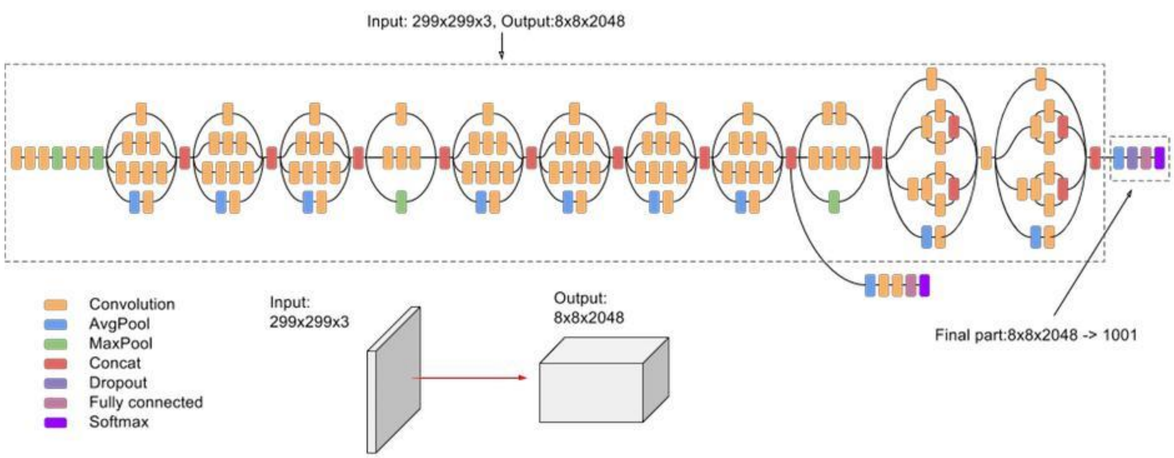


Figure 3 Inception v3 architecture

In classifying we have used convolutional neural networks for the purpose of classification. After acquiring features from inception v3, deep neural networks are used to train the model to detect presence of early blight and late blight in the images. During classification Adam optimiser is used to reduce the training time and quickly converge the loss. Adam is an extension of SGD that is now widely used for computer vision. For classification in different labels Softmax activation function is used. The softmax activation function is a function that converts a vector of n real values to a vector of n real values that totals up to 1. The input values can be any number but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities.

The CNN model contains an input layer where all images are fed in the start. After which the images are fed to the inception v3 architecture to extract features. These feature extracted images are sent to deep neural nets that classify our images on pre-trained knowledge. Finally, the images are sent to the output layer. The model has a couple of layers that are first created and after that they are compiled together using functions from the TensorFlow library.

The model that we have proposed in this paper has used the plant village dataset which has about 1000 leaf images of early blight and 16,012 images of healthy plants. For this model, the dataset has been divided into 2 parts that are the training set and the test set. Training set comprises of 80% of the dataset while the test set comprises of 20% of the dataset. The pre-trained model used on this dataset for feature extraction is Inception V3. Here for classification our CNN model provides an **accuracy of about 92%** on the basis of the training and testing done on it.

3.3 ResNet 50

ResNet50 is having very deep layers so that each and every feature of the image is obtained. Automatically the accuracy or recognition rate also increases. Resnet50 architecture consists of two sub-blocks they are convolution block and identity block. Convolution block comprises of convolution-2D, batch normalization, activation function that is ReLU. At last, the input is given to convolution and batch normalization and they will be added to the output. Finally, it will be given to the activation function. Identity block comprises activation function, batch

normalization, and convolution. At last again the input is added to the obtained output and it is given to the activation function. The main architecture follows the sequence of convolution block and identity blocks as per requirement. It involves max-pooling, zero padding, average pooling and they are connected to the flatten layer and the output image is obtained from the fully connected layer.

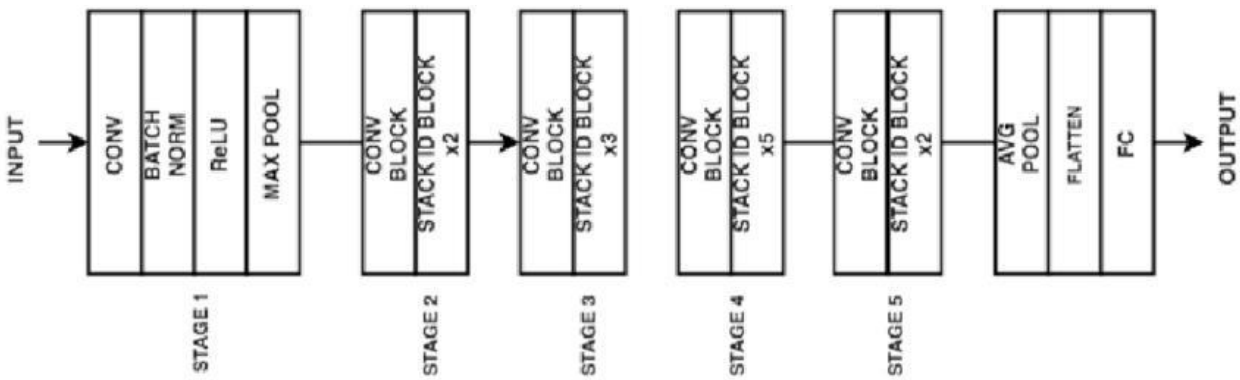


Figure 4 ResNet50 Architecture

ResNet50 is a convolutional neural network trained on more than a million images from the ImageNet database. This network takes a 224 ×224 image and produces an output with a probability of a specific class. ResNet50 contains 50 layers deep, and can classify images into 1000 object categories, including keyboard, mouse, pencil, and animals. In 2015, ResNet emerged as the first winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) classification task.

