“APPLICATION FOR PLANT DISEASE DETECTION”

Project Work Phase-II

(PPW402)

Submitted in Partial Fulfillment of the Requirements for the Award of

The Degree of

B.Tech

(Computer Science & Engineering)

Batch: 2017-21

SUBMITTED TO

HIMALAYAN SCHOOL OF SCIENCE & TECHNOLOGY

SWAMI RAMA HIMALAYAN UNIVERSITY

Swami Ram Nagar, Jolly Grant, Doiwala, Dehradun– 248016

Under the Guidance of:

Mr. Yogesh Pant
Assistant Professor
Computer Science and Engineering

Submitted by:

Sumit Tiwari
(DD1711303024)

DECLARATION

I hereby declare that this submission is my own work and that to the best of my knowledge and belief. It contains no material copied or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Sumit

Signature
Name: Sumit Tiwari
Registration No.: DD1711303024
Date: 09/July/2021

CERTIFICATE

This is to certify that Project Report entitled ***"Application for Plant Disease Detection"*** which is submitted by Sumit Tiwari in partial fulfillment of the requirement for the award of degree B.Tech in Computer Science and Engineering department of Swami Rama Himalayan University is a record of the candidate's own work carried out by him/her under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: 09/July/2021

Guide Name & Signature

Mr. Yogesh Pant
Assistant Professor (CSE)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Assistant Professor Yogesh Pant, Department of Computer Science and Engineering, Himalayan School of Science and Technology, Swami Rama Himalayan University for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We would like to give special thanks to our college dean Dr. R.C Ramola for his valuable support.

We also take the opportunity to acknowledge the contribution of Dr. Bineet Joshi (C.O.D) HSET, Department of Computer Science and Engineering, Himalayan School of Science and Technology, Swami Rama Himalayan University for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Sumit

Signature
Name: Sumit Tiwari
Registration No.: DD1711303024
Date: 09/July/2021

ABSTRACT

Identification of the plant diseases is that the key to stop the losses inside the yield and amount of the agricultural product. The studies of the plant diseases mean the studies of visually evident patterns seen on the plant. Health watching and illness detection on plant is unbelievably important for property agriculture. it is very troublesome to look at the plant diseases manually. It needs tremendous quantity of labor, experience inside the plant diseases, and conjointly need the excessive interval. Hence, image process is used for the detection of plant diseases by capturing the images of the leaves and comparison it with the information sets. the data set encompass totally different plant inside the image format. with the exception of detection users ar directed to Associate in Nursing e-commerce web site wherever totally different pesticides with its rate and usage directions ar displayed. This web site ar usually with efficiency used for comparison the MRP's of varied pesticides and obtain the required one for the detected illness. This paper aims to support Associate in Nursingd facilitate the inexperienced house farmers in an economical means. Keywords: illness detection, Tensor flow, inexperienced house, Convolution neural network, Data model, image to computer memory unit code I.

TABLE OF CONTENTS

\

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Project Name

Application for Plant Disease Detection

1.2 Project Brief

The project presents an android application designed to detect the diseases in leaf of a plant.

1.3 Introduction

India may be a cultivated country and concerning seventieth of the Population depends on agriculture. Farmers have largerange of diversity for selecting numerous appropriate crops and finding the suitable pesticides for plant. Hence, harm to the crops would end in large loss in productivity and would ultimately have an effect on the economy. Leaves being the foremost sensitive a locality of plants show illness symptoms at the earliest. The crops should be monitored against diseases from the terribly 1st stage of their life-cycle to the time they are ready to be harvested. Initially, the manoeuvre accustomed monitor the plants from diseases was the quality eye observation that is a long technique that desires specialists to manually monitor the crop fields. inside the recent years, variety of techniques ar applied to develop automatic and semi-automatic illness detection systems and automatic detection of the diseases by simply seeing the symptoms on the plant leaves makes it easier yet as cheaper. These systems have to this point resulted to be quick, cheap and a lot of correct than the quality technique of manual observation by farmers In most of the cases illness symptoms ar seen on the leaves, stem and fruit. The plant leaf for the detection of illness is taken under consideration that shows the illness symptoms. There ar several cases wherever farmers do not have a completely compact data concerning the crops and conjointly the illness which will get affected to the crops. This paper could also be effectively utilised by farmers thereby increasing the yield instead of visiting the skilled and obtaining their recommendation. the most objective is not solely to discover the illness mistreatment image process technologies. It conjointly directs the user on to Associate in Nursing e-commerce web site wherever the user should buy the medicine for the detected illness by comparison the rates and use befittingly in step with the directions given. Greenhouse conjointly referred to as a building, or, if with enough heating, a Hodr house, could also be a structure with walls and roof created mainly of clear material, like glass, inside that plants requiring regulated atmospheric condition ar fully grown. As greenhouse farming is gaining a lot of importance currently a day’s, this paper helps the greenhouse farmers in Associate in Nursing economical means. numerous techniques may be accustomed review the illness detection and discuss in terms of varied parameters. The paper is organized into the following sections. 1st section provides a quick introduction to the importance of illness detection. Second section discusses the current work disbursed recently throughout this space and conjointly reviews the techniques used. Section 3 includes methodologies utilized in our paper. Lastly, fourth section concludes this paper along side future directions.

1.4 Benefits with these applications

- An android application will prove to be a cheap solution for their plant detection requirements and it will help them keep the plants healthy and safe ensuring the health of both plants as well as people
- Be able to identify a disease and disease-causing agent,
- Be able to narrow the problem down to several possibilities which will require further study in the laboratory before he can make a final diagnosis, or
- Be completely baffled by the problem.

1.5 Languages and Platform Used

Machine Learning (Deep Learning and Neural Network)
Java
TensorFlow
Android Studio
Jupyter Notebook.

