FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

PROJECT REPORT

Submitted by

TeamID: PNT2022TMID04747

G. THARAKHAMURTHY (737819ECR204)

R. THIRUSELVAM (737819ECR207)

S. SUBASH (737819ECR191)

S. YASWANTH (737819ECR219)

In partial fulfilment

for the award of the degree Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

KONGU ENGINEERING COLLEGE

1.INTRODUCTION

Overview In this project, two datasets name fruit datasetand vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolutional Neural Networks (CNN). First, the fruit dataset is trained and then tested with CNN. It has 6 classes and all the classes are trained and tested. Second, the vegetable dataset is trained and tested.The software used for trainingand testing of datasets is Python. All the Python codes are first writtenin Jupyter notebooksupplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally,a web-based framework is designed withhelp Flask a Python library. There are 2 html files are created in templates folder along with their associated files in static folder. The Python program 'app.py' used to interface with these two webpages is written in Spyder-Anaconda python and tested.

Purpose of this project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.

2.LITERATURE SURVEY

2.1 Cloud-Based AI to Identify Early Indicators of Water Stress

Daniel, Shaurya Gupta, D. Hudson Smith. According to this paper, the critical need for sustainable water use in agriculture has prompted the EPA Strategic Plan to call for new technologies that can optimise water allocation in real-time. Demand for freshwater is rising while supply is remaining stagnant. This study evaluates the use of cloud-based artificial intelligence to spot early signs of water stress in six species of ornamental shrubs grown in containers. Earlier, modified Canon and MAPIR Survey II cameras that were flown by a small unmanned aircraft system (sUAS) at a 30 metre altitude collected near-infrared images. Cropped pictures of plants with no, low, and high levels of water stress were divided into four sets for cross-validation, and the Visual Recognition service from IBM Watson was used to train the models.

2.2 Detection of Cercospora leaf spot in okra using CNN

Rangarajan, Aravind Krishnaswamy, and Edwin Jayaraj Balu. Robotic agriculture powered by artificial intelligence (AI) is predicted to significantly disrupt the agricultural industry and offer profitable farming. However, there are significant obstacles to implementing such technologies in economically underdeveloped nations in a way that is both profitable and cost-effective. In this study, the disease Cercospora Leaf Spot (CLS), also known as lady's finger or Abelmoschus esculentus L., is identified in the okra plant. Deep learning models are used to detect the disease from images taken with a quadcopter that has been modified to be inexpensive and fitted with a camera. SqueezeNet and ResNet-18, two deep learning models with validation accuracy of 99.1% and 99%, respectively, were used in this study. The models' accuracy was tested using images obtained by the modified quadcopter, and the results were 92.3% and 94.6%, respectively.

3.THEORITICAL ANALYSIS

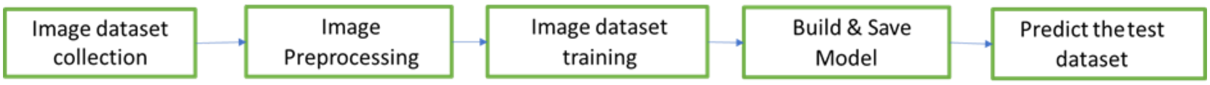


Fig 3.1 Block Diagram

The block diagram of the entire project is shown in Fig.3.1. First step is the image dataset collection followed by image preprocessing. The third step is the training of image datasets with initializing different hyper parameters. Then build the model and save the model file with .h5 format. The final stage is the testing of existing or new datasets using the trained model

4.HARDWARE/SOFTWARE DESIGNING

The software used for training and testing the dataset is Python. The Jupyter notebook (Notebook of IBM cloud also) is used for python programming. The neural network used for training and testing the model is Convolutional NeuralNetwork (CNN).

The CNN has following layers:

- Convolutional layer (32x32kernel (3x3))
- Flatten layer
- Dense layer (different layers with differentsize)
- Drop out layer(optional)

- Final output dense layer (size 6x1 for fruit datasetand 9x1 for Vegetable dataset)
- Max-pool layer (kernel(2x2))

In the preprocessing step, images are normalized to 1 and then resized to 128x128. The images are arranged in different batch sizes. Then train set and test set are formed from the collected datasets. In order to do the above steps in Python, the followingPython libraries must be imported before starting the process:

- NumPy, Matplotlib
- TensorFlow, Keras.

