Image Based Plant Disease Detection Using Neural Network

A

Project Report

Submitted in partial fulfillment of the
Requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

With Specialization in
Oil and Gas Informatics
&
Gaming and Graphics

BY: -

NAME	ROLL NO
ROHIT SINGH YADAV	R970217033
SAHIL TALREJA	R970217034
VIVEL RAJ	R970217048
ANIMESH SINGH	R142217002

Under the guidance of
Dr. RAVI TOMAR,
Assistant Professor (SG),
Department of Informatics

UPES
UNIVERSITY WITH A PURPOSE

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun-248007 2020-21

CANDIDATE’S DECLARATION

We hereby certify that the project work entitled “**Image Based Plant Disease Detection Using Neural Network**” in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Oil And Gas Informatics and Gaming and Graphics submitted to the Department of Informatics at School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **AUGUST,2020 to DECEMBER ,2020** under the supervision of **Dr. RAVI TOMAR** (Assistant Professor(SG) Department of Informatics).

The matter presented in this project has been submitted by us for the award of any other degree of this University **VIVEK RAJ (R970217048) Rohit SINGH YADAV (R970217033), SAHIL TALREJA(R970217034), ANIMESH SINGH(R142217002)**
This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 13/12/2020

Dr. RAVI TOMAR
(Project Guide)

Dr. Thipendra Pal Singh
Professor and H.O.D
Department of Informatics
School of Computer Science
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

Table of Contents

List of Figures

Abstract

The proposed system helps in identification of plant disease and provides remedies that can be used as a defense mechanism against the disease. In Today's time Agriculture productivity has vast role in the economy of any country, this is one of the reasons that disease detection in the plants plays an important role in the agriculture field. In this field you can't take proper care of any plant then it creates one kind of series effects on the plants and due to which respective product quantity, quality or productivity is affected. Detection of plant disease through some advanced computer vision technology is called automatic technique it is very beneficial to reduce the large work of monitoring in big farm of any plants and at very early stage camera can detects the symptoms of diseases that is when they appear on plant leaves. The latest generation of convolutional neural networks (CNNs) has highly effective results in the field of image classification. All essential steps required for implementing this disease recognition model like variety of neural-wise and layer-wise visualization methods were applied using CNN and trained with publicly available plant diseases dataset and after generate the attention maps for identify several layers that were not contributing to inference and remove such layers from the network and decreasing the number of parameters by 75% without affecting the classification accuracy.

KEYWORDS: - Image processing, Convolutional Neural Network, Plant disease Detection, classification, Test-Database, Accuracy, Confidence.

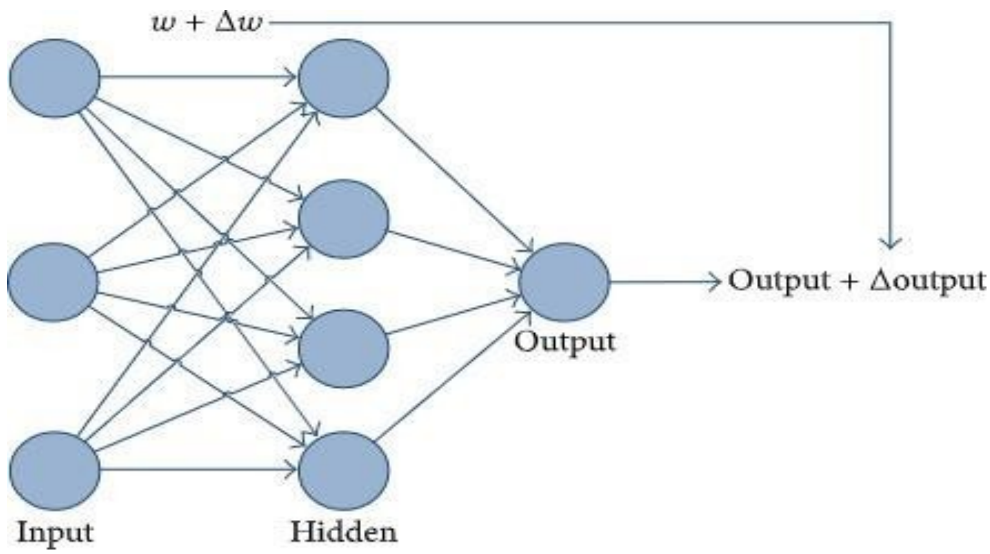


Figure 1: - Simple model of ANN.

Introduction

Plant diseases from the long time is one of the major threats for the food security because it dramatically reduces the crop yield and compromises its quality. Traditionally identification of plants diseases depends upon the faith of human annotation by visual inspection. Now a days there are many perspectives for humans to detects the plant diseases. Digital Images required lots of CPU and RAM for training a model that is extremely difficult for computer to process it required a preprocessing step to extract a certain feature like (color, shape). In such situations, deep learning is typically used because it allows the computer to learn the most suitable feature without human intervention. An initial attempt to use deep learning for image-based plant disease diagnosis was reported in 2016 where the trained model was able to classify 14 crops and 26 diseases with an accuracy of 99.35% against optical images. Since then successive generations of deep-learning-based disease diagnosis in various crops have been reported.

There are several ways to detect plant pathologies. Some diseases do not have any visible symptoms or the effect becomes noticeable too late to act and in those situations, a sophisticated analysis is obligatory. However, most diseases generate some kind of manifestation in the visible spectrum, so the naked eye examination of a trained professional is the prime technique adopted in practice for plant disease detection. In order to achieve accurate plant disease diagnostics a plant pathologist should possess good observation skills so that one can identify characteristic symptoms. Variations in symptoms indicated by diseased plants may lead to an improper diagnosis since amateur gardeners and hobbyists could have more difficulties determining it than a professional plant pathologist. An automated system designed to help identify the plant diseases by the plant's appearance and visual symptoms could be of great help to amateurs in the gardening process and also trained professionals as a verification system in disease diagnostics.