CHAPTER 2
LITERATURE REVIEW

2.1 Survey

Alternia leaf spot, brown spot, mosaic disease, gray spot and rust are 5 common types of apple diseases that seriously affect apple production. However, the current analysis lacks correct and rapid detection of apple diseases to ensure the healthy development of the apple trade. Object detection algorithms like SSD, DSSD and R-SSD may be considered consisting of 2 halfs: the primary part is that the pre-network model, that is employed as a basic options extractor. the opposite is associate auxillary structure that utilizes multi-scale feature map for a detection[1]. A K-means segmentation is employed for partitioning the leaf image into four clusters mistreatment the square geometrician distances. methodology|the tactic|the strategy} applied for feature extraction is color Co-occurrence method for each color and texture features[4]. Finally, abuse classification is accomplished through algorithmic neural network detection and rule-based backpropagation methods. The system malady detection and classification accuracy was found to be around ninety three. variety of crop varieties specifically, fruit crops, vegetable crops, cereal crops and industrial crops to observe fungous diseases on plant leaves. completely different strategies are adopted for every kind of crop[5].For fruit crops, k-means agglomeration is that the segmentation methodology used,texture options are targeted on associated classified mistreatment ANN and nearest neighbour algorithms achieving an overall average accuracy of 90.6%. For vegetable crops, chan-vase technique used for segmentation, native binary patterns for texture feature extraction Associate in Nursingd SVM and nearest neighbour formula for classification achieving an overall average accuracy of 87.9%. The industrial crops are divided mistreatment grab-cut formula. ripple primarily based feature extraction has been adopted mistreatment Mahalnobis distance Associate in Nursingd PNN as classifiers with an overall average accuracy of 84.8%. The cereal crops are divided mistreatment kmeans cluster and smart edge detector. Colour, shape, texture, color texture and random rework options are extracted. SVM Associate in Nursingd nearest neighbour classifiers used obtaining an overall average accuracy of 83.6%.A chilly plant leaf image and processed to see the health standing of the chilly plant. Their technique is making certain that the Chemicals ought to apply to the morbid chilly plant solely. They uses the MATLAB for the feature extraction and image recognition. during this paper pre-processing is completed mistreatment the. Fourier filtering, edge detection and morphological operations. laptop vision extends the image process paradigm for object classification. Here photographic camera is employed for the image Capturing and LABVIEW code tool to create the GUI[7]. The FPGA and DSP primarily based system is developed and used for watching and management of plant diseases . The FPGA is employed to induce the sector plant image or video knowledge for watching and designation. The DSP TMS320DM642 is employed to method and cipher the video or image knowledge. The nRF24L01 single chip a pair of.4 GHz sender is employed for knowledge transfer. it's 2 knowledge compress and transmission technique to satisfy user’s totally different would like and uses multi-channel wireless communication to lower the full system price.

2.2 Common relatable terms

What is machine learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. If there is an equation and input given, it is very easy to find the output using the given equation for any input but if there is only input and output given then it is pretty hard to detect the equation or the output of any random input. This defines the concept of predicting the outputs and figure out the algorithm using some given samples of inputs and outputs. In ML input and output are simple predictors and what we want to predict respectively.

Datasets

There are numerous datasets available for all the type of entities in our surrounding. What dataset is? The dataset is basically the collection of different inputs and output based on different algorithms. In the dataset some of the images are labelled and some are unlabeled. These labelled and unlabeled images are responsible for different algorithms used for training machine. Those dataset in which images are labelled trained using techniques of supervised learning and for those having not proper label we use unsupervised learning techniques. The biggest dataset having nearly all kind of real-world objects in it is COCO dataset. This dataset has more that 330k images in it with 200k labelled images.

Supervised and Unsupervised learning

In Supervised learning, you train the machine using data which is well "labeled." It means some data is already tagged with the correct answer. A supervised learning algorithm learns from labeled training data, helps you to predict outcomes for unforeseen data. Successfully building, scaling, and deploying accurate supervised machine learning. Data science model takes time and technical expertise from a team of highly skilled data scientists. Moreover, Data scientist must rebuild models to make sure the insights given remains true until its data changes. Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabeled data. Unsupervised learning algorithms allow you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning deep learning and reinforcement learning methods.

2.3 Methods of Leaf Recognition

1. Feature Extraction: Feature define the behavior of an image; it plays a vital role in the field of image recognition. First, before extracting features, images are preprocessed. In the technique of preprocessing of an image thresholding, normalization etc. methods are being implemented. After preprocessing, the features extraction techniques are applied to extract the features that will be helpful in different applications like character recognition.

2. Neural Network Structure:

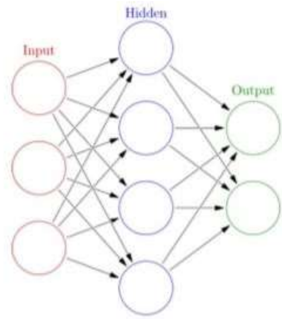


Fig 2.1 Neural Network Structure

- The red circle in the left represents the network's input layer where each circle represents an input neuron.
- The green circle in the rightmost contains the output neurons and are the output layer of the network.
- The middle layer can be described as hidden layer, which contains all the processing of the neural network.
- Circles here represents neurons. Each neuron here is a function which takes output neuron of previous layer and provides a number between 0 and 1.

2.4 Image Processing

Edge Detection: Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness.

Canny Edge Detection: In industry, the Canny edge detection technique is one of the standard edge detection techniques. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the image afterwards it applying the tendency to find the edges and the serious value for threshold.

Sobel Edge Detection The Sobel method of edge detection for image segmentation finds edges using the Sobel approximation to the derivative. It precedes the edges at those points where the gradient is highest. The Sobel technique performs a 2-D spatial gradient quantity on an image and so highlights regions of high spatial frequency that correspond to edges. The operator consists of a pair of 3x3 complication kernels:

-1	-2	-1
0	0	0
+1	+2	+1

G_x

-1	0	-1
-2	0	+2
-1	0	+1

G_y

Fig 2.2 Sobel Edge Detection

Image Contrast Stretching
Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, e.g. the full range of pixel values that the image type concerned allows.

Corner Detection
Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image.

2.5 Process involved for compiling machine

CNNs: Convolutional neural network. That is, a network which has at least one convolutional layer. A typical CNN also includes other types of layers, such as pooling layers and dense layers. **Convolution:** The process of applying a kernel (filter) to an image
Kernel / filter: A matrix which is smaller than the input, used to transform the input into chunks **Padding:** Adding pixels of some value, usually 0, around the input image
Pooling: The process of reducing the size of an image through down sampling. There are several types of pooling layers. For example, average pooling converts many values into a single value by taking the average. However, maxpooling is the most common.
Maxpooling: A pooling process in which many values are converted into a single value by taking the maximum value from among them.
Stride: the number of pixels to slide the kernel (filter) across the image.
Down sampling: The act of reducing the size of an image

Model Compilation
Compilation is very important step. It is always required after model definition. It includes both training from the scratch as well as loading a per-trained file and increase weights to train again. The parameters required for compilation process:
Regression: Mean Squared Error.
Binary Classification (for 2 classes): Cross Entropy or Binary Cross Entropy.
Multiclass Classification (for more than 2 classes): Categorical Cross Entropy.

Model Fit
Once the compilation is complete, the weights of training dataset are adapted. The network training is performed using backpropagation algorithm and is optimized using optimization function and loss functions was specified at the time of compilation. This backpropagation algorithm requires the network to be trained for particular epochs or rounds to the training dataset. Each epoch is grouped with input output patter pairs called batches. This defines the number of patterns that network has to go through. It is important for memory optimization as well; we must have to ensure that too many input patterns should not be loaded into memory simultaneously.

Network Evaluation
Now, when the model is trained and fit then is can be evaluated on the training dataset, but it will not reflect the useful predictions as all the data is know to the model. We can try evaluating the trained model on a different dataset which is unseen during testing. This can provide the overall performance of the network. The accuracy of the model can also be specified after applying the complied model on a different dataset.

Transfer Learning
It allows model to build in a time saving way rather than starting from the scratch. We will use our trained model to generate another model by re-training it to perform another task. Re-training needs less data to train a model instead of doing it from beginning. We have our own trained and compiled model; we have to convert it into tflite format.

