Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID0341

AMILENENI DHANUSH

HEMANTH N

KANCHARLA LIKITH CHOWDARY

VENKATA ASI SUPREETH

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION ENGINEERING

SAVEETHA ENGINEERINGCOLLEGE ,

THANDALAM.

1.INTRODUCTION

1.1 Overview Two datasets—the fruit dataset and the vegetable dataset—are gathered for this research. Convolutional Neural Networks, a deep learning neural network, is used to train and test the datasets that have been collected (CNN). The fruit dataset is first trained, and then CNN is tested. There are 6 courses total, and each class is trained and tested. The vegetable dataset is then tested and trained. Python is the programme used to train and test datasets. All of the Python code is initially created in the Jupyter notebook that comes with Anaconda Python, and it is then tested in the IBM cloud. Finally, Flask, a Python package, is used to construct a web-based framework. Along with their related files in the static folder, two html files are created and placed in the templates folder. The Spyder-Anaconda Python software "app.py" that interfaces with these two webpages was developed and tested.

1.2 Purpose This study is used to test samples of fruits and vegetables and find out which diseases they may have. Additionally, this project suggests fertilisers for certain ailments.

2.LITERATURE SURVEY

2.1 Existing issue Indumathi suggested a technique for spotting leaf illnesses and suggested fertilisers to treat them. However, the method's low number of train and test sets leads to subpar accuracy. A straightforward approach of prediction for a soil-based fertiliser prescription system for anticipated crop diseases was put out by Pandi Selvi. This approach offers less predictability and accuracy. Shiva Reddy proposed a machine learning-based IoT-based system for recommending fertiliser and detecting leaf disease that has less than 80% accuracy.

2.2 Proposed remedy A deep learning-based neural network is utilised in this project's effort to train and test the datasets that were gathered. CNN, a deep learning-based neural network, provides classification accuracy rates of greater than 90%. By boosting The accuracy rate can rise to 95% to 98% by adding more dense layers and changing hyperparameters like the number of epochs and batch size.

3.THEORITICAL ANALYSIS

3.1 Block diagram

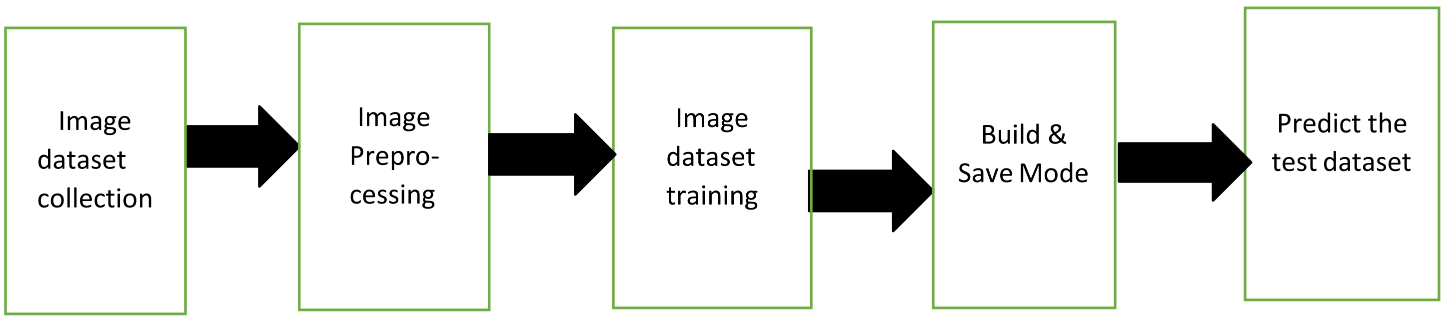


Figure.3.1. Block Diagram of the project

Fig.3.1 displays the project's overall block diagram. The collecting of picture datasets comes first, then comes image preparation. The training of picture datasets using various hyperparameter initializations is the third phase. After that, create the model and save it in.h5 format. The final step involves applying the trained model to test new or existing datasets.

3.2 Hardware/Software designing

Python is the programme used to train and test the dataset. Python programming is done in a notebook tool called Jupyter, which also works with the IBM cloud. Convolutional Neural Network is the neural network that was utilised to train and test the model (CNN).