The model that we have proposed in this paper has used the plant village dataset which has about 16,012 images. For this model, the dataset has been divided into 2 parts that are the training set and the test set. Training set comprises of 80% of the dataset while the test set comprises of 20% of the dataset. The pre-trained model used on this dataset for feature extraction is ResNet50. Here for classification our CNN model provides an **accuracy of about 76%** on the basis of the training and testing done on it.

3.4 Pseudo-Code for an ANN Transfer Learning Model

Input:

(X) Our gathered feature metric of (n x m) dimension where n denotes no of features and m denotes total no of training examples.

(Y) Metric of (1 x m) dimension where each unit defines actual output value of given input X.

Output:

Trained Model on which we can predict our results.

BEGIN

Initialize Variables

Let A = X (Input Dataset)

Let NE = total number of epoches.

Let BL = total no of layers in our Base Part.

Let HL = total no of layers in our Head Part.

(W and B) should be initialized for layers that are in head part only.

Let W = Weight Metrics of Dimension ($n^{[l+1]}$, $n^{[l]}$), Initialize Randomly to Break Symmetry

Let B = Bias Metrics of Dimension ($n^{[l+1]}$,1), Initialize all values to random or zeros.

Let l = no of hidden units in a layers of head part (lets assume as for now that each layer has same number of hidden units)

Divide your model into two parts

Base Part (Layers of Pre-trained Model, this part will provide us with all the low level features)

Head Part (Layers of our own model which will use those low level features to minimize cost)

Main Formulae:-

Generalized step of Forward Propagation = relu(w*X + b) #matrix Multiplication Generalized step of

BackProp = dz[l] = (w[l+1].*dz[l+1]) * relu_backwords(z[l])

```
FOR (iteration_count →0 to NE):
    FOR(layer_number→0toBL):
        A=compute_forward_prop(A)
    END FOR
FOR(layer_number→0toHL):
A=compute_forward_prop(A)
    END FOR
    cost = calculate_loss(A, Y)
    FOR(layer_number → HLto0):
        (dw[layer_number], db[layer_number]) = compute_backprop(A,cost,W,b)
    END FOR
    FOR (layer_number →BL to0):
        (dw[layer_number], db[layer_number]) = compute_backprop(A,cost,W,b)
    END FOR
    update_parameters(W, dW,b,dB)
END FOR
```

END

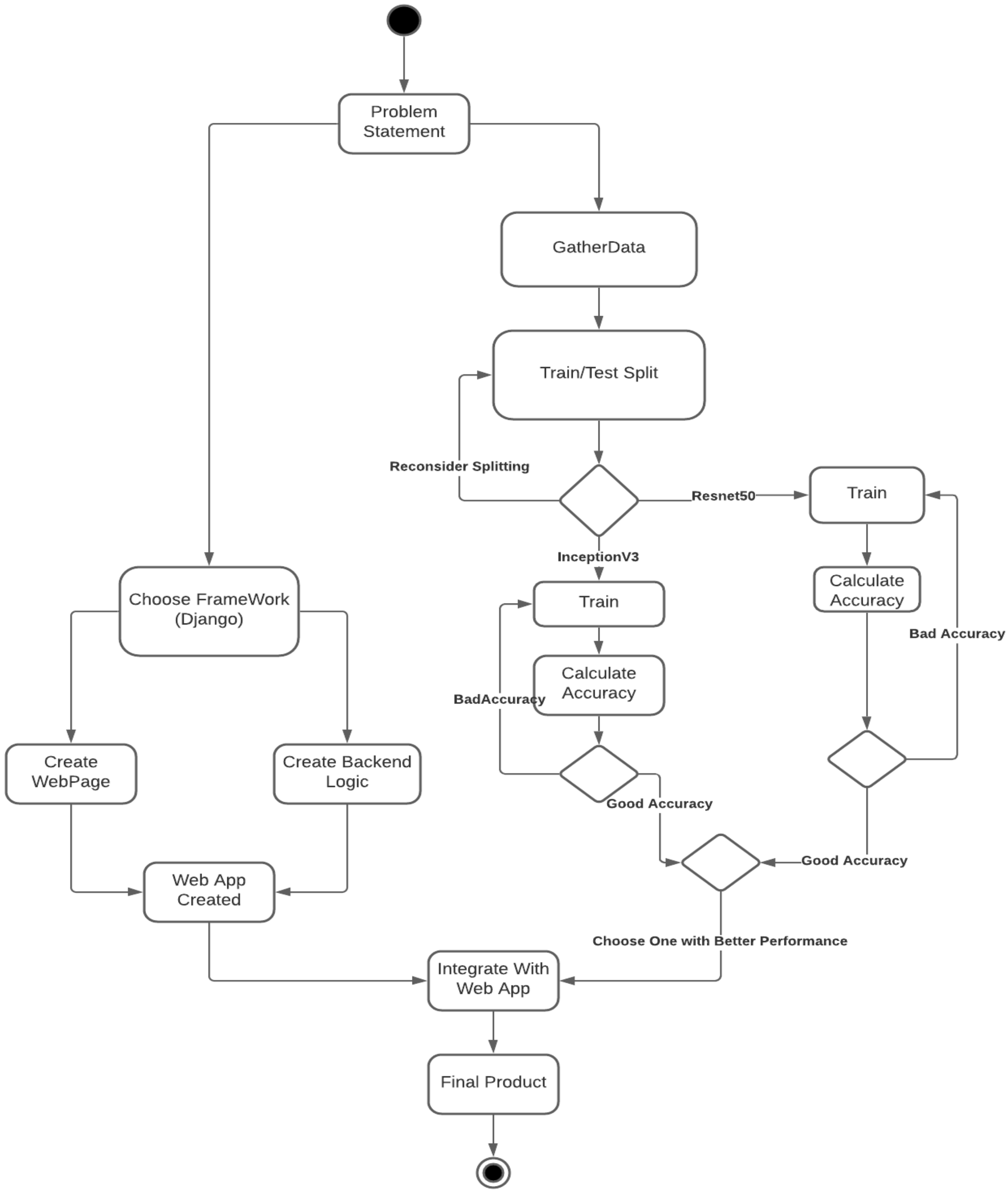
After implementing the transfer learning on Inception v3 and ResNet50, both these trained models needed to be integrated on a Django Based web page application.