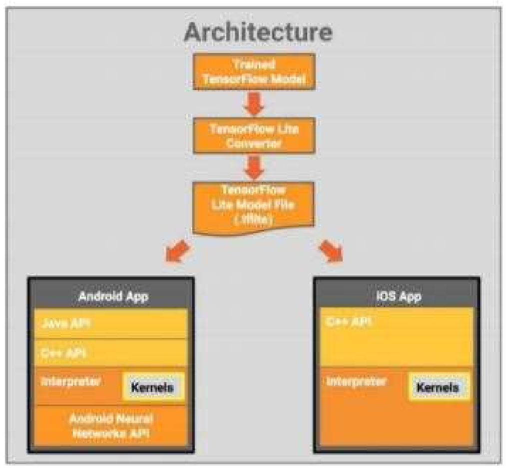


Fig 2.3 Transfer Learning

Convolution Neural Network
Layers used to build ConvNets:
A covnets is a sequence of layers, and every layer transform one volume to another through differentiable function.

2.6 Types of layers:

Let's take an example by running a covnets on of image of dimension 32 x 32 x 3. 1.

Input Layer: This layer holds the raw input of image with width 32, height 32 and depth 3. 2.
Convolution Layer: This layer computes the output volume by computing dot product between all filters and image patch. Suppose we use total 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12. 3.
Activation Function Layer: This layer will apply element wise activation function to the output of convolution layer. Some common activation functions are RELU: max(0, x),

Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12. 4.

Pool Layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents from overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

Fully-Connected Layer: This layer is regular neural network layer which takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

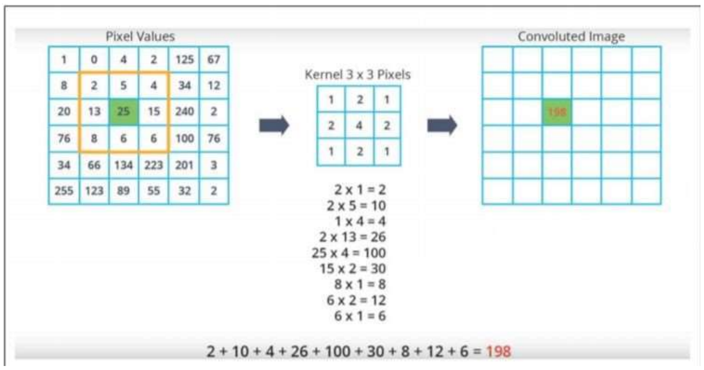


Fig 2.4 Convolution

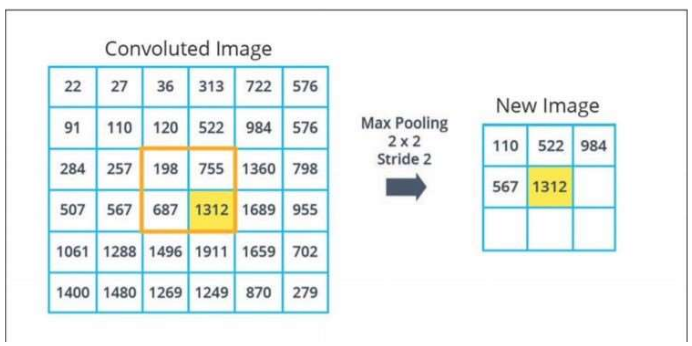


Fig 2.5 Max Pooling

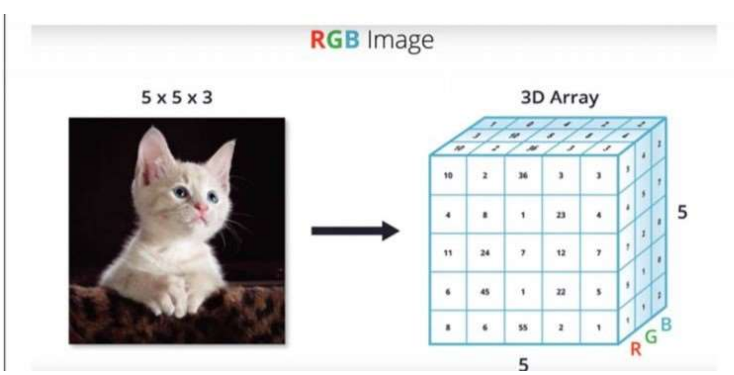


Fig 2.6 Convolution on colored image

Convolution on colored images:

In colored images we work on three-dimension array, height width and depth. Depth here will be determined with the number of color channels. Most color images are represented by three color channels namely Red Green Blue. In RGB images each color channel is represented by its own 2- dimensional array. Therefore, in the case of RGB images the depth of 3-dimensional array is just the stack of these 3 two-dimensional arrays.