The CNN has following layers:

- Convolutional layer (32x32 kernal (3x3))

- Max-pool layer (kernel(2x2))
- Flatten layer
- Dense layer (different layers with different size)
- Drop out layer (optional)
- Final output dense layer(size 6x1 for fruit dataset and 9x1 for Vegetable dataset)

Images are normalised to 1 and then downsized to 128x128 in the preparation stage. Different batch sizes are used to arrange the photos. Then, using the gathered datasets, train set and test set are created. The following Python libraries need to be imported before beginning the process in order to perform the aforementioned actions in Python:

- NumPy
- TensorFlow
- Keras
- Matplotlib (optional for data visualization)

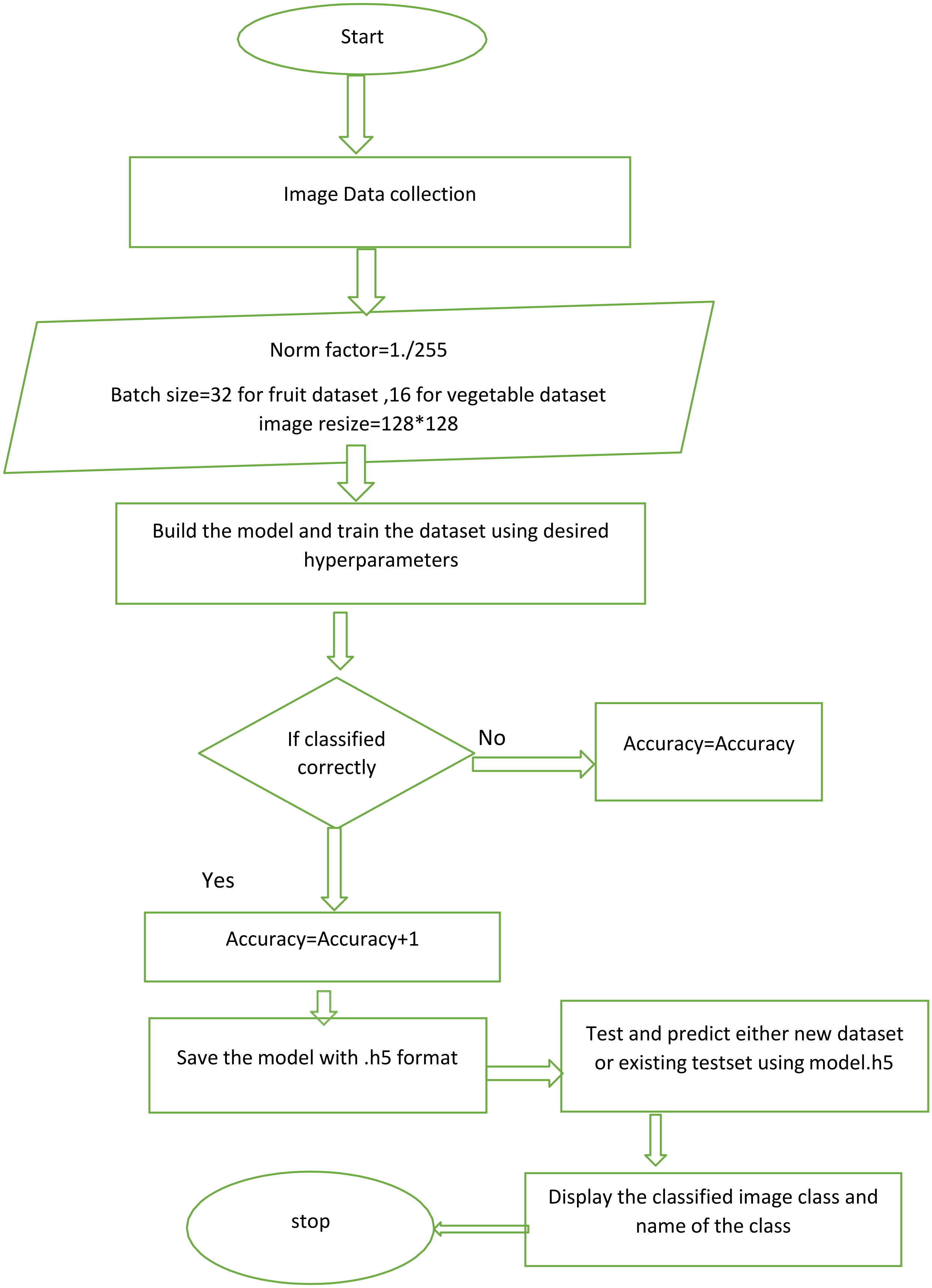
The following activation functions used in the CNN training:

- Max Pool layer and RELU at the conclusion of the convolution layer
- End of output dense layer SoftMax
- The dataset argmax is utilised for testing, and it's optional

4.EXPERIMENTAL INVESTIGATIONS

Analysis performed when developing the solution There are various and tested batch sizes. The accuracy provided by CNN varies with batch size. Size of the batch specifies how many iterations there are in each epoch. The quantity of epochs is a key hyper parameter. When compared to other hyper metrics, this affects accuracy and has a significant impact on accuracy. By increasing the number of epochs, the accuracy may be adjusted from 80% to 90% for the vegetable dataset and from 95% to 98% for the fruit dataset. The size of the training and test datasets also has a significant impact on accuracy. More photos can be added to the train dataset to boost accuracy. The size of the train dataset and the number of epochs both increase the computational time required to create a model. The train dataset and test dataset batch sizes are also very important in terms of processing time. When there are additional convolutional layers, the complexity of the neural network increases. A higher layer count will produce results with greater accuracy. The CNN algorithm requires more time to develop a model and spend more time training as the number of layers increases. The size of the train datasets affects the model.h5 size. However, the amount of the train dataset and the complexity of the CNN architecture determine the memory need.

5.FLOWCHARTS



6.RESULTS

The following is a screenshot of the project's final results: Fruit dataset testing and training