Advances in computer vision present an opportunity to expand and enhance the practice of precise plant protection and extend the market of computer vision applications in the field of precision agriculture. Exploiting common digital image processing techniques such as color analysis and threshold were used with the aim of detection and classification of plant diseases. Various different approaches are currently used for detecting plant diseases and most common are artificial neural networks (ANNs) and Support Vector Machines (SVMs). They are combined with different methods of image pre-processing in favor of better feature extraction. Among various network architectures used in deep learning convolutional neural networks (CNN) are widely used in image recognition. The first CNNs the neocognitron and LeNet were introduced in the 1980s although the study of neural networks originally started in the 1940s. CNNs have been used for plant image analysis since the early days of their evolution. A major turning point for the CNNs was the introduction of AlexNet which significantly outperformed the image classification accuracy of traditional machine learning approaches in ImageNet Large Scale Visual Recognition Challenge (LSVRC) 2012.

CNNs consist of convolutional layers which are sets of image filters convoluted to images or feature maps along with other layers (pooling). In image classification, feature maps are extracted through convolution and other processing layers repetitively and the network eventually outputs a label indicating an estimated class. Given a training dataset CNN unlike traditional machine learning techniques that use hand-crafted features optimizes the weights and filter parameters in the hidden layers to generate features suitable to solve the classification problem. In principle, the parameters in the network are optimized by back- propagation and gradient descent approaches to minimize the classification error.

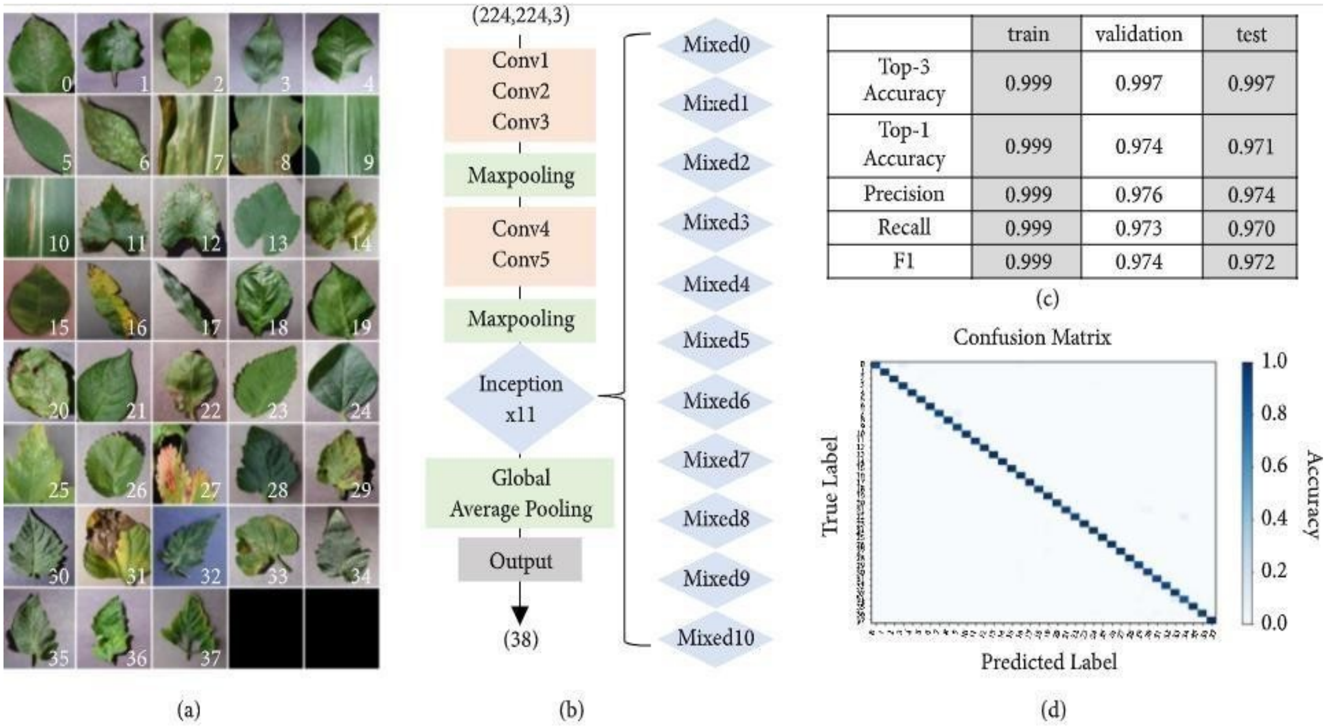


Figure 2: - Plant Village image dataset and training diagnosis.

Image based disease diagnosis training using convolutional neural networks

a) The Plant Village image dataset used in this study. This dataset contains 38 categories of diseased or healthy leaf images.
b) InceptionV3-based convolutional neural network (CNN) architecture used in this study. convolutional layer; Mixed, inception module.
c) Accuracy, precision, recall, and mean F1 scores against the training, validation and test data using the trained weights.
d) Confusion matrix drawn against the test dataset.

Problem Statement

Farming has always been a key player in GDP with a holding of more than 15.9%. Plant disease have always been a procrastinating threat for farming and indirectly for food security. As the detection has always been done through human eyes (in India from C.E) and most of the time it's too late which makes them not fit for this kind of detection which requires high persistence. So, to reduce this threat is to design a machine which can detect Visual plant diseases and reduce the continues monitoring job and time for the farmers.

Objective

Main objective is to identify the plant diseases using image processing.

- To design such system that can detect crop disease and pest accurately.
- Which can reduce the attack of pests by using proper pesticides and remedies.
- Give better accuracy of detection.