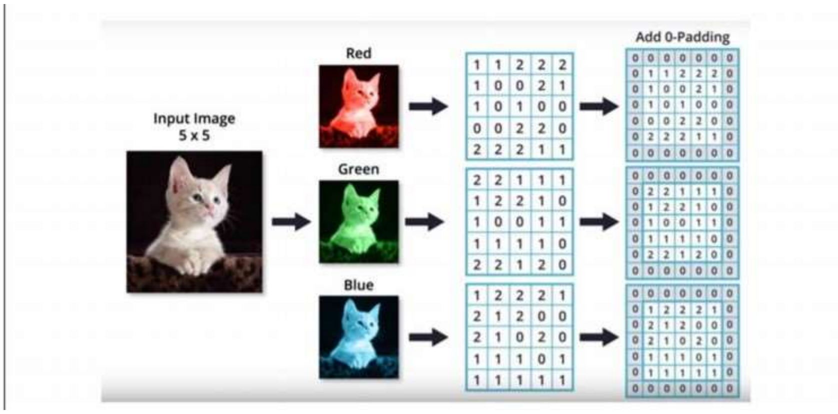


Fig 2.7 CNN on colored images

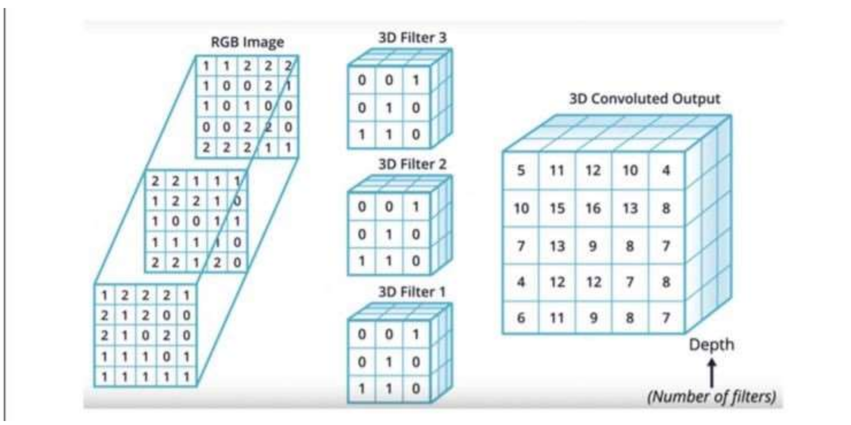


Fig 2.8 Convulated image

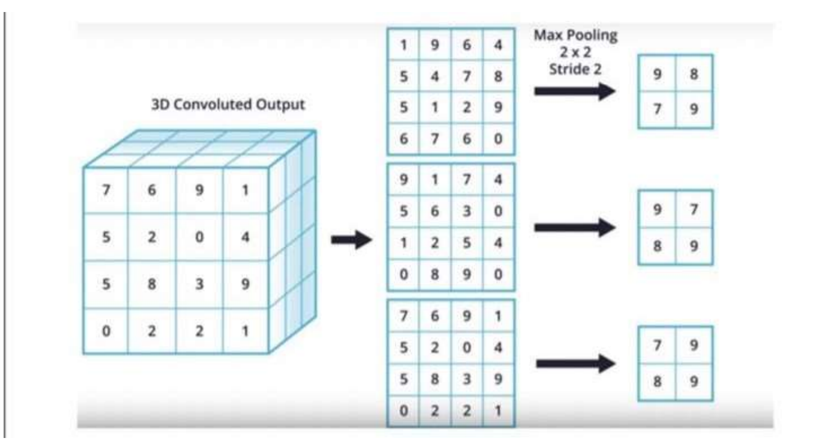


Fig 2.9 Max Pooling on colored image

2.7 Strategies of disease detection

The process of disease detection system primarily involves four phases as shown in Fig three.1. the primary part involves acquisition of pictures either through camera and mobile or from internet. The second part segments the image into varied numbers of clusters that completely different techniques will be applied. Next part contains feature extraction strategies and therefore the last part is regarding the classification of diseases.

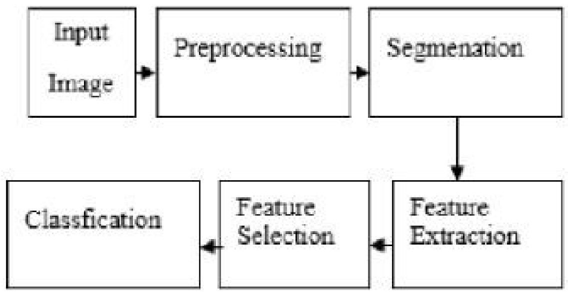


Fig 2.10- Phases of plant disease detection system

Image Acquisition

In this part, pictures of plant leaves area unit gathered victimisation digital media like camera, mobile phones etc. with desired resolution and size. the photographs may also be taken from internet. The formation of information of pictures is totally enthusiastic about the applying system developer. The image information is to blame for higher potency of the classifier within the last part of the detection system. Image Segmentation This part aims at simplifying the illustration of a picture specified it becomes additional significant and easier to investigate. because the premise of feature extraction, this part is additionally the basic approach of image process. There area unit varied strategies victimisation that pictures will be segmental like k-means agglomeration, Otsu's algorithmic rule and thresholding etc. The k-means agglomeration classifies objects or pixels supported a group of options into K variety of categories. The classification is completed by minimizing the add of squares of distances between the objects and their corresponding clusters.

Feature Extraction

Hence, during this step the options from this space of interest have to be compelled to be extracted. These options area unit needed ito verify the that means of a sample image.
Features will be supported color, shape, and texture. Recently, most of the researchers square measure aspiring to use texture options for detection of plant diseases. There square measure varied ways of feature extraction that may use for developing the system like gray-level co-occurrence matrix (GLCM), color cooccurrence technique, spacial greylevel idependence matrix, and bar graph based mostly feature extraction. The GLCM technique may be a method for texture classification.

Classification
The classification section implies to see if the input image is healthy or unhealthy. If the image is found to be unhealthy, some existing works have additional classified it into variety of diseases. For classification, a package routine is needed to be written in MATLAB, conjointly cited as classifier. variety of classifiers are utilized in the past few years by researchers like k-nearest neighbour (KNN), support vector machines (SVM), artificial neural network(ANN), back propagation neural network (BPNN), Naïve Bayes and call tree classifiers. the foremost ordinarily used classifier is found to be SVM. each classifier has its benefits and drawbacks, SVM is straightforward to use and sturdy technique.

2.8 Overview of disease

Plant diseases area unit usually caused by infectious agents like fungi, bacteria, and viruses. Signs of sickness|disease} area unit discernible proof of infection and symptoms area unit the visible effects of those types of disease. flora infections cause signs like visible spores, mildew, or mould and also the basic symptoms area unit like leaf spot and yellowing. flora diseases area unit plant infections caused by fungi. Fungi may be single or cellular, however either manner infect plants by stealing nutrients and breaking down tissue. flora diseases area unit the foremost common infection in plants. There area unit some characteristic symptoms, or discernible effects of the illness, in plants. Fungi infections may be recognized by symptoms like spots on plant leaves, yellowing of leaves, and birdseye spots on berries. With some flora diseases, the organism itself will really be viewed on the leaves seem as a growth and as a mould,

Fig 2.11- Leaf affected by fungal infection

These might a malformations on stems or the bottom of leaves. These direct observations of the disease-causing organism area unit known as signs of infection bacterium area unit acellular, prokaryotic organisms. bacterium area unit everyplace and lots of may be useful, however some will cause illness each in humans and plants. The signs of bacterium area unit typically more durable to notice than fungi, since bacterium area unit microscopic. Upon cutting Associate in Nursing infected stem, a opaque white substance might seem, known as microorganism ooze. this can be one sign of a microorganism infection. different signs embody water-soaked lesions, that area unit wet spots on leaves that ooze bacterium. Eventually, because the illness progresses, the lesions enlarge and kind achromatic spots on the leaves. a standard symptom of microorganism infection is leaf spots or fruit spots. in contrast to flora spots, these area unit typically contained by veins on the leaf.

Fig 2.12- Leaf affected by bacteria

Viruses ar infectious particles that ar too tiny to be detected by a light-weight magnifier. They invade host cells and hijack host machinery to force the host to create innumerable copies of the virus. infectious agent diseases do not show any signs in plants since viruses themselves can not be seen even with a light-weight magnifier. However, there ar symptoms that the trained eye will observe. A mosaic leaf pattern, yellowed, or wavelike leaves ar all characteristic of virus infection. This classic pattern of discoloration is wherever several plant viruses get their name, like the mosaic virus. Also, cut plant growth is additionally usually seen in infectious agent infections.

Fig 2.13- Leaf affected by virus

So, these ar our observation on the way to classify the varied disease and the way to use caution this.