3.5 Django Integration

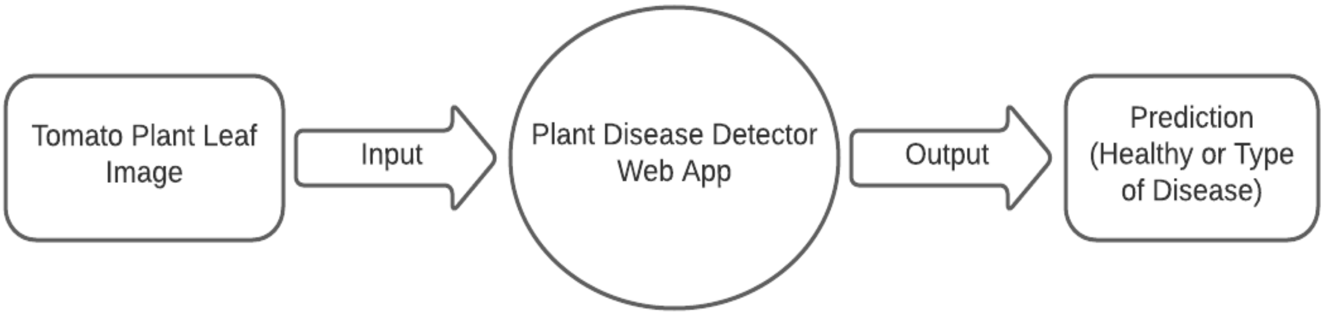
Then we developed the frontend for our web based app which will fetch the image from database and give the result whether the plant is diseased or not and will also tell what type of disease it has. The result was given in the form of an ajax query. The Trained model was integrated on this Django Based web page which worked as a common functionality for predicting healthy or not tomato leaves.