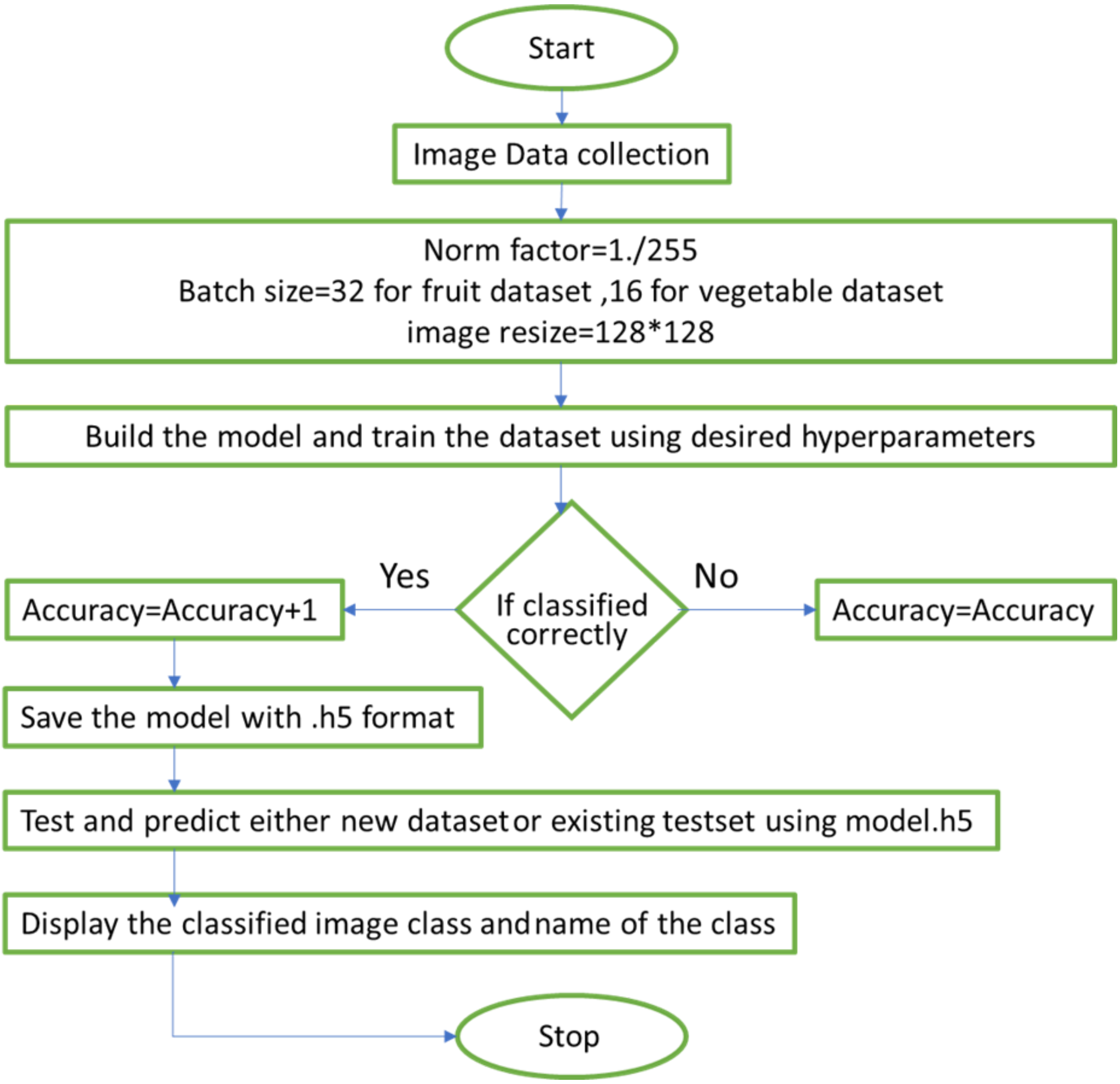
The followingactivation functions used in the CNN training:

- RELU at the end of convolution layer and Max Pool layer
- SoftMax at the end of output dense layer
- For testing the dataset argmax is used, it’s an optional.

4.EXPERIMENT INVESTIGATION

Analysis made while working on the solution The batch sizes are varied and tested. For different batch sizes, the CNN gives different accuracies. The batch size determines the number of iterations per epoch. Another important hyper parameteris the number of epochs. This determines accuracy and it has high influence on accuracy comparedto other hyper parameters. The accuracy can be variedfrom 80% to 90% in vegetable datasetand 95% to 98% in the case of fruit dataset by increasing the number of epochs. The size of test dataset and train dataset also has very highinfluence on accuracies. The accuracy can be increased by using a greater number of imagesin train dataset. The computational time for model building is increased when the size of the train datasetincreased and also number of epochs increased. The batch size of train dataset and test datasetalso play a vital role in computational time. The NeuralNetwork complexity is increased when a greater number of convolutional layersincreased. If the number of layers increased, better accuracy result will obtain. At the same increasing the number of layers in CNN leads to more training time and also requires more time to build a model. The model .h5 size depends on the size of train datasets. But the memory requirement depends on the size of train dataset and CNN architecture complexity.