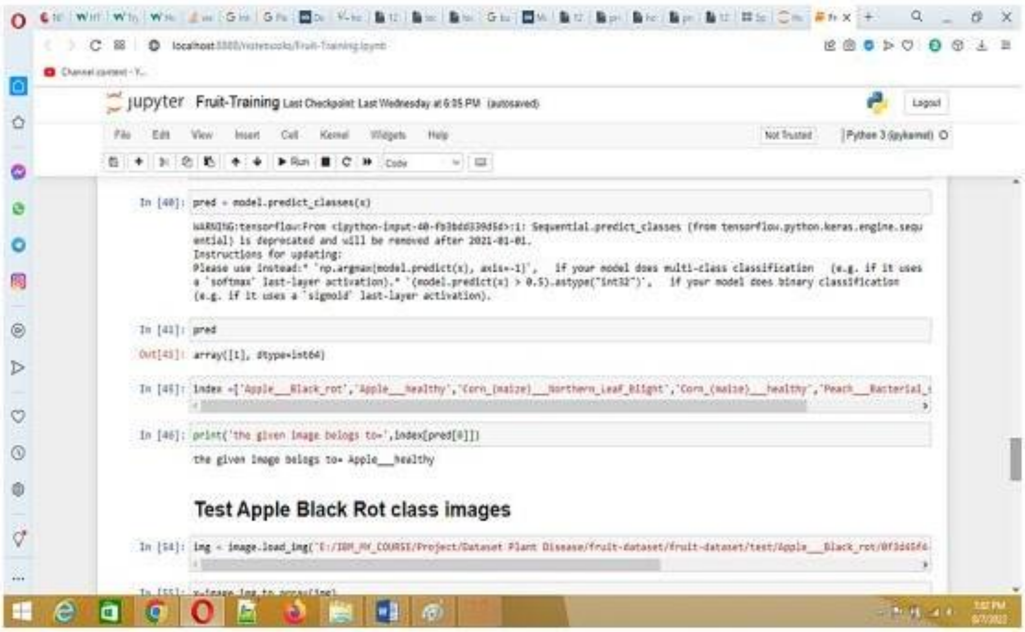


Figure.6.2 Test the Fruit dataset

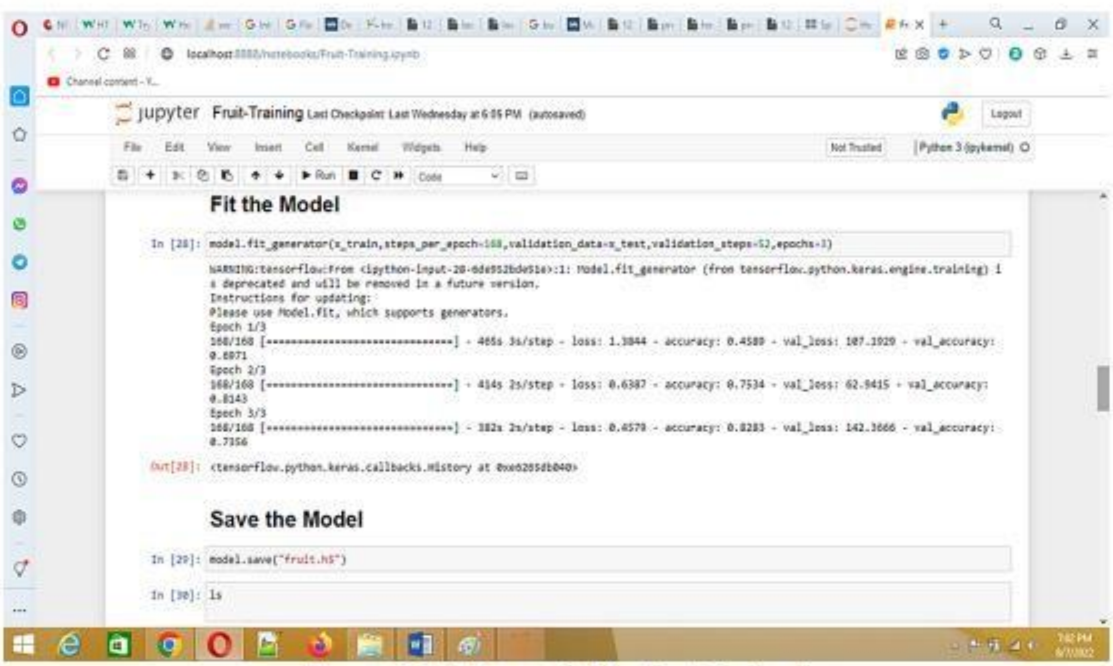


Figure.6.1. Fit a model for Fruit dataset

Train and Test Vegetable dataset

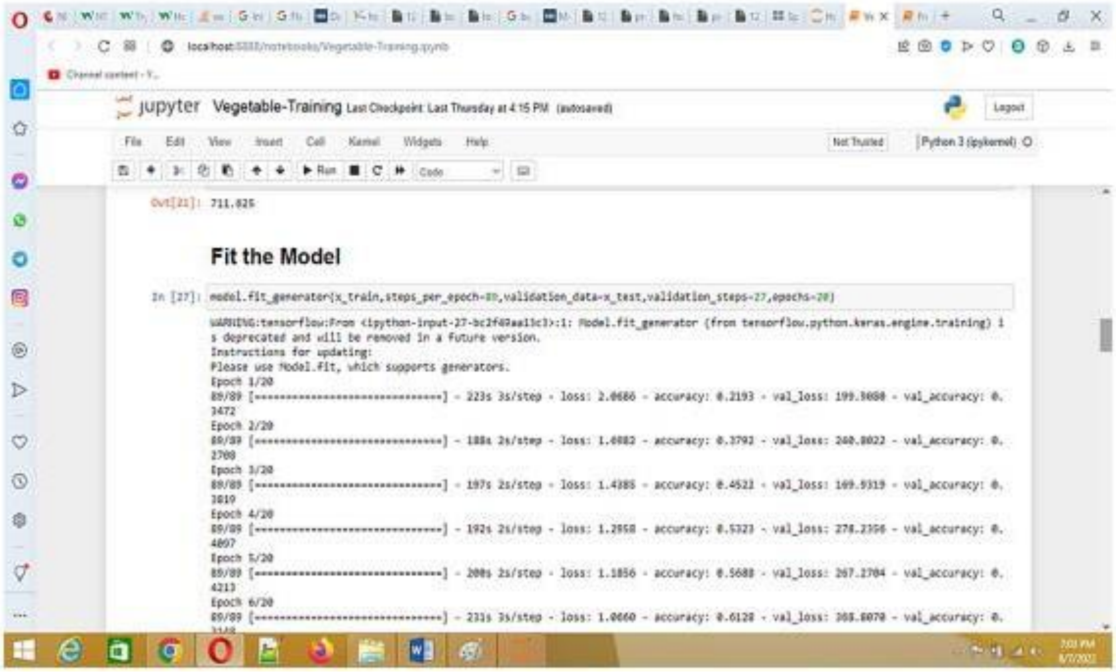