3.6 Diagrams

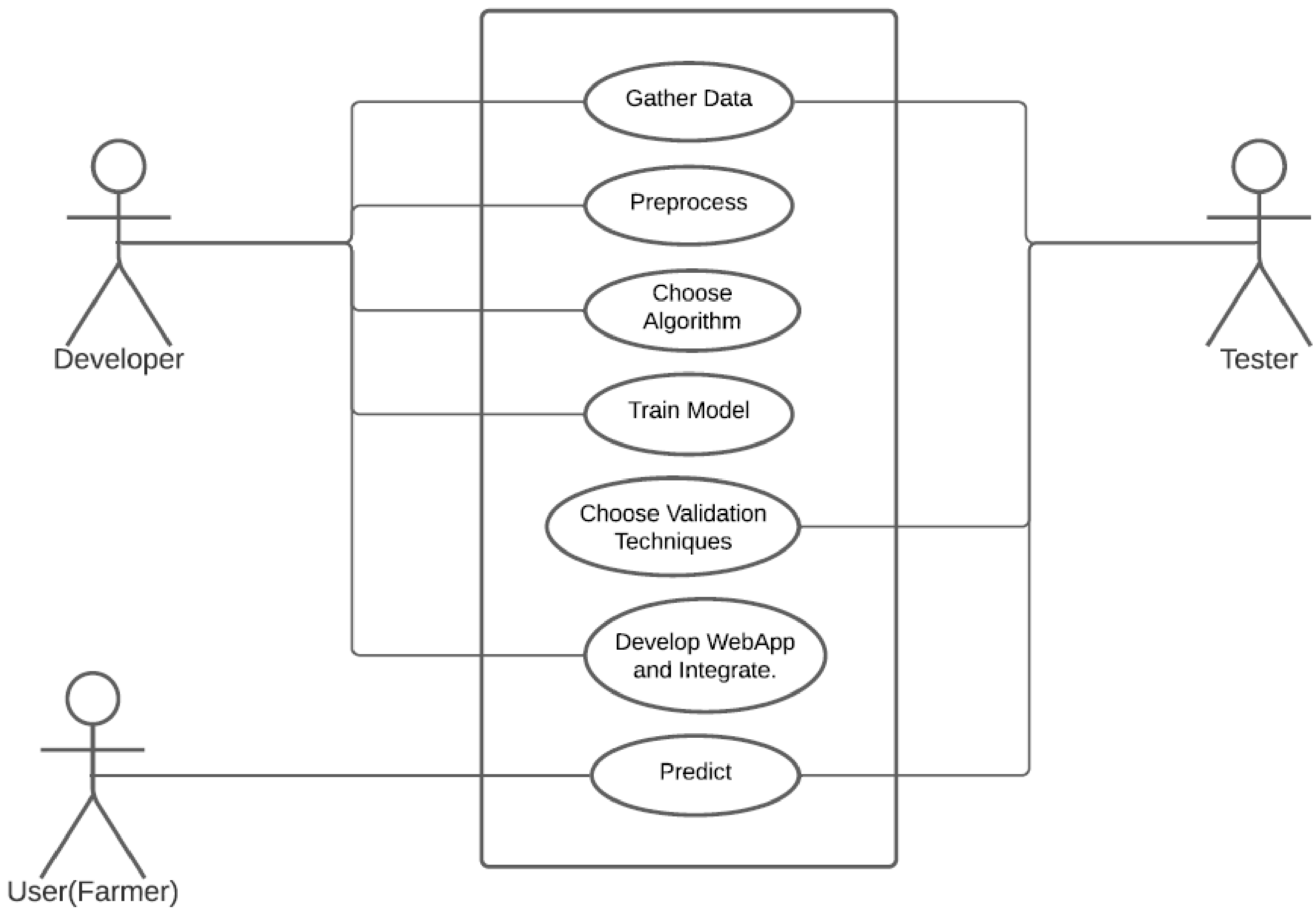
3.6.1 Activity Diagram



3.6.2 Flow Diagram



3.6.3 Use Case Diagram



Chapter -4

RESULTS

4.1 Output

This is how our web page looks like, we can select the image and it will display the result for the supplied image.

There are some screen shots depicting the working of our web page integrated model, like how the front web page looks, how the input is taken and how the result is displayed; as well as a screenshot for the backend working of the trained Ml model.