2.9 Planned system

Proposed system have Associate in Nursing end-to-end automaton application with TFLite. planned system opted to develop Associate in Nursing automaton application that detects plant diseases. it's the algorithms and models to acknowledge species and diseases within the crop leaves by victimisation Convolutional Neural Network. planned system use Colab to edit ASCII text file. A dataset of fifty four,305 pictures of unhealthy and healthy plant leaves collected below controlled conditions Plant Village dataset. the pictures cowl fourteen species of crops, including: apple, blueberry, cherry, grape, orange, peach, pepper, potato, raspberry, soy, squash, strawberry and tomato. It contains pictures of seventeen basic diseases, four microorganism diseases, two diseases caused by mould (oomycete), two infectious agent illnesss and one disease caused by a mite. twelve crop species even have healthy leaf pictures that aren't visibly tormented by disease. Our dataset contains solutions for many plant textures like,

1. Apple___Apple_scab
2. Apple___Black_rot
3. Apple___Cedar_apple_rust
4. Apple___healthy
5. Blueberry___healthy
6. Cherry_(including_sour)___Powdery_mildew
7. Cherry_(including_sour)___healthy
8. Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
9. Corn_(maize)___Common_rust_
10. Corn_(maize)___Northern_Leaf_Blight
11. Corn_(maize)___healthy
12. Grape___Black_rot
13. Grape___Esca_(Black_Measles)
14. Grape___Leaf_blight_(Isariopsis_Leaf_Spot)
15. Grape___healthy
16. Mango_Unhealthy
17. Mango__Healthy
18. Orange___Haunglongbing_(Citrus_greening)
19. Peach___Bacterial_spot
20. Peach___healthy
21. Pepper,_bell___Bacterial_spot
22. Pepper,_bell___healthy
23. Potato___Early_blight
24. Potato___Late_blight
25. Potato___healthy
26. Raspberry___healthy
27. Soybean___healthy
28. Squash___Powdery_mildew
29. Strawberry___Leaf_scorch
30. Strawberry___healthy
31. Tomato___Bacterial_spot
32. Tomato___Early_blight
33. Tomato___Late_blight
34. Tomato___Leaf_Mold
35. Tomato___Septoria_leaf_spot
36. Tomato___Spider_mites Two-spotted_spider_mite
37. Tomato___Target_Spot
38. Tomato___Tomato_Yellow_Leaf_Curl_Virus
39. Tomato___Tomato_mosaic_virus
40. Tomato___healthy

41. background
42. fresh cotton leaf
43. fresh cotton plant
44. fusarium wilt diseased cotton leaf
45. fusarium wilt diseased cotton plant
46. mango_Anthracnose
47. mango_healthy

Data generators which will browse footage in our supply folders, convert them to `float32` tensors, and feed them (with their labels) to our network is about up. As information that goes into neural networks ought to typically be normalized in a way to form it additional amenable to process by the network. In our case, we are going to pre-process our pictures by normalizing the constituent values to be within the `[0, 1]` vary (originally all values area unit within the `[0, 255]` range). we are going to got to ensure the input file is resized to 224x224 pixels or 299x299 pixels as needed by the networks. you have got the selection to implement image augmentation or not.

Apart from simply police work the disease victimization the on top of ways our system directs the user to associate ecommerce web site. This web site displays all the pesticides that area unit on the market for the detected malady with its MRP rate. in conjunction with this the directions to use it's conjointly on the market within the web site. so by scrutiny the speed and options of the pesticides the user should buy it

CHAPTER 3
REQUIREMENTS OF THE PROJECT

3.1 Minimum software & hardware requirements for development of the project

- **Software and platform required for the development of the project:**
  - Anaconda distribution.
    - Python 3.8.2 with the Jupyter notebook framework for model coding.
  - Android Studio
    - Android development platform to build native section of the android application. TensorFlow and Keras
    - Keras' neural network functions will be used to train the model and TensorFlow numerical computation makes machine learning faster and easier.

- **Hardware requirements for the development of the project:**
  - GPU power to train the model
    - During model training it need a high processing power may be a GPU or CPU. TensorFlow and Keras can use GPU to process.
  - Android phone for the user
    - Final output will be an android application, so a device which can run apk file.

3.2 Minimum software & hardware requirements for running the project

- **Minimum software requirements for the application:**
  - Minimum software version is KitKat, i.e., Android Version > 4.4
  - Camera to api support

- **Minimum hardware requirements for the application**:
  - RAM recommended is minimum 2GB
  - Camera module

CHAPTER 4
PROCESS MODEL

4.1 Model Name

The Software Development Life Cycle (SDLC) model used in this project is **Iterative V – Model**

4.2 Iterative V – Model

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

- **Verification:** It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

- **Validation:** It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements. So, V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phase in V-shape. Thus, it is called V-Model.

- **Design Phase:**
  - **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.
  - **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.
  - **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.
  - **Module Design:** In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

- **Testing Phases:**
  - **Unit Testing:** Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.
  - **Integration testing:** After completion of unit testing Integration testing is performed. In integration testing, the modules are integrated and the system is tested. Integration

testing is performed on the Architecture design phase. This test verifies the communication of modules among themselves.

- **System Testing:** System testing test the complete application with its functionality, interdependency, and communication. It tests the functional and non-functional requirements of the developed application.
- **User Acceptance Testing (UAT):** UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.

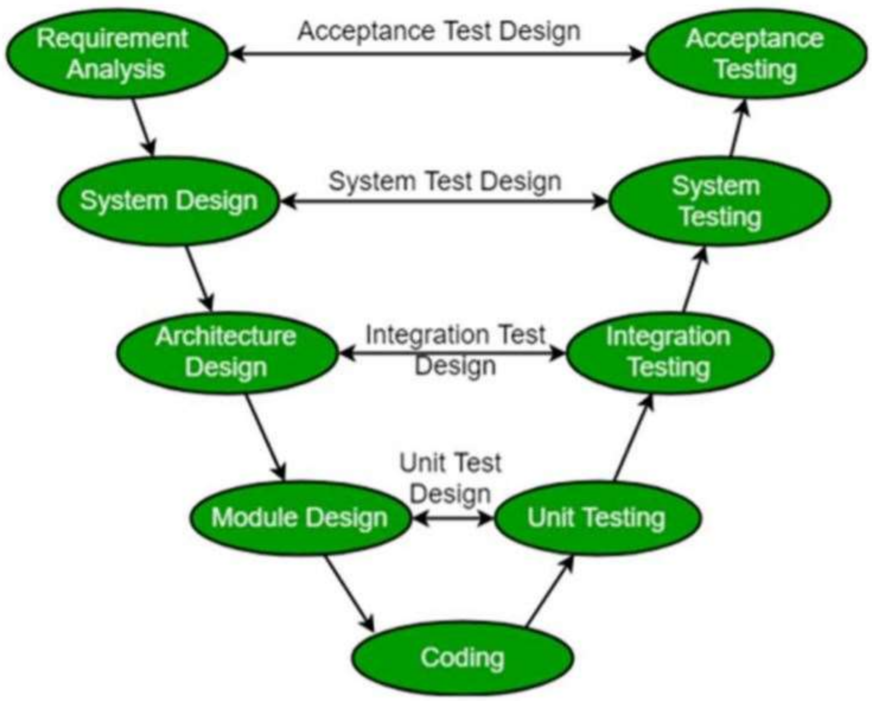


Fig 4.1- Itterative V-model

CHAPTER 5

MODULES IN THE PROJECT

5.1 Module 1: Home module:

This module will contain the description of project and list of all the leaves used in project. This section will be pure native and use the android studio inbuilt functions to design the whole activity.