Flow chart

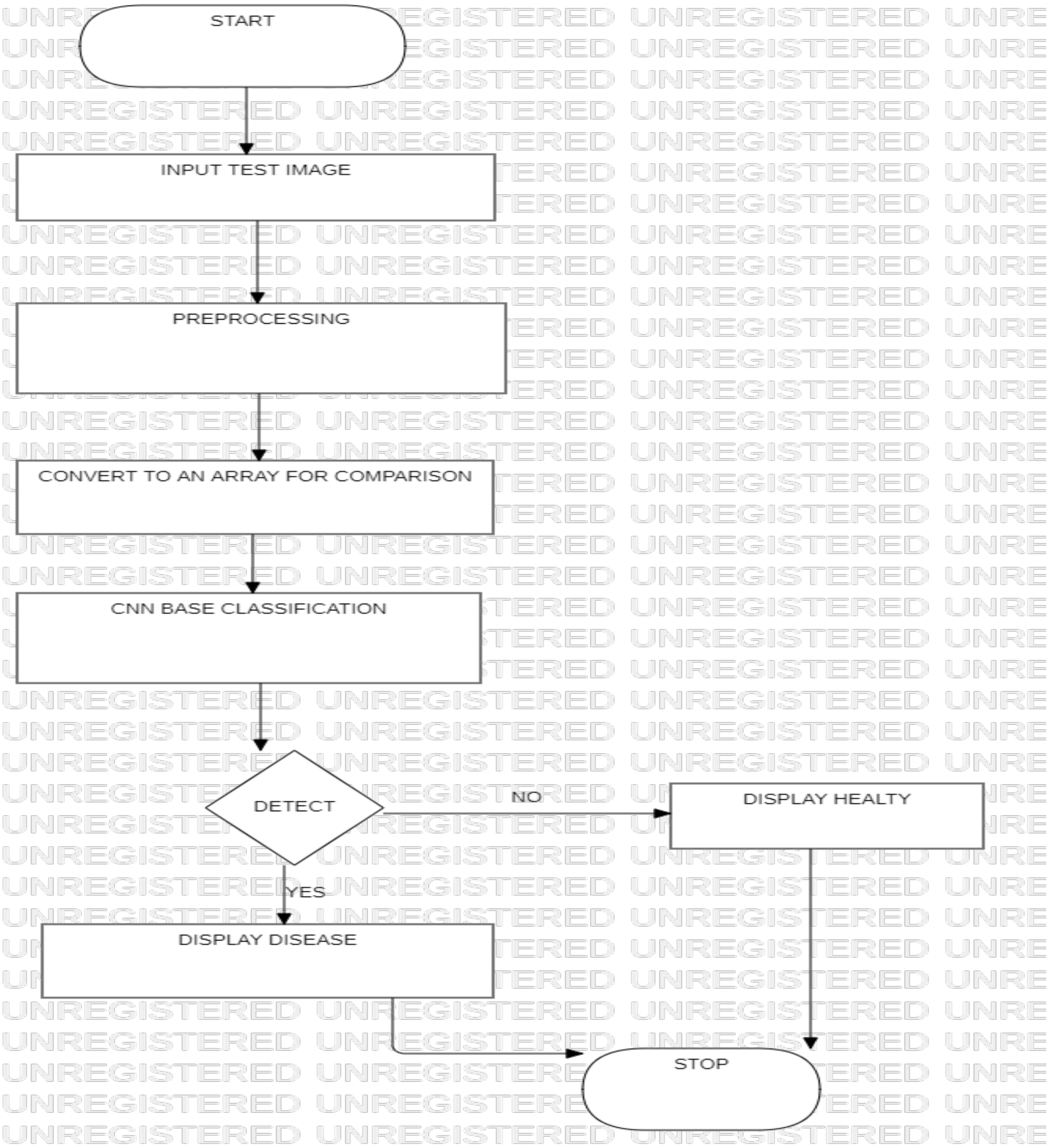


Figure 3: - Flow chart explaining the process of machine.

Flow chart general explanation

- The input test image is acquired and preprocessed in the next stage and then it is converted into array form for comparison.
- The model is properly trained using CNN and then classification takes place.
- The comparison of the test image and the trained model take place followed by the display of the result.
- If there is a defect or disease in the plant the software displays the disease along with the remedy.

Methodology

Feasible study phase

- Study of different distributed algorithms.
- Different types of plant images are studied and corresponding. After detail study, labeling in done by segregating the images and with different diseases

Requirement analysis

- The database is preprocessed such as Image reshaping, resizing and conversion to an array form.
- There is a huge database so basically the images with better resolution and angle are selected. After selection of images we should have deep knowledge about the different leaves and the disease they have. Huge research is done from plant village organization repository.

Design phase

- Designing algorithm and graphs for getting more accuracy and satisfaction.
- Adjusting the Images in categories so that our model identifies the diseases.

Implementation phase

- CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D and MaxPooling2D these all will implement.

Testing phase

- After the model is trained successfully the software can identify the disease if the plant species is contained in the database.
- After successful training and preprocessing, comparison of the test image and trained model takes place to predict the disease.

Deployment phase

- After follow all above steps then deploy the model on the Internet so that our client's or Famer's can use it for more production in the farming growth.

Diseases of different plants

- Apple black spot
- Apple broad leaf spot
- Apple needle leaf spot
- Apple normal
- Bell paper normal
- Blueberry normal
- Cherry normal
- Cherry powder normal
- Corn blight
- Corn rust

System requirements

Hardware requirements

- GPU - Quadro 2000 (CUDA enabled, Compute capability - 2.1)
- Processor - Intel® Xeon® CPU E5-2640 @ 2.50GHz
- RAM - 16 GB
- System type - 64-bit Windows 10 Education OS, x64-based processor

Software requirements

- Jupyter Notebook
- Any Operating Systems (Linux, Window, Mac etc.)