5.FLOW CHART


Start
Image Data collection
Norm factor=1./255
Batch size=32 for fruit dataset ,16 for vegetable dataset
image resize=128*128
Build the model and train the dataset using desired hyperparameters
If classified correctly
Yes
No
Accuracy=Accuracy+1
Accuracy=Accuracy
Save the model with .h5 format
Test and predict either new dataset or existing testset using model.h5
Display the classified image class and name of the class
Stop

6.RESULTS

Final findings(output) of the project given below in the form of screenshot: Training and Testing of Fruit dataset

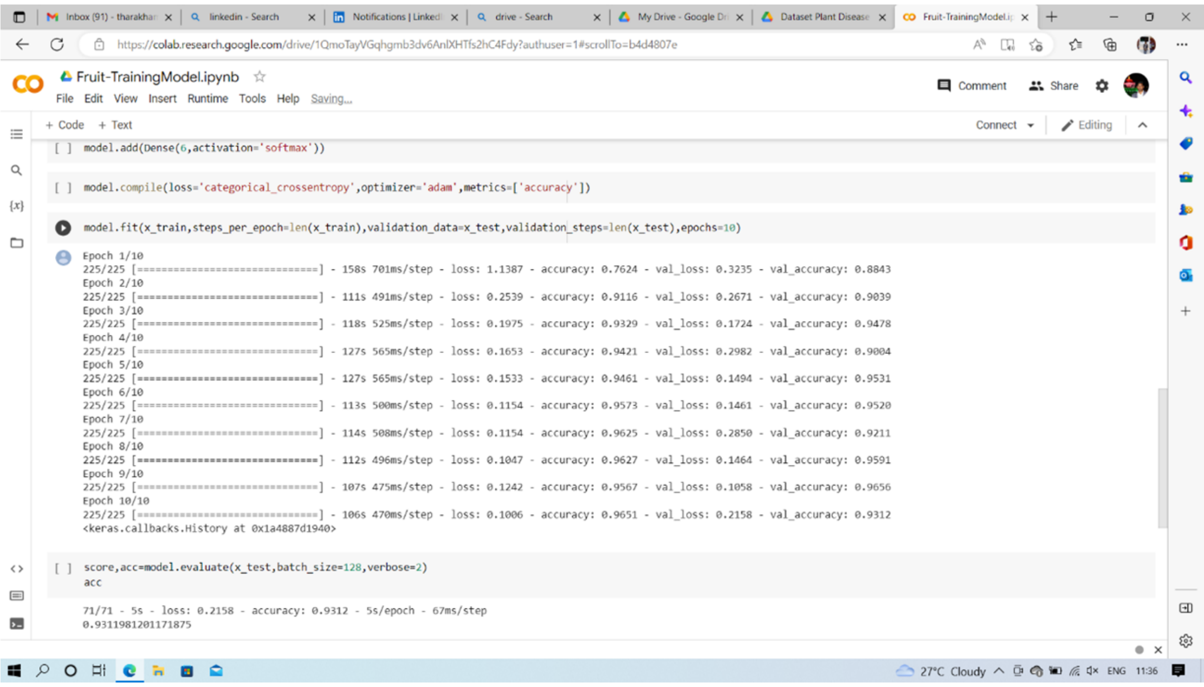


Figure 6.1 Training of Fruit data

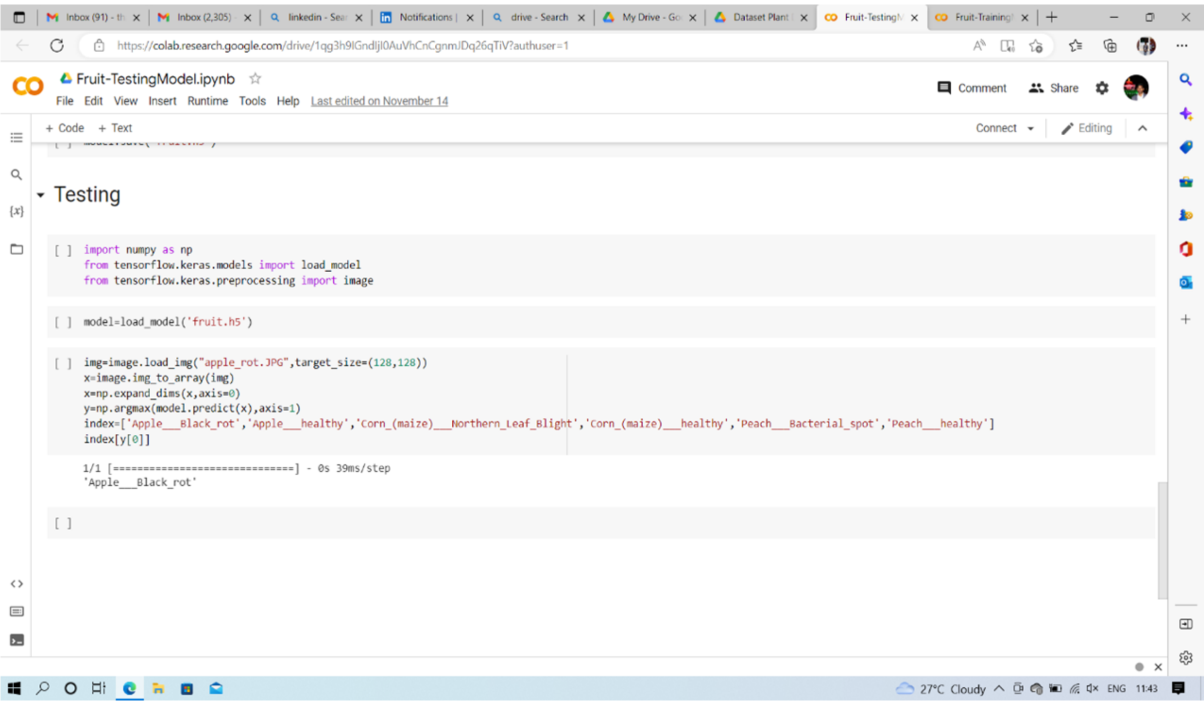


Figure 6.2 Testing of Fruit data

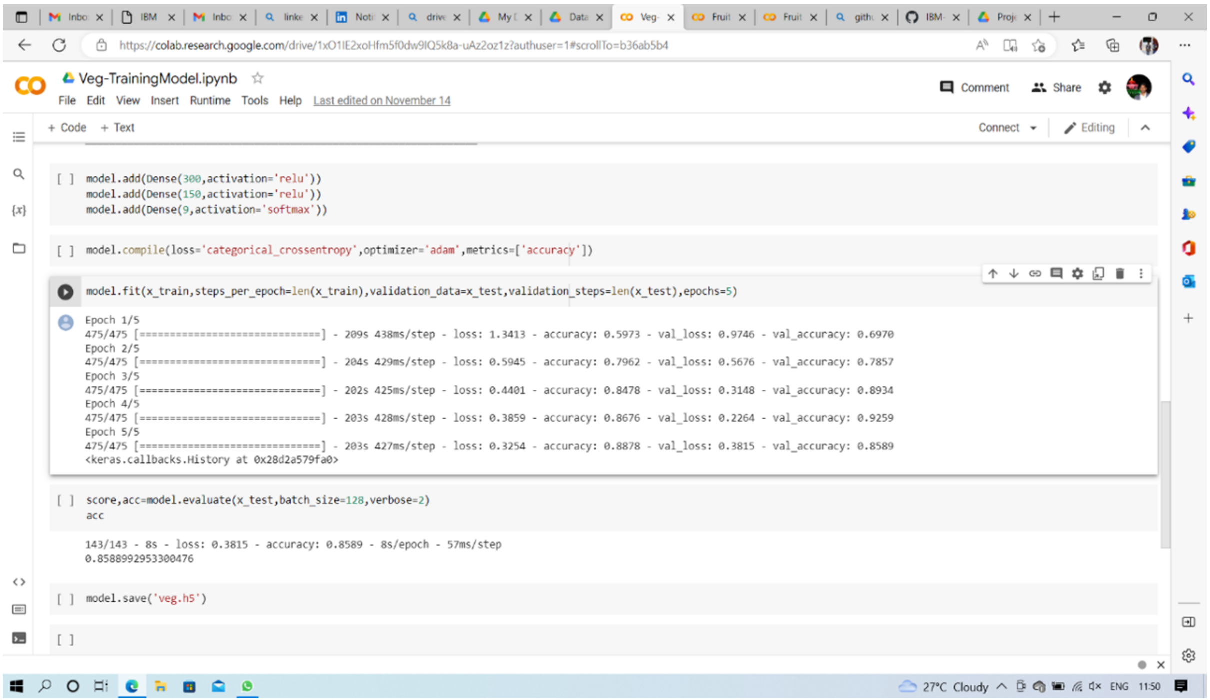


Figure 6.3 Training of Vegetable Data.