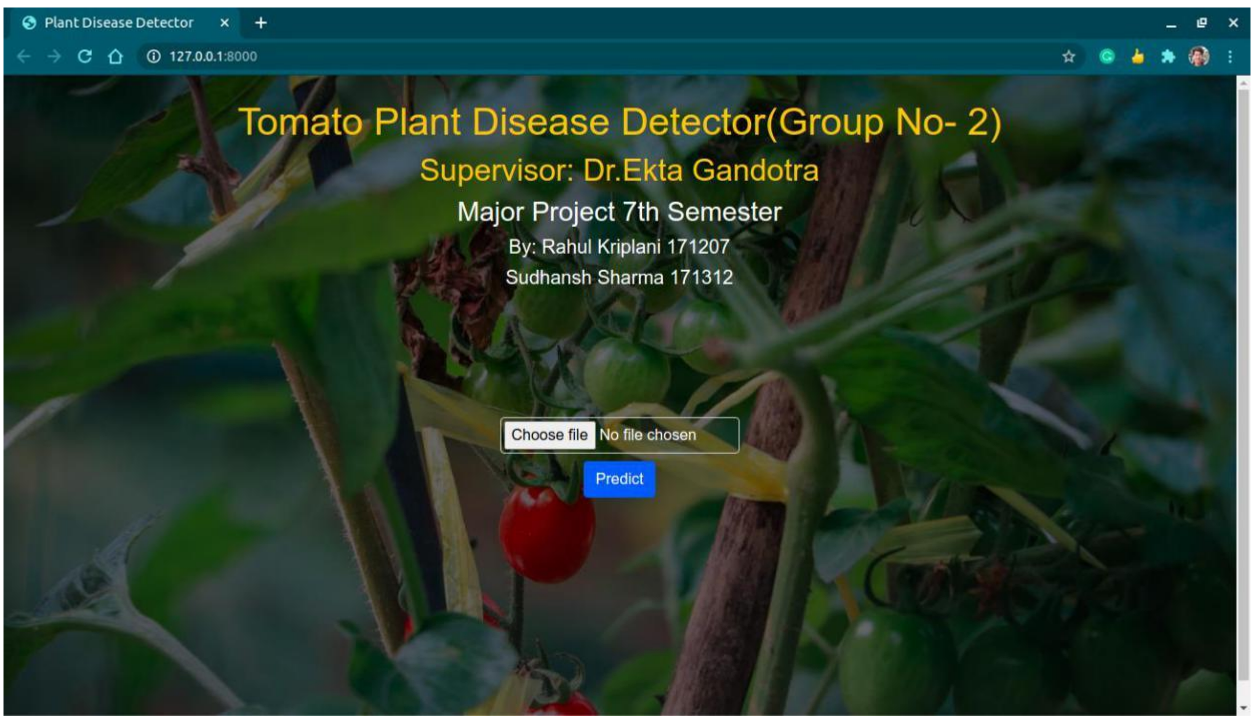


Figure 5 Web Page

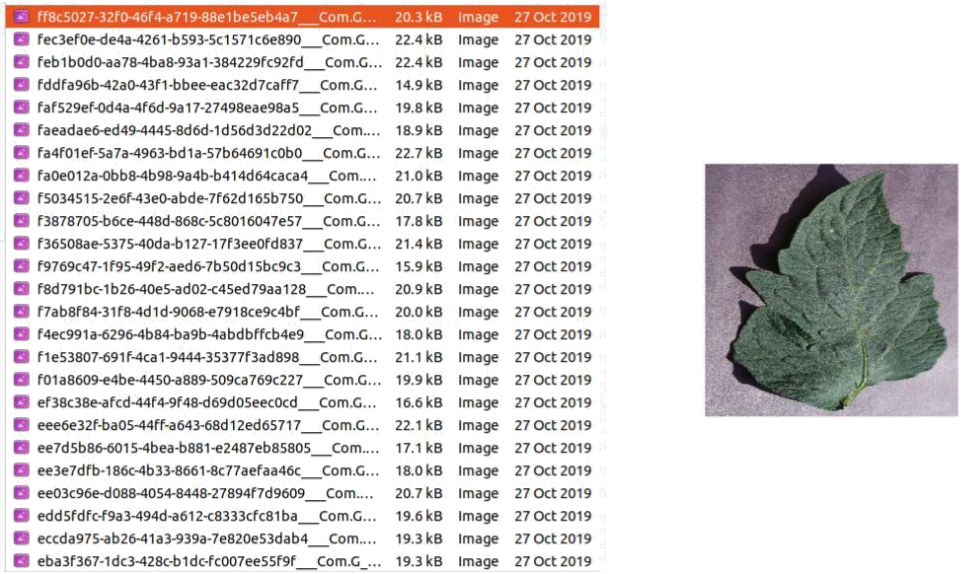


Figure 6 Selecting Image

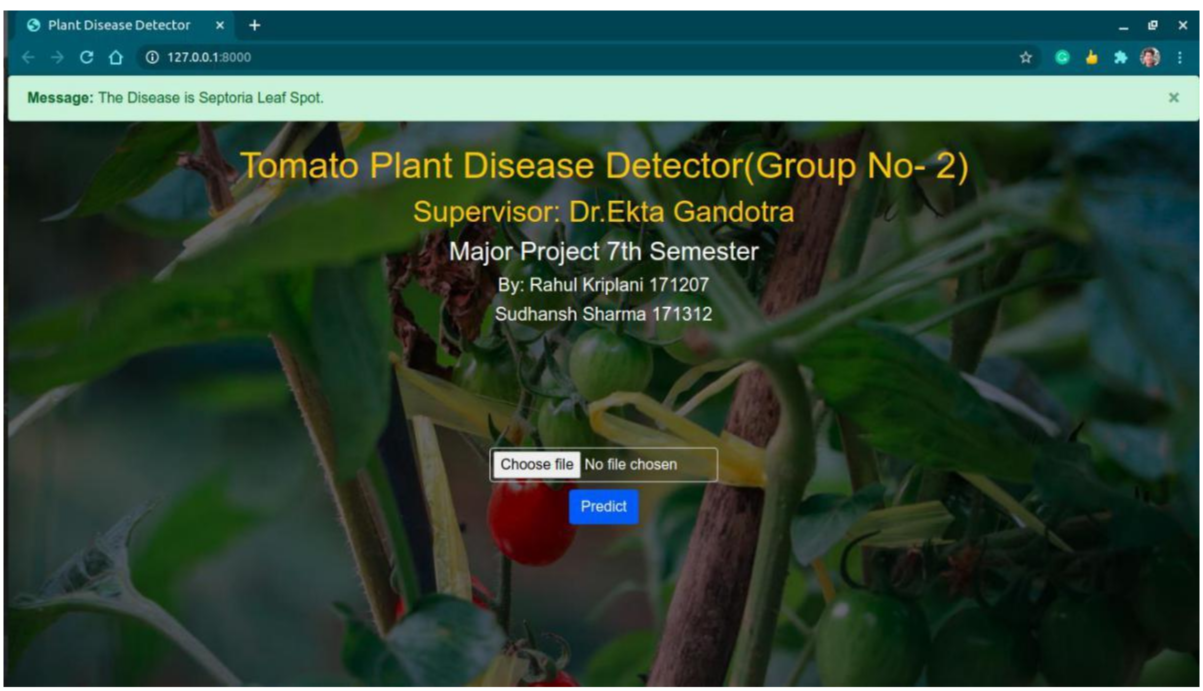


Figure 7 Result

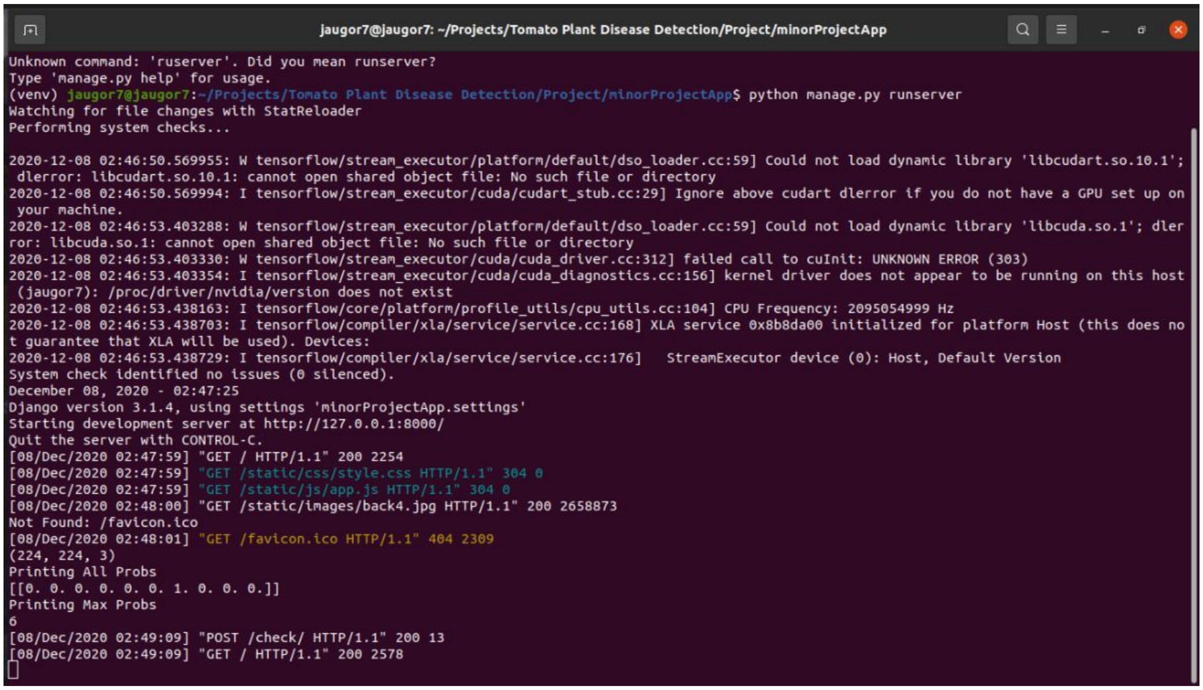


Figure 8 Backed Working

4.2 Performance Graphs

Here we have added the resultant graphs which depicts the accuracy and losses for inception v3 and resnet models, which were formed during the training.