Pert Chart

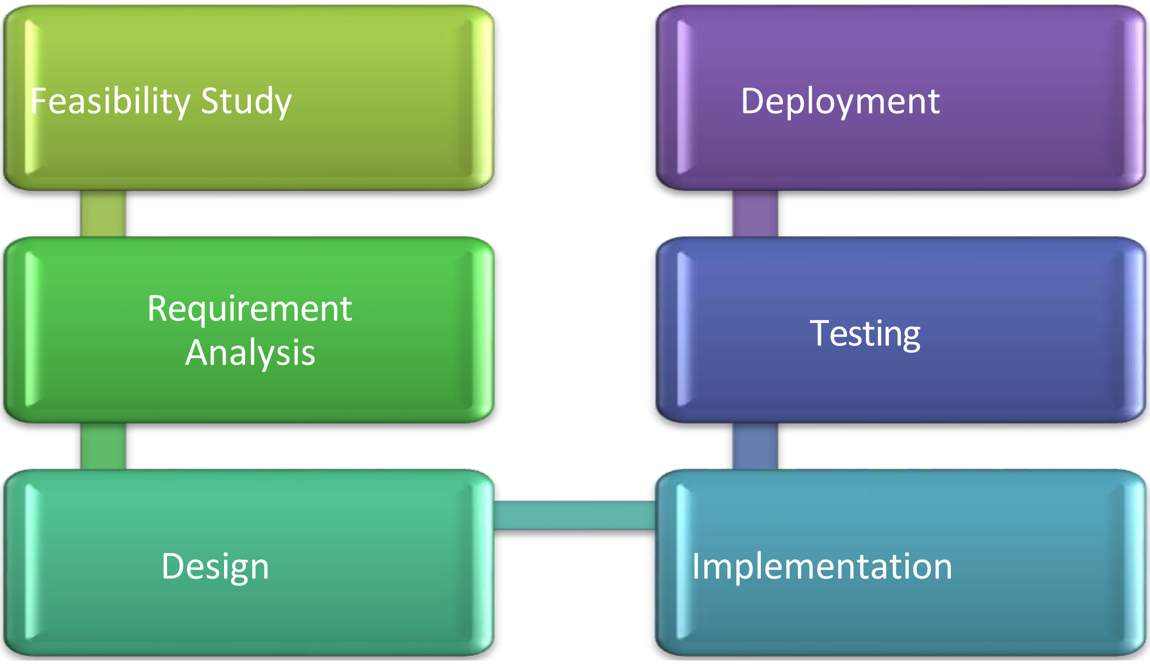


Figure 4: - Pert chart.

Implementation

To implement the idea of disease detection and to train the machine accordingly requires lot of steps which are mentioned below: -

⇒ Label Data for input like Training Data, Testing Data and Valid Data in different folders.

⇒ Import all libraries like NumPy, Pandas, Matplotlib, tensorflow, Convolution2D, MaxPooling2D, Flatten, Dense, Sequential Model, ImageDataGenerator, image, random etc.

⇒ Assign paths of the folders where training and testing data are available.

⇒ Assign Dictionary to write all the diseases name in a sequential way.

⇒ Add Convolutional Layer with 32 Filters each of filter size 3*3 and apply Relu activation function.

⇒ Add Maxpooling layer for extracting the features from convolutional layer.

⇒ Repeat Step3 and Step4.

⇒ Add Flatten Layer to convert 2D Array to 1D Array.

⇒ Add Hidden Layers or Dense Layers with 128 neurons and activation function is relu.

⇒ Add Output Layer with 24 neurons and Activation function Softmax.

⇒ Apply Compile function to compile all the layers with loss function categorical_crossentropy and optimizer='Adam'.

⇒ Model Fit Function is used to fit all variables like training set, steps per epoch=7234, epochs=2, validation data=test set etc.

⇒ Start Training, minimum time taken to train data in this dataset will be 2 hours.

⇒ After the model gets trained, apply path of the folder where valid images are available.

⇒ Apply Predict function to predict the valid image to get output.

While implementation of the following steps some acute procrastinated discussion took place, which have been mentioned in the discussion part.

DISCUSSION

In this project we had taken 7234 Images for training and 3328 Images for Testing, before putting this much data into the model, we applied 2 epochs which we have been declared numero uno. In first epoch, model will be trained through all 7234 images and same process will be started for 2^{nd} epoch. After model.fit() function will start to train the model. In First epoch the results will be in two variables like Loss and Accuracy.
Loss: 0.0675 and Accuracy: 0.8124 in first epoch.
After second epoch Loss: 0.0345 and Accuracy: 0.9432.

After this the model is saved in .h5 extension and then we import load_model library to load the model. We have now all valid images in a folder and then it is called randomly, these images are in PIL format but our predict function doesn't understand about PIL format it understand only array format in 4D so we need to call one function img_to_array() and then convert into 4D by using expand_dims() function.

This led to annexing of result which have been mentioned below.

RESULT

This led to output which will be in 1D array format, out of these 24 classes only one value will be higher and rest of the classes will be zero approximately. Higher value of the output will denote the class of the disease. For more clarification please consider images 9 – 12.
And then output will be: -
Category number: (1 to 24)
Leaf and disease: Name_of_disease.

Machine

Different sections used and designed in machine.

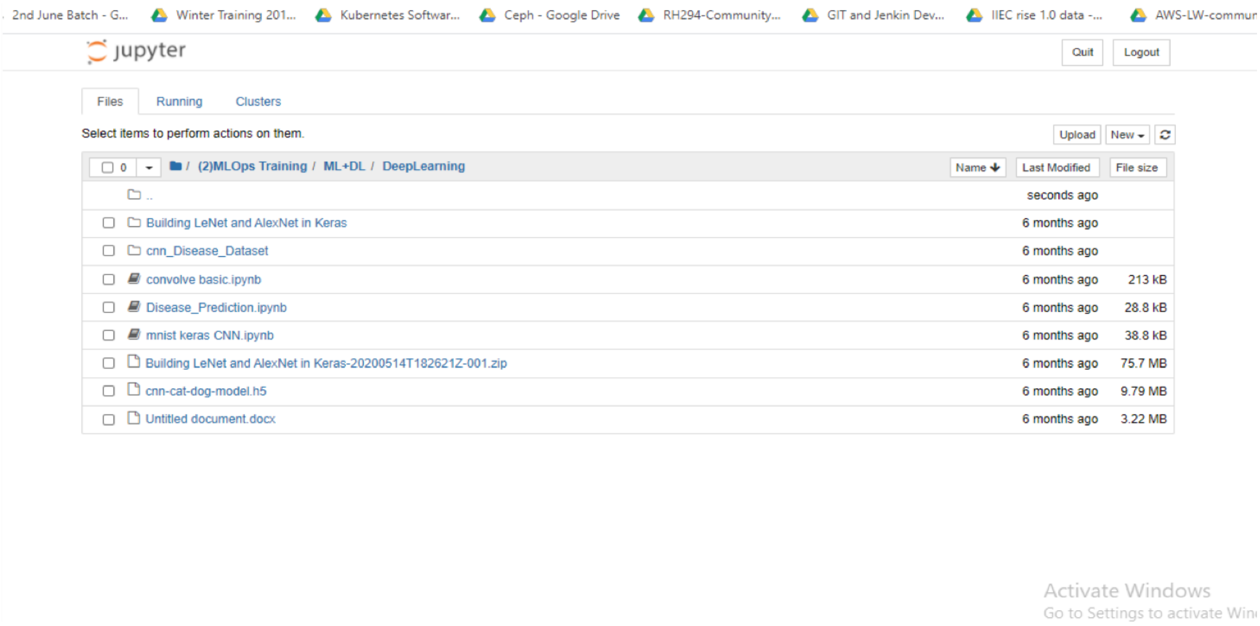


Figure 5: - Directories used in machine.