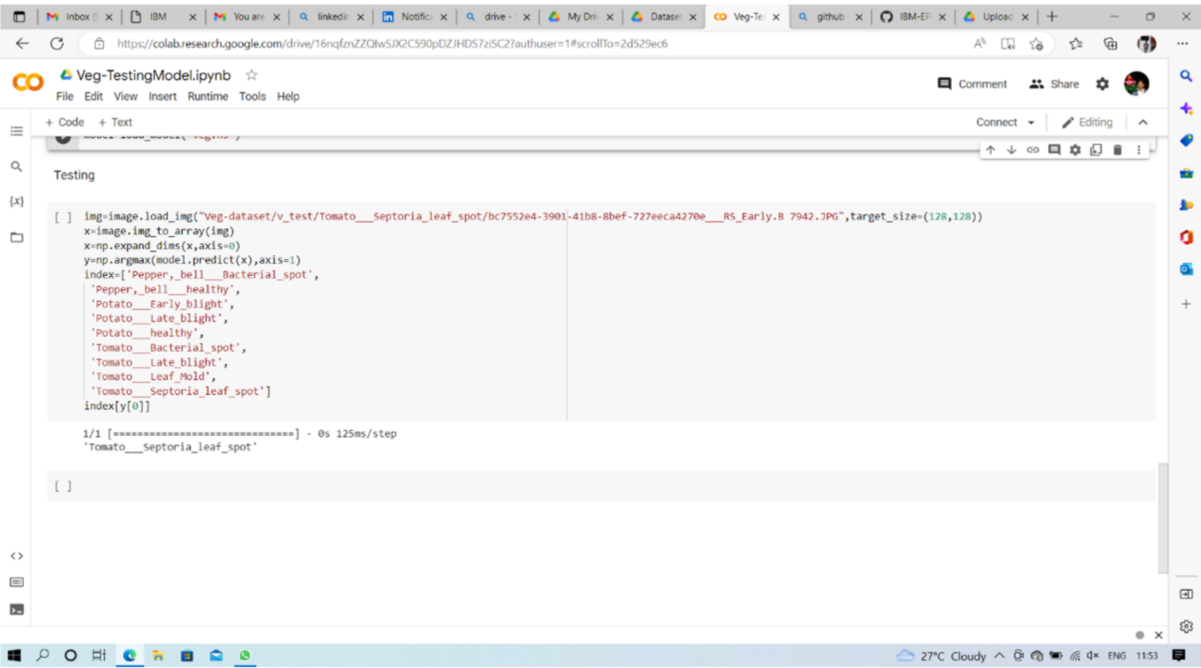


Figure 6.4 Testing of Vegetable Data

OUTPUT

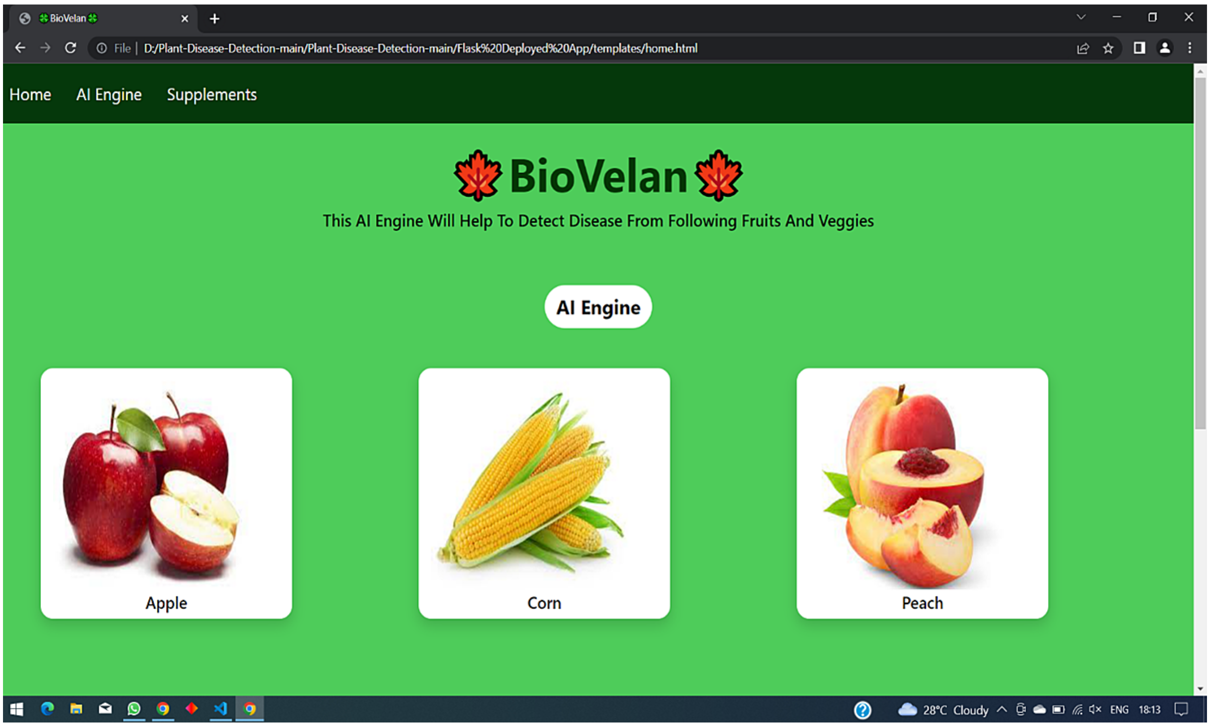


Figure 6.5 Home Page

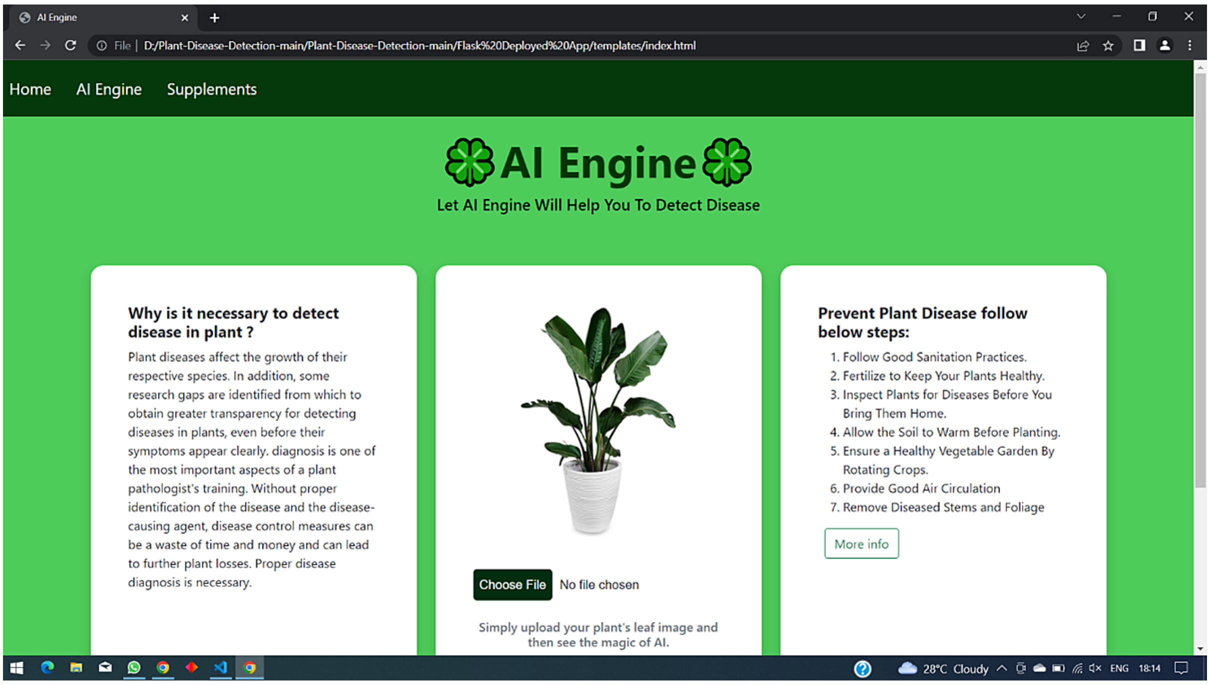


Figure 6.6 Image Upload Page

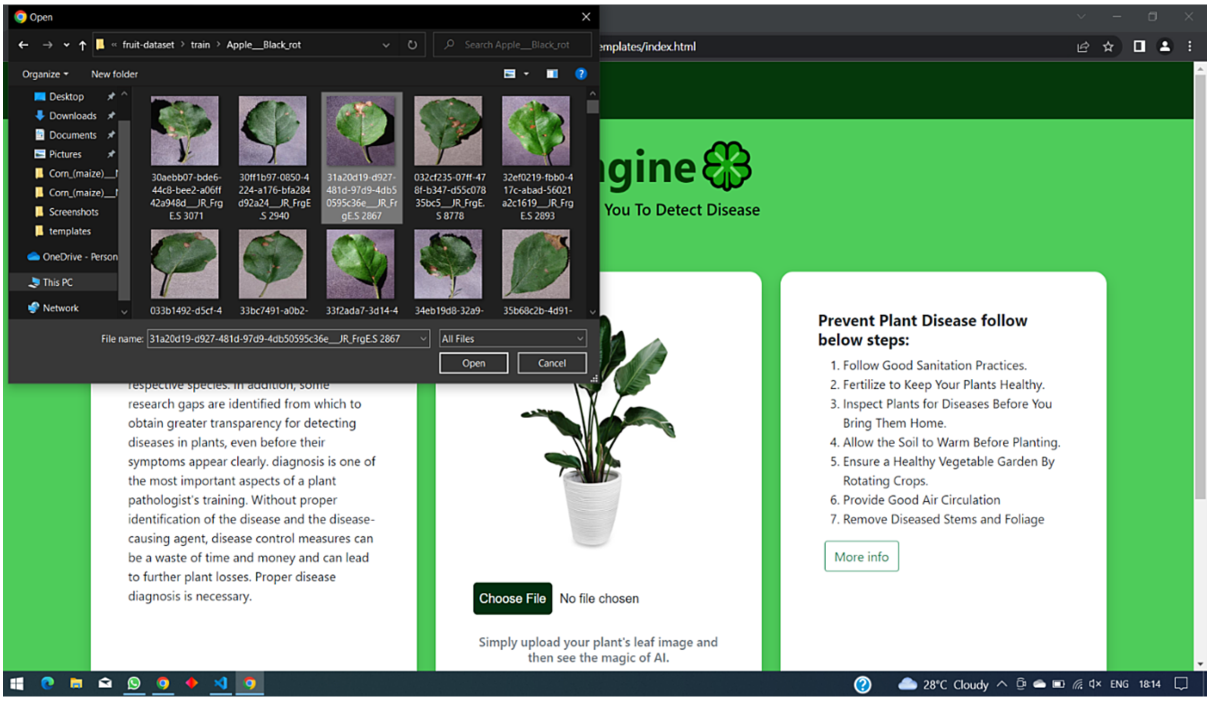


Figure 6.7 Upload the Image of Diseased Leaf

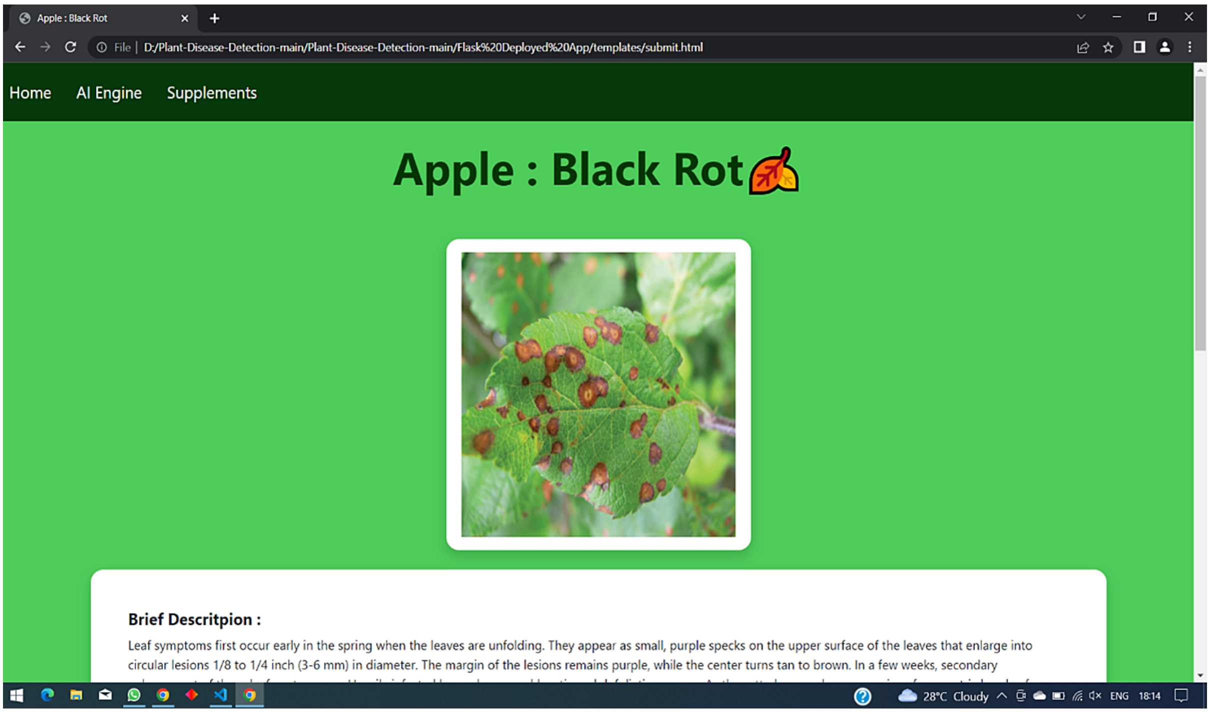


Figure 6.8 Disease Information

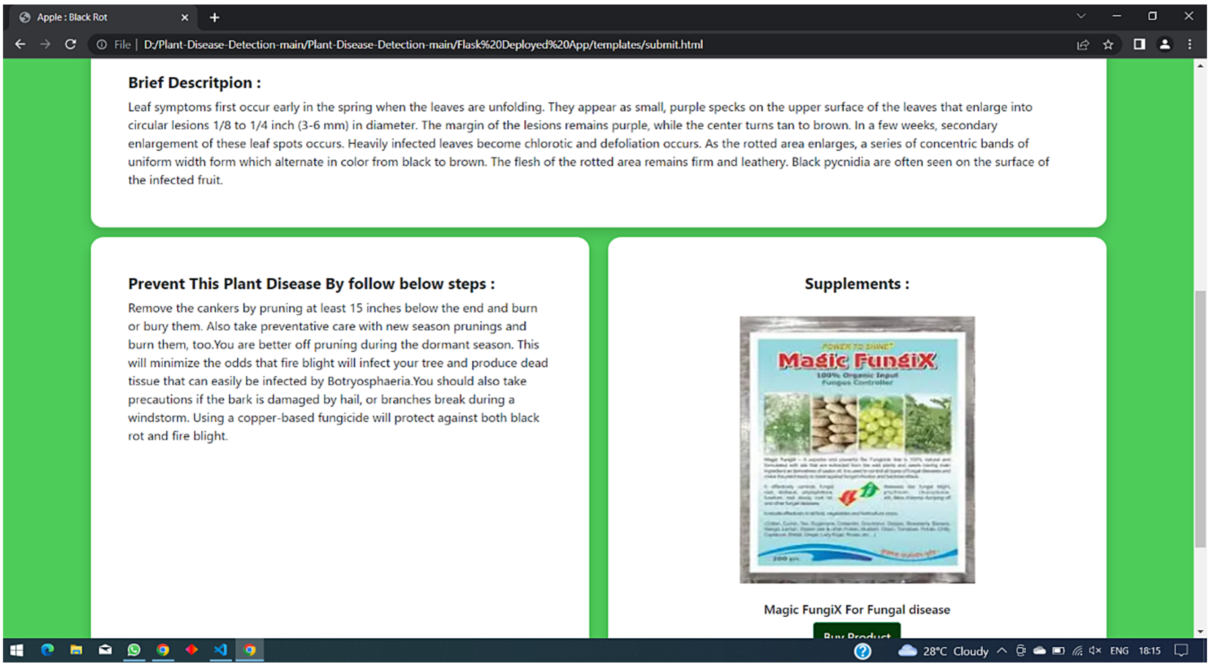


Figure 6.9 Fertilizer Recommendation

7.ADVANTAGES & DISADVANTAGES

List of advantages

- The proposed model here producesvery high accuracyof classification.
- Very large datasetscan also be trained and tested.
- Images of very high can be resized withinthe proposed itself.

Listof disadvantages

- For trainingand testing, the proposed model requires very high computationaltime.
- The neuralnetwork architecture used in this project work has high complexity.

8. APPLICATIONS

- The trained network model used to classify the image patterns with high accuracy.
- The proposed model not only used for plant disease classification but also for otherimage pattern classification such as animal classification.