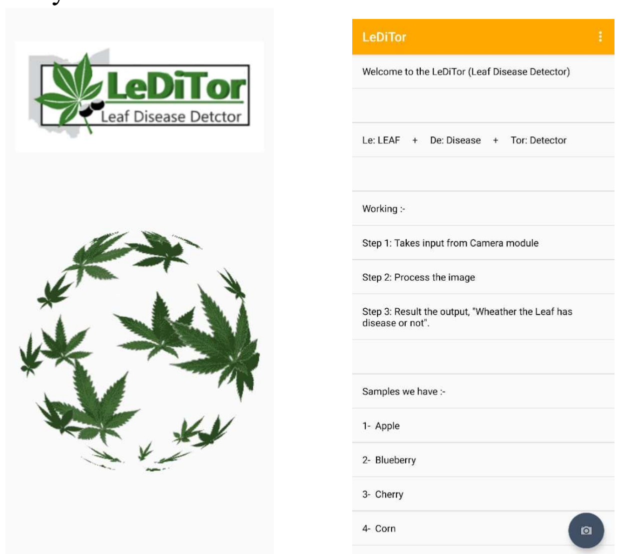


Fig 5.1- Home Module

5.2 Module 2: Camera / Upload

This module will provide the user to click the photo of leaf or the plant to get the details related to disease. The prediction of the image will use the concept of machine learning and image processing. In the process of prediction, the very first step is to convert the image into the size on which our machine is trained for the dataset. Then the image is set to its color mode according to the trained dataset. Then the image will be pass on to the machine (trained dataset code) and it will return the image class as the output of the image.

Fig 5.2- Camera Module

CHAPTER 6
RESULTS AND DISCUSSION

There 2 completely different conditions for coaching and testing. One is below the science lab conditions, which implies that the model is tested with the photographs from constant dataset from that it's used for each coaching and testing. the opposite condition is that field condition; this suggests that our model has tested with the photographs taken from the $64000 world conditions (land). Since the lighting conditions and background properties of the photographs ar whole completely different after we take samples from the $64000 field, there's an opportunity that our model to provide a awfully low accuracy, once scrutiny to the accuracy values noninheritable throughout the science lab conditions. thus to beat this impact, we tend to had a plan of getting a mixed style of pictures throughout the coaching section (heterogeneity).

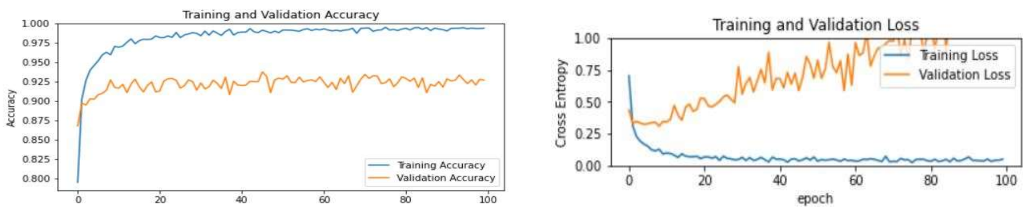


Fig 6.1 PB Model Accuracy and Loss

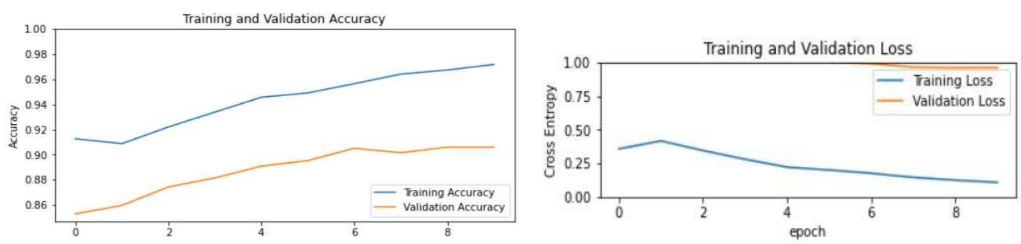


Fig 6.2 TF Model Accuracy and Loss

The accuracy of period detection of apple plant disease victimisation deep learning approach supported improved convolution neural networks is a smaller amount compared to the planned system as a result of it detects multiple diseases in an exceedingly single system.

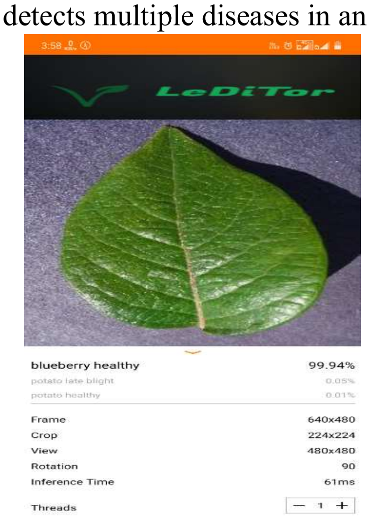


Fig 6.3 Application sample

CHAPTER 7
CONCLUSION

7.1 Conclusion: Thus associate application engineered for the identification of unwellness affected plants and healthy plants is finished and this planned work is focuses on the accuracy values throughout the $64000 field conditions, and this work is enforced by having many disease pictures.
Overall this work is enforced from scratch and produces a good accuracy. the long run work is to extend the amount of pictures gift within the predefined info and to change the design in accordance with the dataset for achieving higher accuracy

7.2 Future Scope:
A country's economy is very much dependent on its agricultural productivity. Thus, the detection of diseases in plants hold an importance for the product quality and productivity. Sometimes, these diseases are hazardous, so the application can be further extended by implementing database which store details of the diseased plant and how it can be cured, also a module which suggest the early symptoms so that the farmers can prevent there crops from disease and loss can be minimized.

REFERENCES

[1] Peng Jiang , Yuehan Chen ,Bin Liu , Dongjian He , Chunquan Liang ,' period Detection of Apple Leaf Diseases mistreatment Deep Learning Approach supported Improved Convolutional Neural Networks', (Volume: seven), pp. 06 might 2019

[2] Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., Shimizu, M., 'Disease detection of Cercospora Leaf Spot in sugar beet by strong templet matching', Computers and physical science in Agriculture, Volume 108, pp.58-70, 2014

[3] Barbedo, J.G.A., Godoy, C.V., 'Automatic Classification of Soybean Diseases supported Digital pictures of Leaf Symptoms', SBI AGRO,2015

[4] Barbedo, J.G.A., 'A review on the most challenges in automatic disease identification supported visible vary images', 2016 ,Biosystems Engineering, Volume 144, pp. 52-60.

[5] Bashish, D.A., Braik, M., Ahmad, S.B., 'A Framework for Detection and Classification of Plant Leaf and Stem Diseases', International Conference on Signal and Image process, pp. 113-118, 2010

[6] Punajari, J.D., Yakkundimath, R., Byadgi, A.S., 'Image process based mostly Detection of fungous Diseases In Plants', International Conference on data and Communication Technologies, Volume 46, pp. 1802-1808, 2015

[7] Zulkifli Bin Husin, Abdul Hallis Bin Abdul Aziz, Ali Yeon Bin Md Shakaff Rohani Binti S Mohamed Farook, "Feasibility Study on Plant Chili unwellness Detection mistreatment Image process Techniques", 2012 Third International Conference on Intelligent Systems Modelling and Simulation.