Figure.6.3. Train the Vegetable dataset

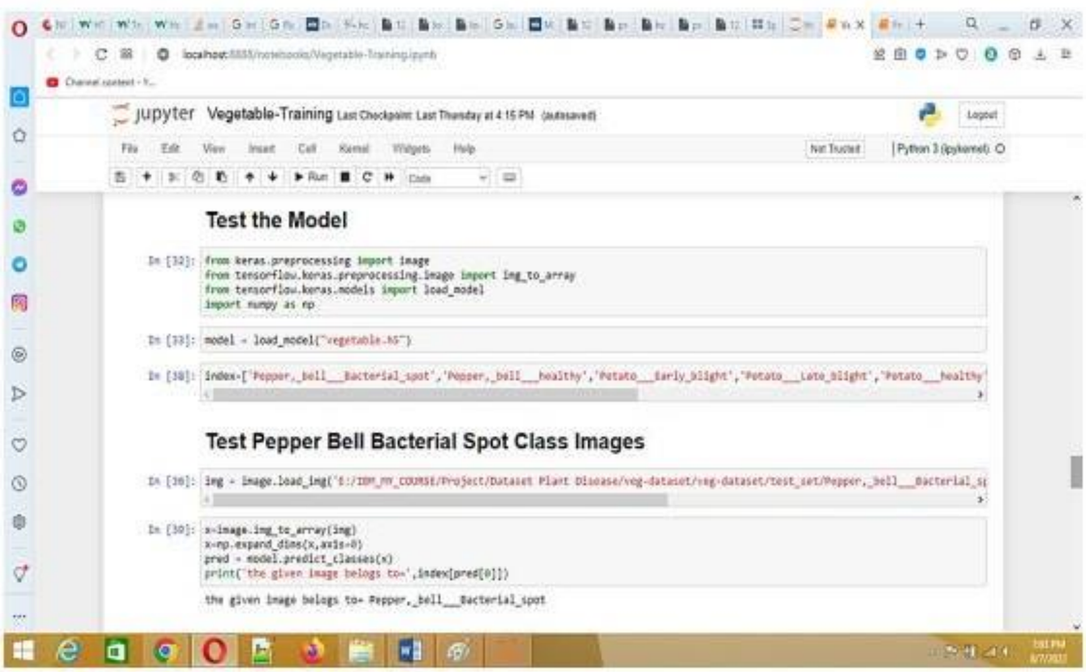


Figure.6.4. Test the Vegetable dataset

Train and Test Vegetable dataset IBM Cloud

Due to CUH limit exceeds, I have downloaded the notebooks and opened in Jupyter notebook