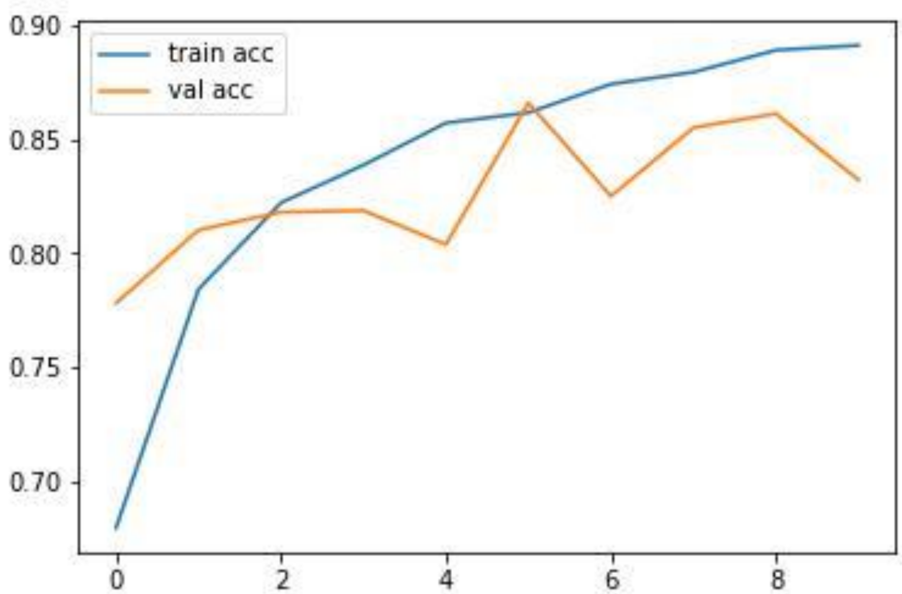


Figure 9 Inception v3 Accuracy

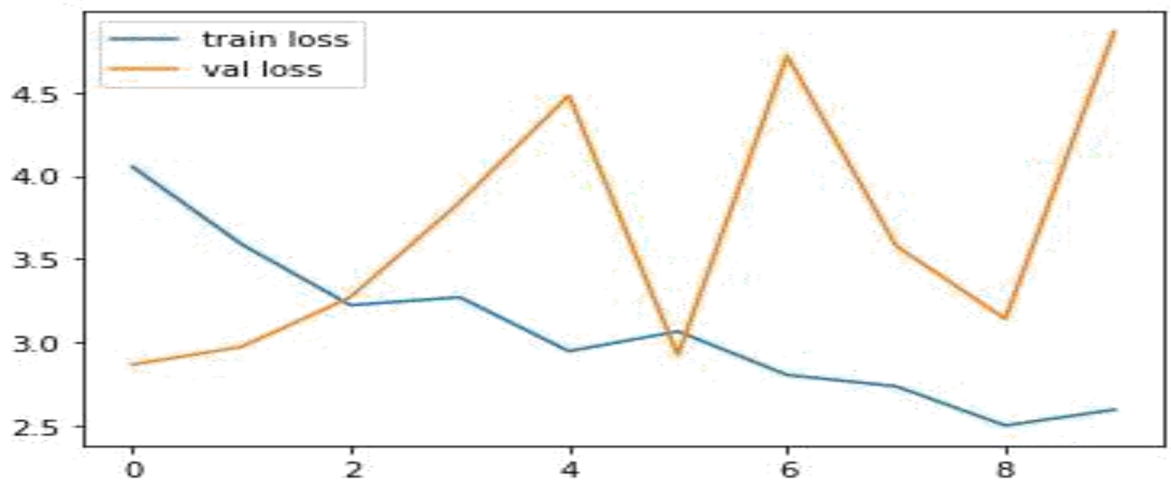


Figure 10 Inception v3 Loss

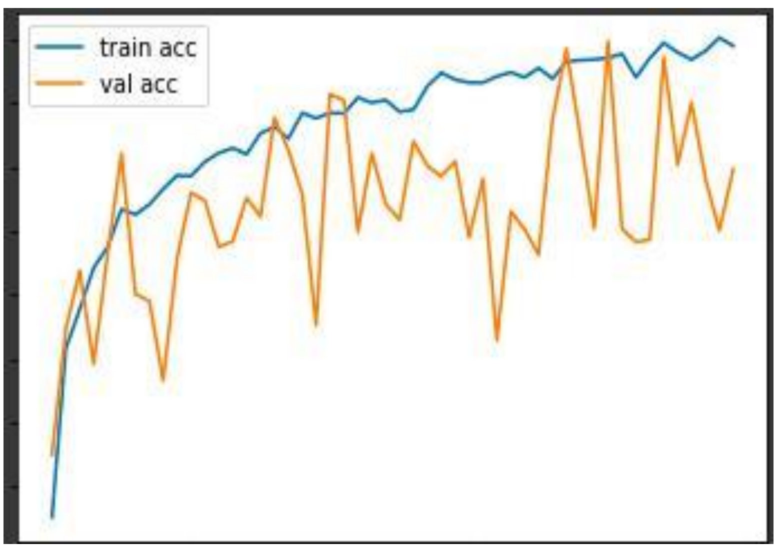


Figure 11 ResNet50 Accuracy

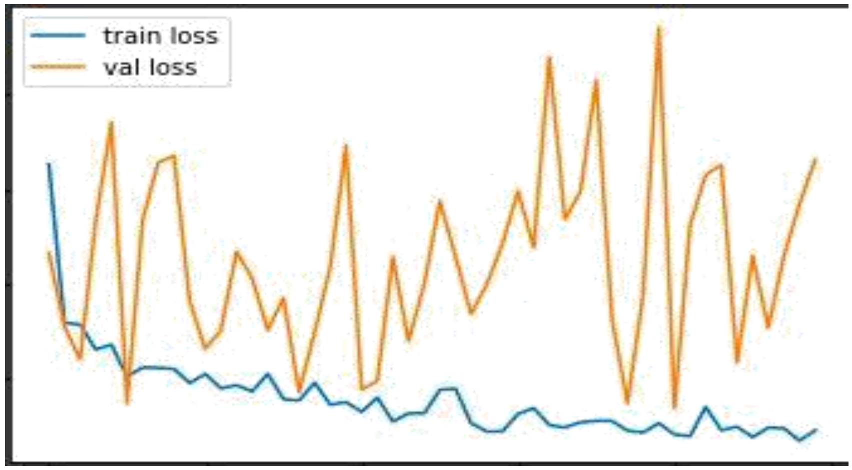


Figure 12 ResNet50 Loss

Chapter -5

CONCLUSION

5.1 Conclusion

In conclusion, the proposed detection models based neural networks are very effective in recognizing tomato leaf diseases with a some what similar accuracy for Inception v3 model and ResNet50 model.

To validate our approach, various cucumber leaves were picked and their images were taken using a smartphone. These images were then edited to remove the noise, or unnecessary background data, using image editing tools. The original images were resized from various different resolutions to a constant resolution of 512 x 512 and other colour spaces such as RGB and HSV. All these images were preprocessed and fed into our proposed algorithms. The images were then colour reduced using our proposed approach and Various features of the image based on its colour were extracted. The prediction of whether the leaf was diseased or healthy were then made using the results obtained.

5.2 Application

In the ever advancing world, the concept of smart farming has been introduced and is blooming. In smart farming, the field conditions are controlled and monitored using Internet Of Things and other hardware without human intervention. Recognizing the diseased plants from the healthy ones is the main purpose of the proposed approach. The self-recognition of diseases is an important aspect of smart farming. It is of great importance so that information about diseased plants could be quickly and precisely informed to the farmers and experts thus, reducing the need for monitoring large fields and also reducing the huge amount of manual labour required.

5.3 Limitation

Following are the limitations of our approach:

1. It might sometimes misclassify healthy leaves as diseased if the image is unclear or have only minor issues.
2. It does not account for the background noise, which had to be removed manually thus, it still requires some manual work.
3. The decision parameters were decided on a hit-and-trial basis and may require some more fine-tuning.

5.4 Future Work

The future work that can improve our proposed work can be**:**

1. Will implement our Own CNN Model, Rather Than Using Internal Models, then Compare it With ResNet and Inception V3 Models.
2. Figuring out a way to account for the background noise.
3. Comparing with other models.
4. It can be observed that better results can be obtained using several other techniques, so exploring them as well.
5. Try to figure out a way for detecting plant diseases using Object Detection rather than Classification models.

REFERENCE

[1] R.Chaerani and R. E.Voorrips, Tomato early blight (Alternaria solani): The pathogen, genetics, and breeding for resistance, Journal of General Plant Pathology, vol. 72, no. 6, pp. 335347, 2006.

[2] A. M. Dickey, L. S. Osborne, and C. L. Mckenzie, Papaya (Carica papaya, Brassicales: Caricaceae) is not a host plant of tomato yellow leaf curl virus (TYLCV; family Geminiviridae, genus Begomovirus), Florida Entomologist, vol. 95, no. 1, pp. 211213, 2012.

[3] G. Wei, L. Baoju, S. Yanxia, and X. Xuewen, Studies on pathogenicity differentiation of corynespora cassiicola isolates, against cucumber, tomato and eggplant, Acta Horticulturae Sinica, vol. 38, no. 3, pp. 465470, 2011.

[4] P. Lindhout, W. Korta, M. Cislik, I. Vos, and T. Gerlagh, Further identification of races of Cladosporiumfulvum(Fulvia fulva) on tomato originating from the Netherlands France and Poland, Netherlands Journal of Plant Pathology, vol. 95, no. 3, pp. 143148, 1989.