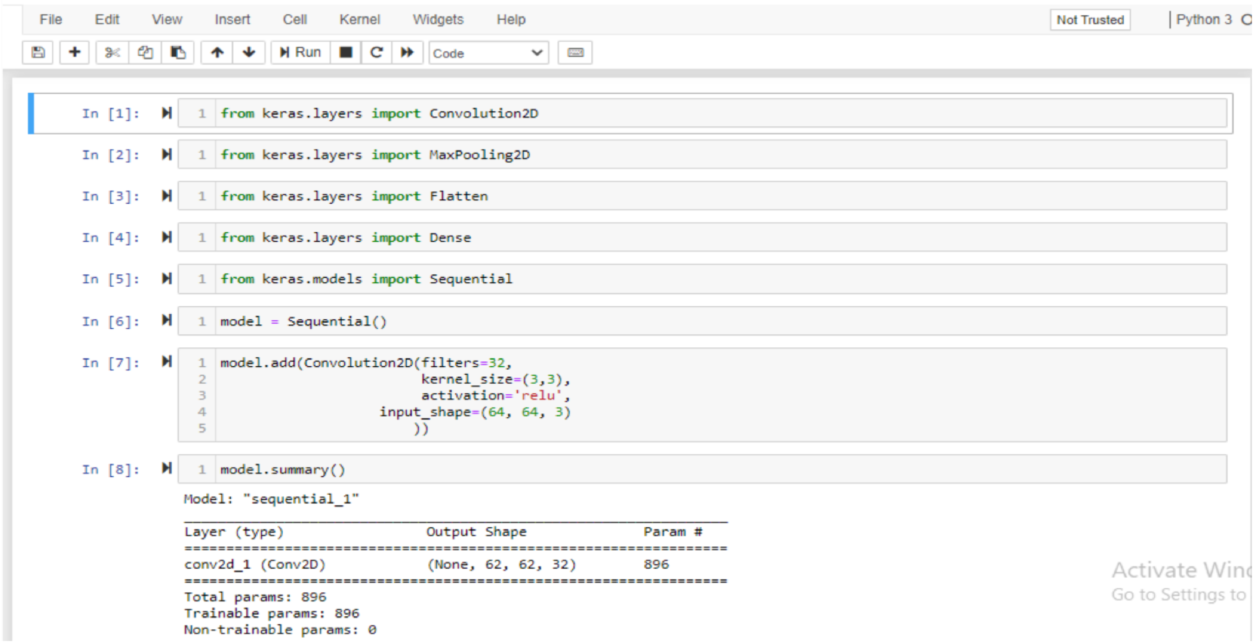
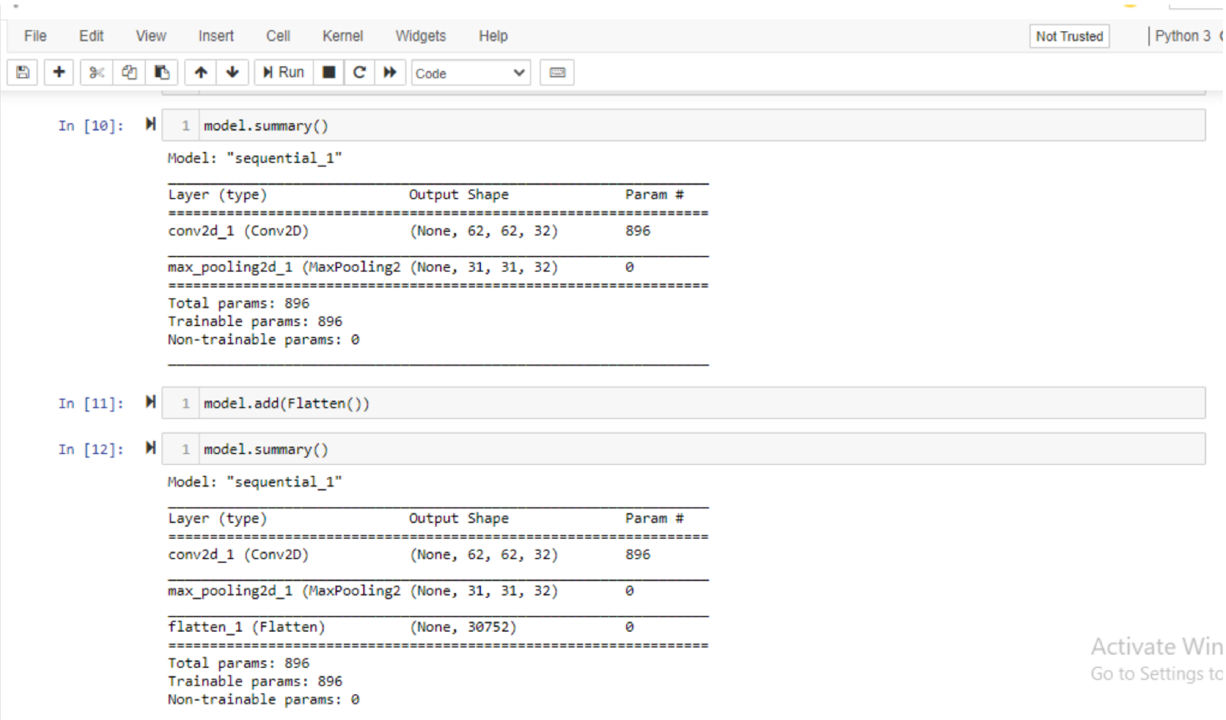
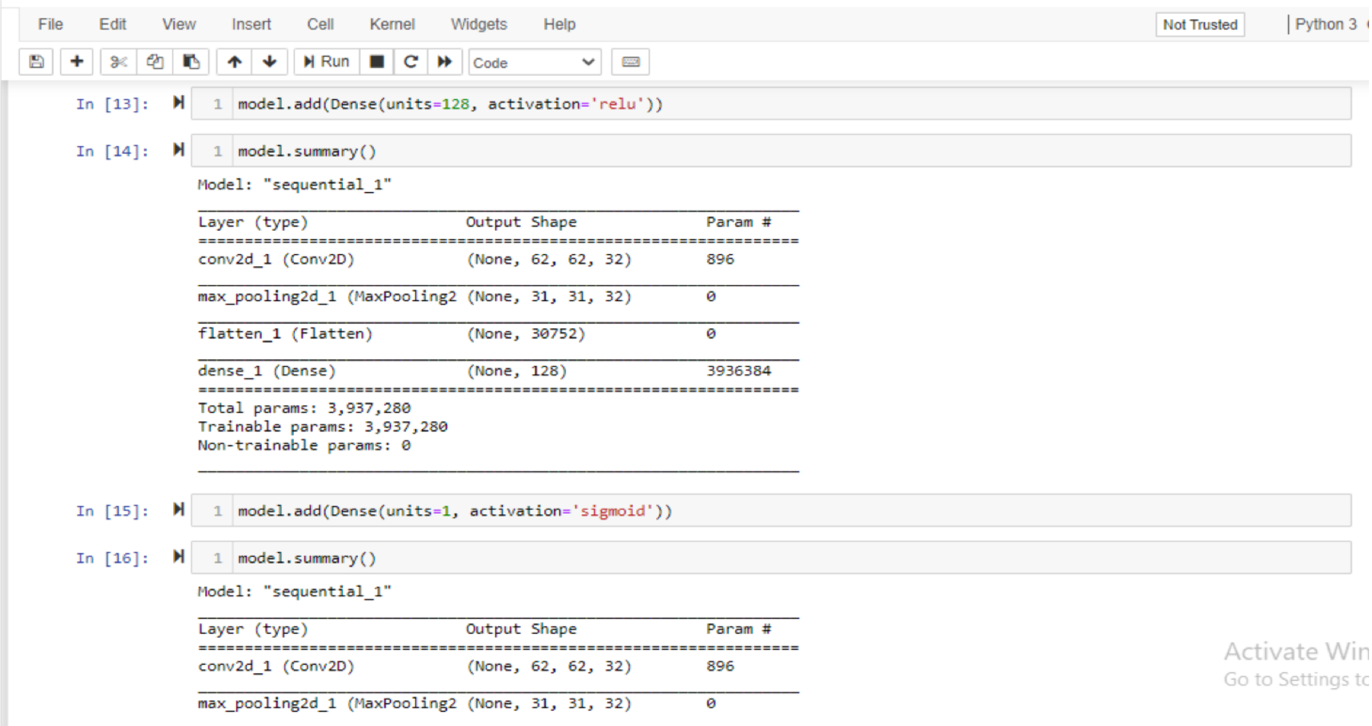