[8] Chunxia Zhang, Xiuqing Wang, Xudong Li, "Design of observation and management disease System supported DSP&FPGA", 2010 Second International Conference on Networks Security, Wireless Communications and TrustedComputing.

[9] Omrani, E., Khoshnevisan, B., Shamshirband, S., Saboohi, H., Anuar, N.B., Nasir, M.H.N., 'Potential of radial basis functionbased support vector regression for apple unwellness detection', Journal of measuring, pp. 233-252, 2014

[10] Gharge, S., Singh, P., 'Image process for Soybean unwellness Classification and Severity Estimation', rising analysis in Computing, data, Communication and Applications, pp. 493-500, 2016

APPENDIX I

Datasets Samples

Apple Scrap Dataset

Apple Healthy Dataset

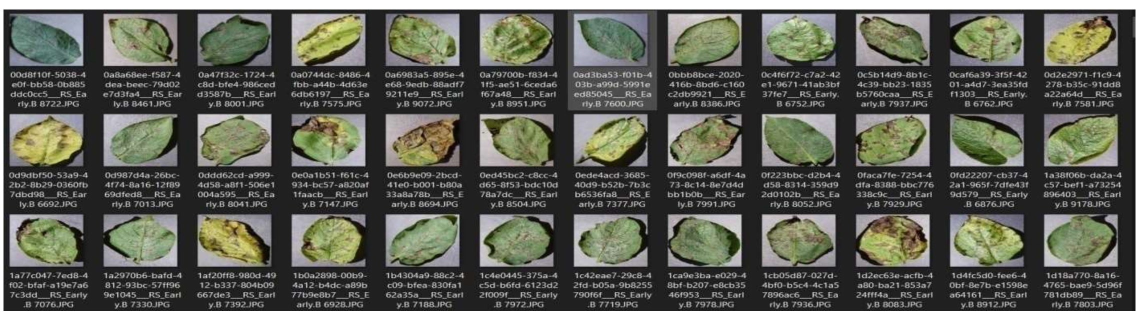
Cotton Wilt

Cotton Healthy

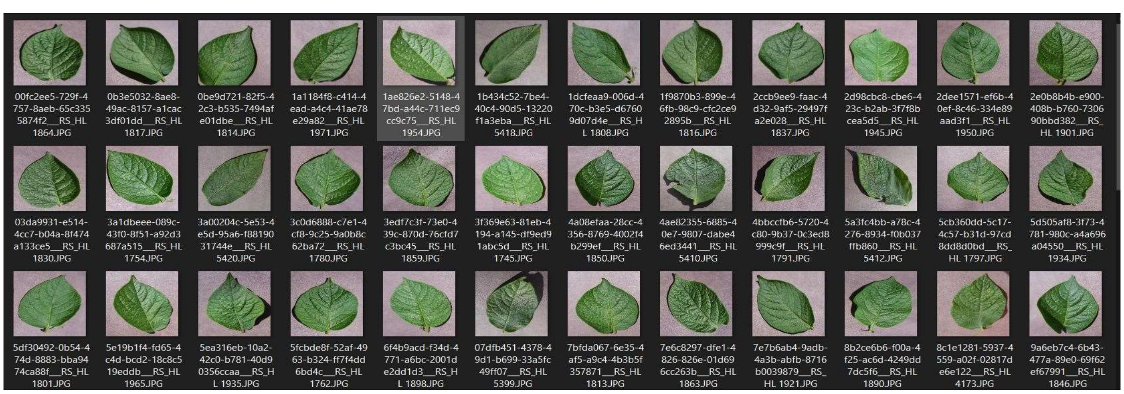
Mango Healthy

Mango Unhealthy

Potato Blight

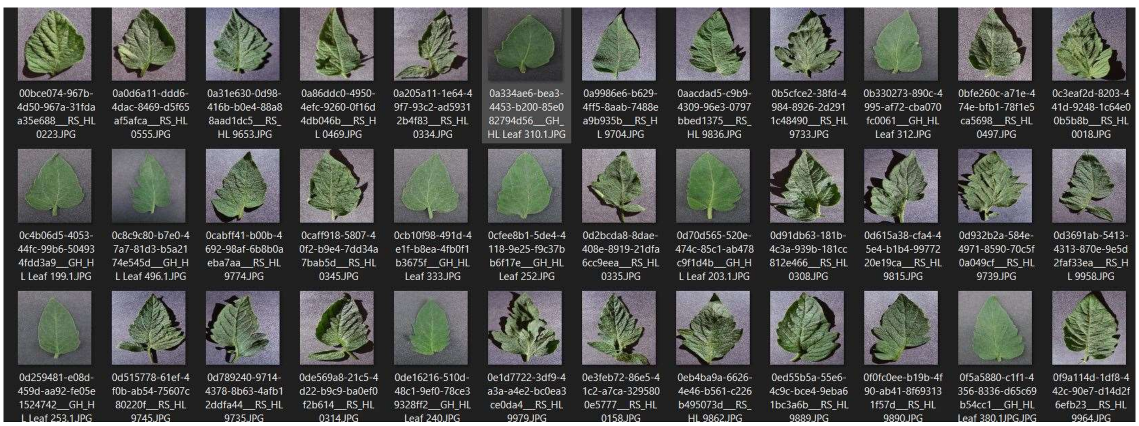


Potato Healthy

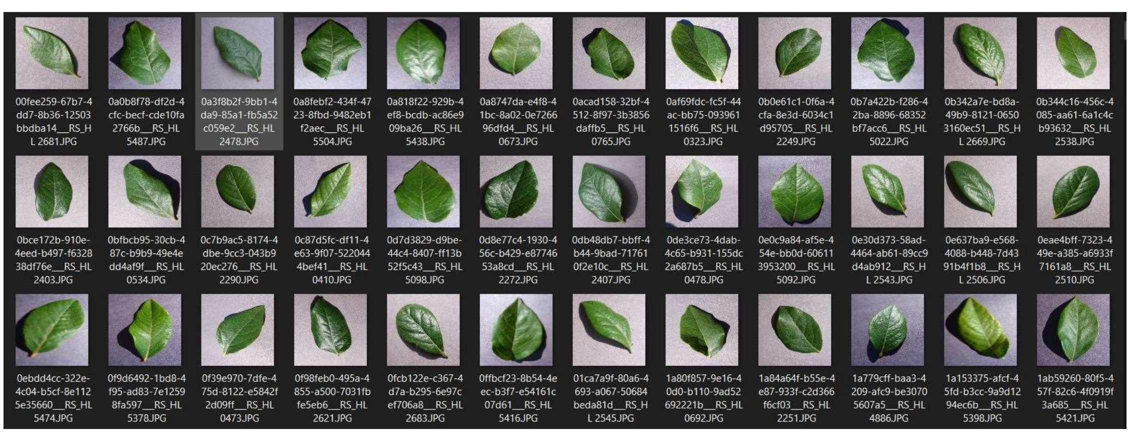


Tamato Blight

Tamato Healthy



Blueberry Healthy



MODEL TRAINING SAMPLES

Pb model training

```
epochs = 100
```

```
history = model.fit_generator(train_generator,
                    epochs=epochs,
                    validation_data=val_generator)
```

```
C:\Anaconda\lib\site-packages\tensorflow\python\keras\engine\training.py:1844: UserWarning: `Model.fit_generator` is deprecated
and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