[5] K. Kubota, S. Tsuda, A. Tamai, and T. Meshi, Tomato mosaic virus replication protein suppresses virus-targeted posttranscriptional gene silencing, Journal of Virology, vol. 77, no. 20, pp. 1101611026, 2003.

[6] M. Tian, B. Benedetti, and S. Kamoun, A second Kazallike protease inhibitor from Phytophthora infestans inhibits and interacts with the apoplastic pathogenesis-related protease P69B of tomato, Plant Physiology, vol. 138, no. 3, pp. 17851793, 2005.

[7] L. E. Blum, Reduction of incidence and severity of Septoria lycopersici leaf spot of tomatowith bacteria and yeasts, Ciencia Rural, vol. 30, no. 5, pp. 761765, 2000.

[8] E. A. Chatzivasileiadis and M.W. Sabelis, Toxicity of methyl ketones from tomato trichomes to Tetranychus urticae Koch,Experimental and Applied Acarology, vol. 21, no. 6-7, pp. 473484, 1997.

[9] Smita Naikwadi, NiketAmoda, Advances in image processing for detection of plant diseases, International journal of application or innovation in engineering and managrment(IJAIEM) Volume 2,Issue11,November 2013

[10] K. Zhang, Q. Wu, A. Liu and X. Meng, Artists, Can Deep Learning Identify Tomato Leaf Disease?. [Art]. Northeast Agricultural University, Harbin 150030, China, 2018.

[11] S. Minaee, A. Abdolrashidi, and Y. Wang, ”Iris recognition using scattering transform and textural features,” in Signal Processing and Signal Processing Education Workshop (SP/SPE), 2015 IEEE, 2015, pp. 37-42.

[12] S. Minaee, A. Abdolrashidi, and Y. Wang, ”Face Recognition Using Scattering Convolutional Network,” arXiv preprint arXiv:1608.00059, 2016.

[13] J. Zhang, G. Li and S. He, ”Texture-Based Image Retrieval by Edge Detection Matching GLCM,” 2008 10th IEEE International Conference on High Performance Computing and Communications, Dalian, 2008, pp. 782-786.

[14] Wu, Xiaosheng and Sun, Junding, ”The Visual Computer:Joint-scale LBP: a new feature descriptor for texture classification”, vol. 33, March. 2017.

[15] T. Song, H. Li, F. Meng, Q. Wu and J. Cai, ”LETRIST: Locally Encoded Transform Feature Histogram for Rotation-Invariant Texture Classification,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 7, pp. 1565-1579, July 2018.

[16] S. Murala, R. P. Maheshwari and R. Balasubramanian, ”Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval,” in IEEE Transactions on Image Processing, vol. 21, no. 5, pp. 2874- 2886, May 2012.

[17] D. T. Nguyen, Z. Zong, P. Ogunbona and W. Li, ”Object detection using Non-Redundant Local Binary Patterns,” 2010 IEEE International Conference on Image Processing, Hong Kong, 2010, pp. 4609-4612.

[18] S. Gu, L. Ding, Y. Yang and X. Chen, ”A new deep learning method based on AlexNet model and SSD model for tennis ball recognition,” 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA), Hiroshima, 2017, pp. 159-164.

[19] H. Qassim, A. Verma and D. Feinzimer, ”Compressed residual-VGG16 CNN model for big data places image recognition,” 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 169-175.

[20] T. Carvalho, E. R. S. de Rezende, M. T. P. Alves, F. K. C. Balieiro and R. B. Sovat, ”Exposing Computer Generated Images by Eyes Region Classification via Transfer Learning of VGG19 CNN,” 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, 2017, pp. 866-870.

[21] P. Soucy and G. W. Mineau, ”A simple KNN algorithm for text categorization,” Proceedings 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 2001, pp. 647-648.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097-1105.

[23] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg and D. P. Hughes,”Deep Learning for Image-Based Cassava Disease Detection”, frontiers in Plant Science, 27 october 2017. [Online]. Available: https://doi.org/10.3389/fpls.201701852.

[24] S. Kumar, B. Sharma, V. K. Sharma, H. Sharma and J. C. Bansal, ”ScienceDirect: Sustainable Computing,” 27 October 2018. [Online]. Available: https://doi.org/10.1016/j.suscom.201810004.

[25] S. Zhang, X. Wu, Z. You and L. Zhang, ”ScienceDirect:Computers and Electronics in Agriculture,” March 2017. [Online]. Available: https://doi.org/10.1016/j.compag.201701014.

[26] A. Krizhevsky, et al., ImageNet classification with deep convolutional neuralnetworks, in: International Conference on Neural Information ProcessingSystems, 2012, pp. 10971105.

[27] B. A. M. Ashqar and S. S. Abu-Naser, Artists, Image-Based Tomato Leaves Diseases Detection Using Deep Learning. [Art]. Al-Azhar University, Gaza, Palestine, 2018.

[28] M. Hasan, B. Tanawala and K. J. Patel, ”Deep Learning Precision Farming: Tomato Leaf Disease Detection by Transfer Learning,” 2nd ICACSE-2019, 2019.

[29] C. Xie, Y. Shao, X. Li and Y. He, ”Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging,” SCIENTIFIC REPORTS, 2015.

[30] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, ”Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective,” IEEE Access, 2017.

[31] Ms. Kiran, R. Gavhale, Prof. Ujwalla Gawande, An Overview of the Research on Plant Leaves Disease detection using Image Processing Technique IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, PP 10-16, 2014.