Figure 6: - Modules imported in machine.



```
File Edit View Insert Cell Kernel Widgets Help                Not Trusted | Python 3

In [10]:  model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________

In [11]:  model.add(Flatten())

In [12]:  model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 30752)             0
=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
```

Figure 7: - Implementation of CNN (first part) in machine.



```
File Edit View Insert Cell Kernel Widgets Help                Not Trusted | Python 3

In [13]:  model.add(Dense(units=128, activation='relu'))

In [14]:  model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 30752)             0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               3936384
=================================================================
Total params: 3,937,280
Trainable params: 3,937,280
Non-trainable params: 0
_________________________________________________________________

In [15]:  model.add(Dense(units=1, activation='sigmoid'))

In [16]:  model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
```

Figure 8: - Implementation of CNN (second part) in machine.

Predicted output from the machine

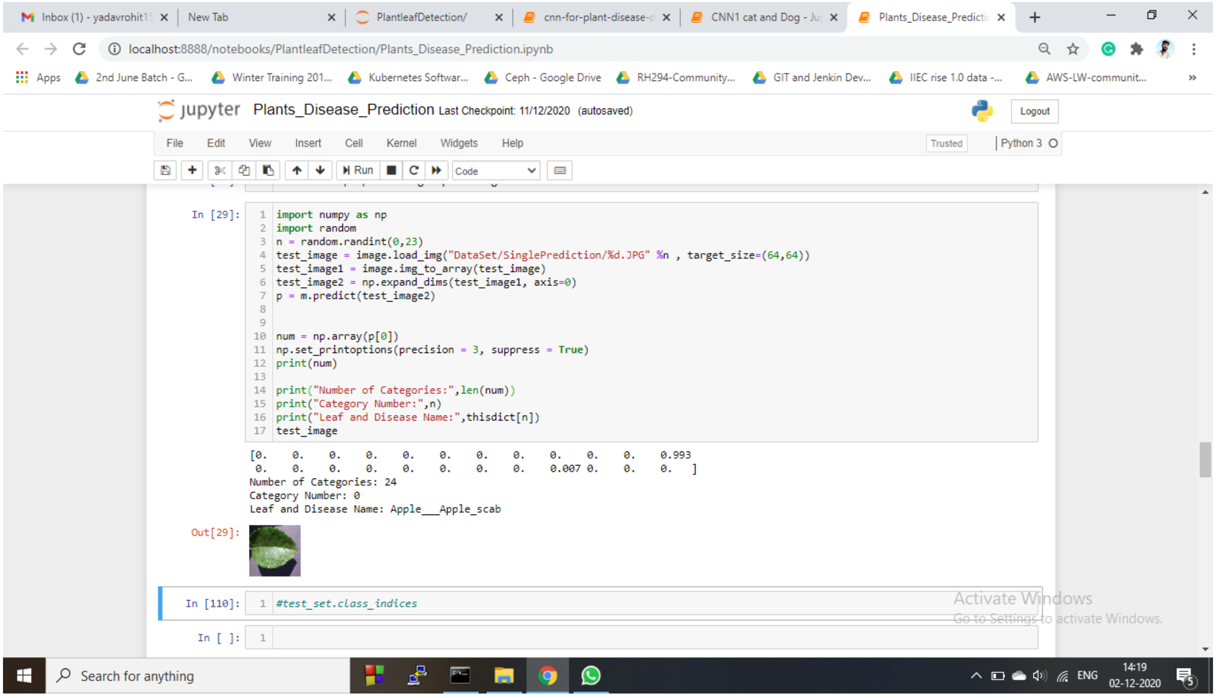


Figure 9: - Prediction 1, Diseased, Apple Scab Disease

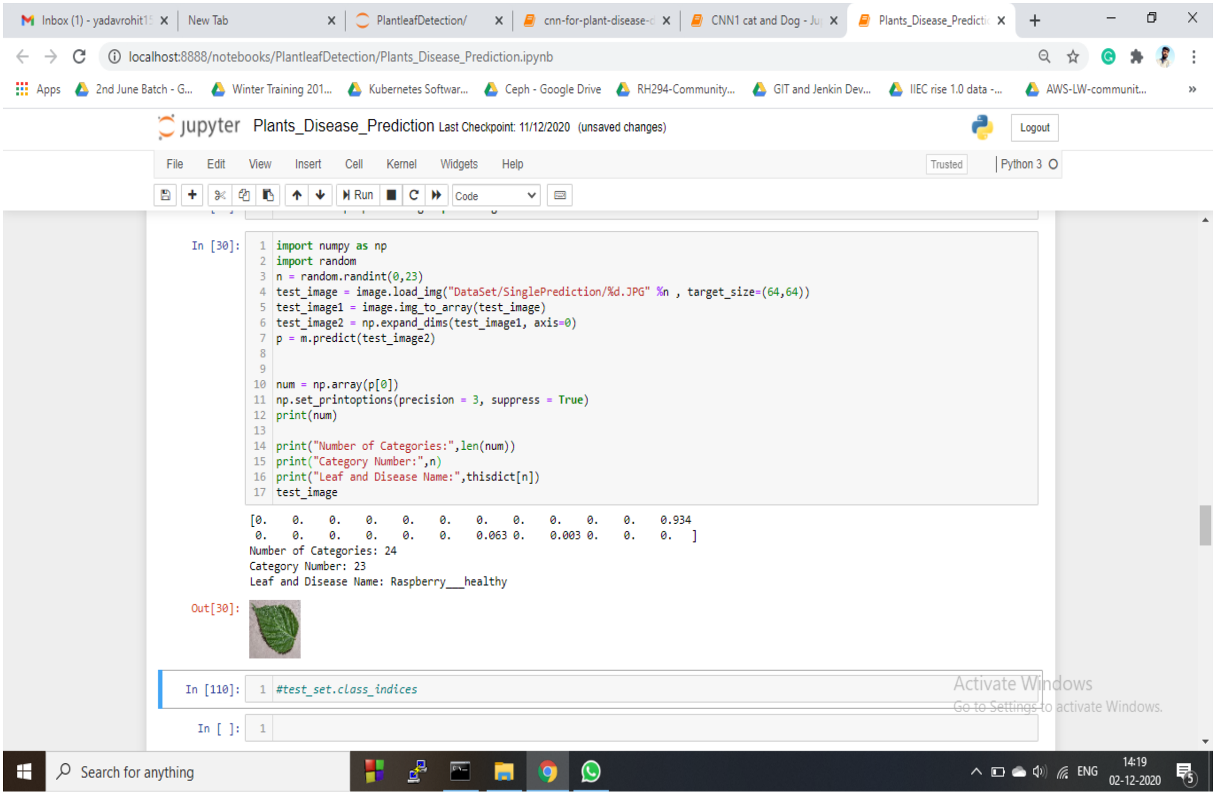


Figure 10: - Prediction 2, Healthy

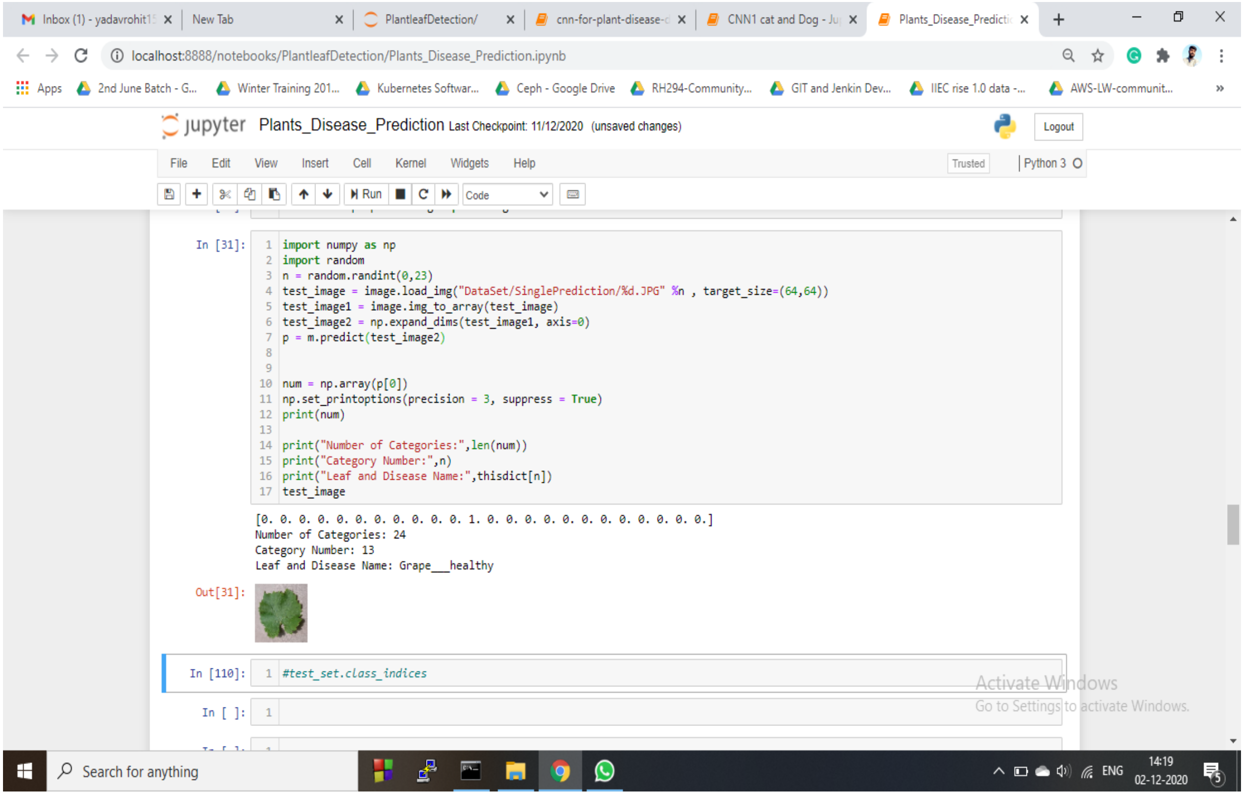


Figure 11: - Prediction 3, Healthy

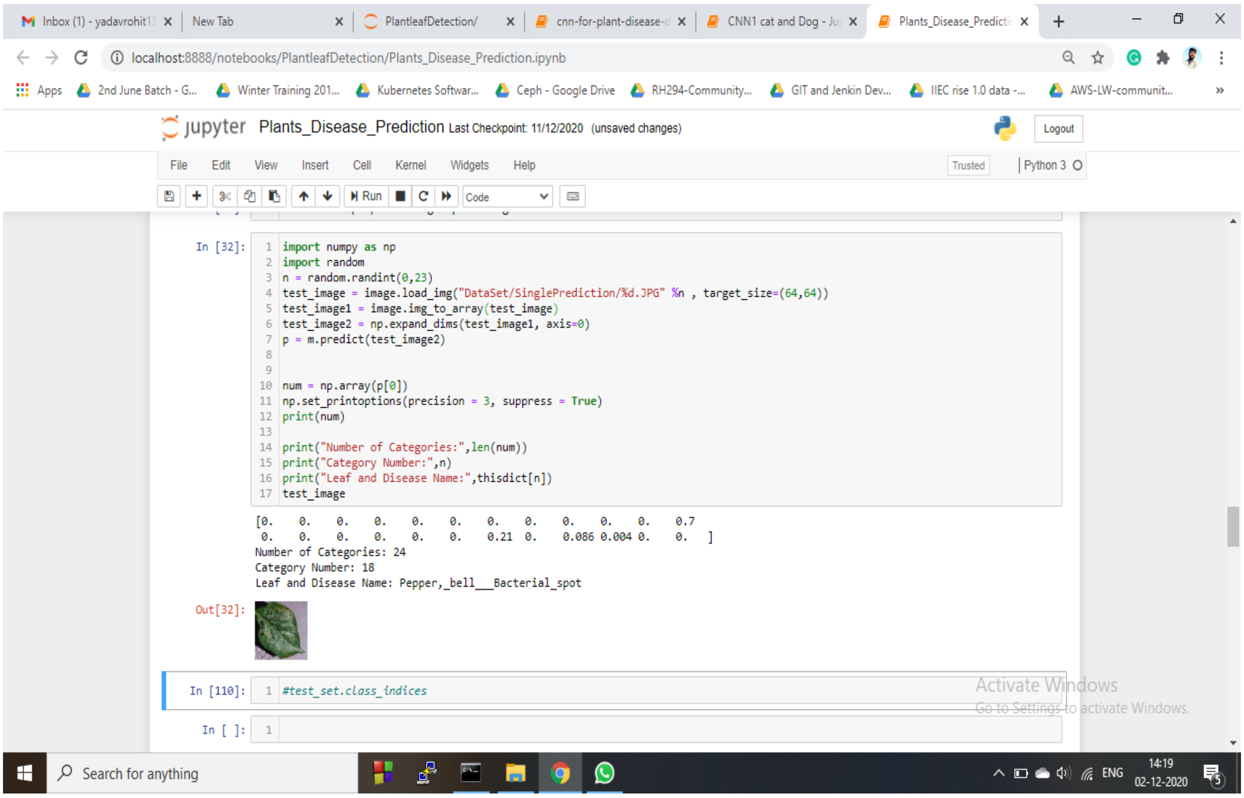


Figure 12: - Prediction 4, Diseased, Bell Bacterial Spot

References

1- Savita N. Ghaiwat, Parul AroraDetection and classification of plant leaf diseases using image processing techniques: a review
Int J Recent Adv Eng Technol, 2 (3) (2014),

2- Sanjay B. Dhaygude, Nitin P. KumbharAgricultural plant leaf disease detection using image processing Int J Adv Res Electr Electron Instrum Eng, 2 (1) (2013)

3- Sanjay B. Patil, *et al.*Leaf disease severity measurement using image processing Int J Eng Technol, 3 (5) (2011), pp. 297-301

4- K. A. Garrett, S. P. Dendy, E. E. Frank, M. N. Rouse, and S. E. Travers, "Climate change effects on plant disease: genomes to ecosystems," *Annual Review of Phytopathology*, vol. 44, pp. 489–509, 2006.

5- J. Schmidhuber, "Deep Learning in neural networks: an overview," Neural Networks, vol. 61, pp. 85– 117, 2015.

6- "Leafsnap: An Electronic Field Guide," 2016

7- M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

8- M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: detection and saliency map visualisation," in *Human and Machine Learning*, J. Zhou and F. Chen, Eds., pp. 93–117, Springer International Publishing, Cham, Switzerland, 2018.

Approved By: -

* Students made significant improvement in project report
* Still some casing errors persist.
* Implementation and discussion can be explained in more details.
* Acceptable considering the remote environment.

14 Dec 2020

Dr. T.P Singh
(Head of Department)

Dr. RAVI TOMAR
(Assistant Professor (SG))