```
Epoch 1/100
580/580 [==============================] - 430s 726ms/step - loss: 1.1893 - accuracy: 0.6748 - val_loss: 0.4365 - val_accuracy:
0.8682
Epoch 2/100
580/580 [==============================] - 231s 399ms/step - loss: 0.3250 - accuracy: 0.8967 - val_loss: 0.3302 - val_accuracy:
0.8976
Epoch 3/100
580/580 [==============================] - 210s 362ms/step - loss: 0.2264 - accuracy: 0.9260 - val_loss: 0.3467 - val_accuracy:
0.8949
Epoch 4/100
580/580 [==============================] - 207s 356ms/step - loss: 0.1783 - accuracy: 0.9440 - val_loss: 0.3345 - val_accuracy:
0.9029
Epoch 5/100
580/580 [==============================] - 211s 363ms/step - loss: 0.1595 - accuracy: 0.9490 - val_loss: 0.3230 - val_accuracy:
0.9025
Epoch 6/100
580/580 [==============================] - 212s 365ms/step - loss: 0.1364 - accuracy: 0.9554 - val_loss: 0.3284 - val_accuracy:
0.9083
Epoch 7/100
580/580 [==============================] - 213s 366ms/step - loss: 0.1154 - accuracy: 0.9629 - val_loss: 0.3366 - val_accuracy:
0.9101
Epoch 8/100
580/580 [==============================] - 215s 371ms/step - loss: 0.1080 - accuracy: 0.9650 - val_loss: 0.3403 - val_accuracy:
0.9141
Epoch 9/100
580/580 [==============================] - 217s 373ms/step - loss: 0.1245 - accuracy: 0.9611 - val_loss: 0.3104 - val_accuracy:
0.9274
Epoch 10/100
580/580 [==============================] - 222s 379ms/step - loss: 0.0885 - accuracy: 0.9725 - val_loss: 0.3463 - val_accuracy:
0.9176
Epoch 11/100
580/580 [==============================] - 301s 518ms/step - loss: 0.0897 - accuracy: 0.9716 - val_loss: 0.3425 - val_accuracy:
0.9167
Epoch 12/100
580/580 [==============================] - 442s 761ms/step - loss: 0.0808 - accuracy: 0.9749 - val_loss: 0.3647 - val_accuracy:
0.9216
Epoch 13/100
580/580 [==============================] - 460s 793ms/step - loss: 0.0714 - accuracy: 0.9786 - val_loss: 0.4727 - val_accuracy:
0.9110
Epoch 14/100
580/580 [==============================] - 285s 490ms/step - loss: 0.0661 - accuracy: 0.9796 - val_loss: 0.3967 - val_accuracy:
0.9221
Epoch 15/100
580/580 [==============================] - 224s 386ms/step - loss: 0.0757 - accuracy: 0.9779 - val_loss: 0.3568 - val_accuracy:
0.9283
Epoch 16/100
580/580 [==============================] - 240s 413ms/step - loss: 0.0693 - accuracy: 0.9803 - val_loss: 0.4574 - val_accuracy:
```

TF model training

```
history_fine = model.fit_generator(train_generator,
                         epochs=10,
                         validation_data=val_generator)
```

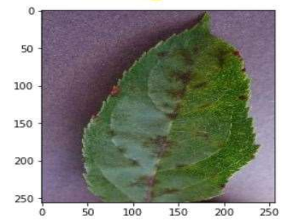
```
Epoch 1/10
580/580 [==============================] - 750s 1s/step - loss: 0.3570 - accuracy: 0.9127 - val_loss: 1.2482 - val_accuracy: 0.
8531
Epoch 2/10
580/580 [==============================] - 710s 1s/step - loss: 0.4169 - accuracy: 0.9089 - val_loss: 1.2483 - val_accuracy: 0.
8598
Epoch 3/10
580/580 [==============================] - 687s 1s/step - loss: 0.3450 - accuracy: 0.9221 - val_loss: 1.1932 - val_accuracy: 0.
8744
Epoch 4/10
580/580 [==============================] - 740s 1s/step - loss: 0.2803 - accuracy: 0.9340 - val_loss: 1.2029 - val_accuracy: 0.
8816
Epoch 5/10
580/580 [==============================] - 813s 1s/step - loss: 0.2214 - accuracy: 0.9457 - val_loss: 1.0797 - val_accuracy: 0.
8909
Epoch 6/10
580/580 [==============================] - 707s 1s/step - loss: 0.1999 - accuracy: 0.9492 - val_loss: 1.0034 - val_accuracy: 0.
8954
Epoch 7/10
580/580 [==============================] - 737s 1s/step - loss: 0.1762 - accuracy: 0.9565 - val_loss: 0.9911 - val_accuracy: 0.
9052
Epoch 8/10
580/580 [==============================] - 717s 1s/step - loss: 0.1451 - accuracy: 0.9641 - val_loss: 0.9655 - val_accuracy: 0.
9016
Epoch 9/10
580/580 [==============================] - 767s 1s/step - loss: 0.1247 - accuracy: 0.9674 - val_loss: 0.9616 - val_accuracy: 0.
9061
Epoch 10/10
580/580 [==============================] - 782s 1s/step - loss: 0.1083 - accuracy: 0.9718 - val_loss: 0.9623 - val_accuracy: 0.
9061
```

APPENDIX II

WEB APPLICATION SAMPLES

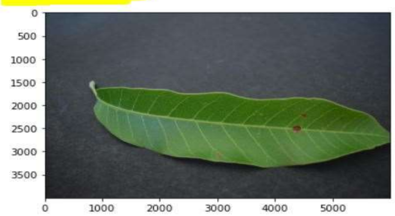
```
result2 = model.predict_classes([prepare('C:\\Users\\RoboRock\\Projects\\Leditor2\\Check\\Apple___Apple_scab.JPG')])
disease2=image.load_img('C:\\Users\\RoboRock\\Projects\\Leditor2\\Check\\Apple___Apple_scab.JPG')
plt.imshow(disease2)
print (Classes[int(result2)])
```

```
Apple___Apple_scab
```




```
result42 = model.predict_classes([prepare('C:\\Users\\RoboRock\\Projects\\Leditor2\\Check\\mango_Anthracnose.JPG')])
disease42=image.load_img('C:\\Users\\RoboRock\\Projects\\Leditor2\\Check\\mango_Anthracnose.JPG')
plt.imshow(disease42)
print (Classes[int(result42)])
```

```
mango_Anthracnose
```



ANDROID APPLICATION SAMPLE OUTPUT