- This project work application involves not only image classification but also for pattern recognition.

9.CONCLUSIONS

The model proposed here involves image classification of fruit datasetsand vegetable datasets. The following points are observed during model testing and training:

- The accuracyof classification increased by increasing the number of epochs.
- For differentbatch sizes, differentclassification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracyof classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different whilevarying the size of the train and test dataset.

10.FUTURE SCOPE

Theproposed model in this projectwork can be extended to image recognition. The entire model can be converted to application softwareusing python to exe software.The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

11.BIBILOGRAPHY

[1]. Clemente, Andressa Alves, Gabriel Mascarenhas Maciel, Ana Carolina Silva Siquieroli, Rodrigo Bezerra de Araujo Gallis, Lucas Medeiros Pereira, and Jéssyca Gonçalves Duarte. "High-throughput phenotyping to detect anthocyanins, chlorophylls, and carotenoids in red lettuce germplasm." International Journal of Applied Earth Observation and Geoinformation 103, 2021

[2]. Kerkech, Mohamed, Adel Hafiane, and Raphael Canals. "Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach." Computers and Electronics in Agriculture 174, 2020.

12. APPENDIX

A. Source Code (Jupyter notebookpython code)

```
"""Fruit-TestingModel.ipynb"""
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=Fa
```

```
lse)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory('fruit-dataset/f_train',
                                           target_size=(128,128),batch_size=24,class_mode='categorical')
x_test=test_datagen.flow_from_directory('fruit-dataset/f_test',
                                         target_size=(128,128),batch_size=24,class_mode='categorical')

x_train.class_indices

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.summary()

model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(6,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs
=10)

score,acc=model.evaluate(x_test,batch_size=128,verbose=2)
acc

model.save('fruit.h5')

"""#Testing"""

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

model=load_model('fruit.h5')
```

```
img=image.load_img("apple_rot.JPG",target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']
index[y[0]]
```

```
"""Veg-TestingModel.ipynb"""

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory('veg-dataset/v_train',
                                          target_size=(128,128),batch_size=24,class_mode='categorical')
x_test=test_datagen.flow_from_directory('veg-dataset/v_test',
                                        target_size=(128,128),batch_size=24,class_mode='categorical')

x_train.class_indices

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.summary()

model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs
=5)

score,acc=model.evaluate(x_test,batch_size=128,verbose=2)
acc

model.save('veg.h5')




import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

model=load_model('veg.h5')

"""Testing"""

img=image.load_img("Veg-dataset/v_test/Tomato___Septoria_leaf_spot/bc7552e4-3901-41b8-8bef-
727eeca4270e___RS_Early.B 7942.JPG",target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Pepper,_bell___Bacterial_spot',
 'Pepper,_bell___healthy',
 'Potato___Early_blight',
 'Potato___Late_blight',
 'Potato___healthy',
 'Tomato___Bacterial_spot',
 'Tomato___Late_blight',
 'Tomato___Leaf_Mold',
 'Tomato___Septoria_leaf_spot']
index[y[0]]
```