(i). Fruit dataset

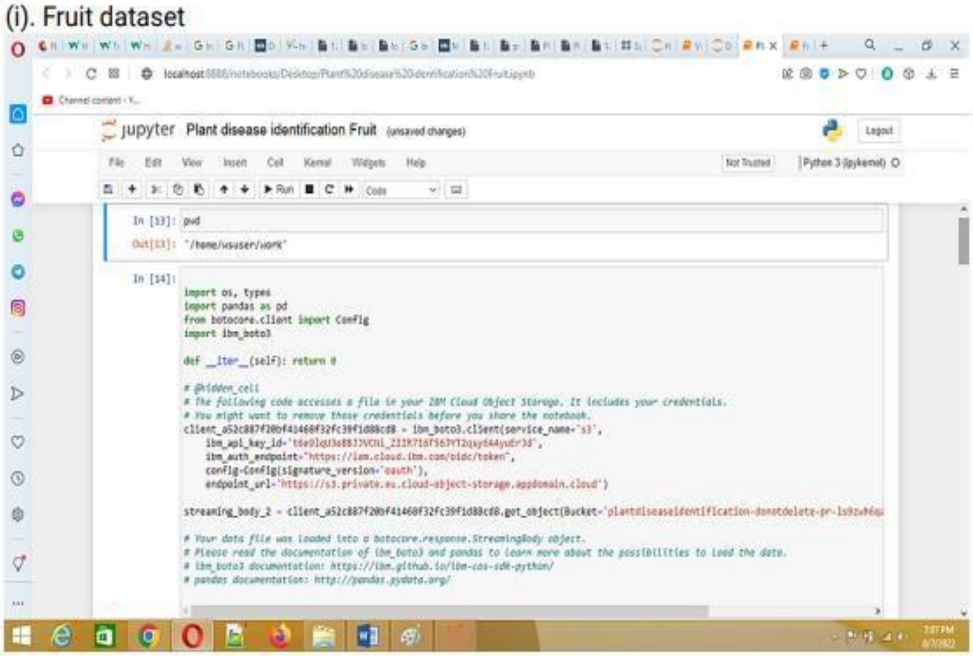


Figure.6.5. Training Fruit Dataset in IBM Cloud

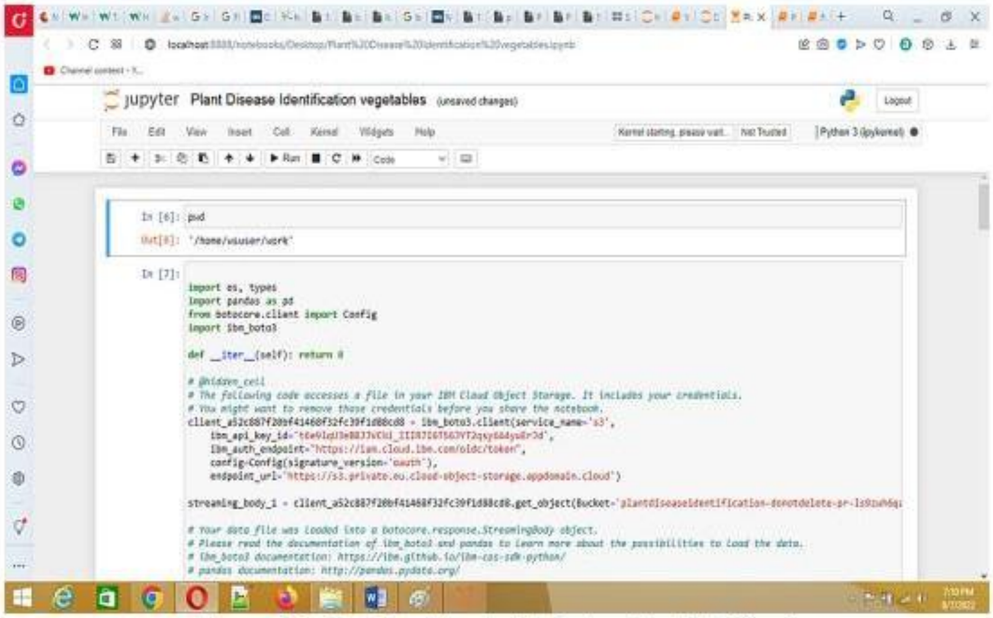
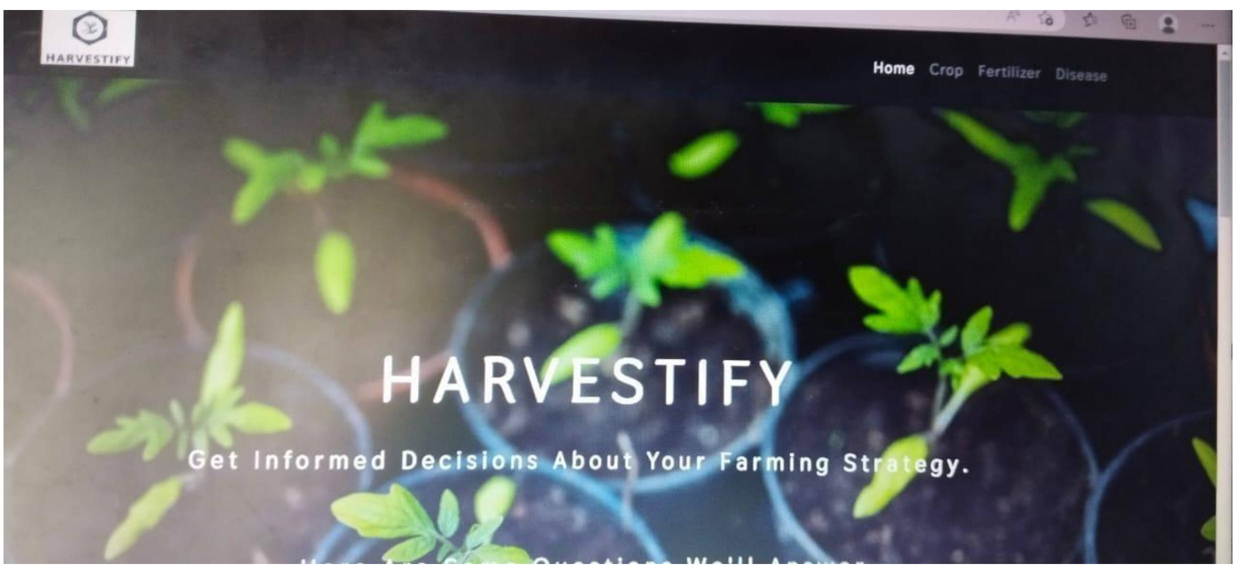
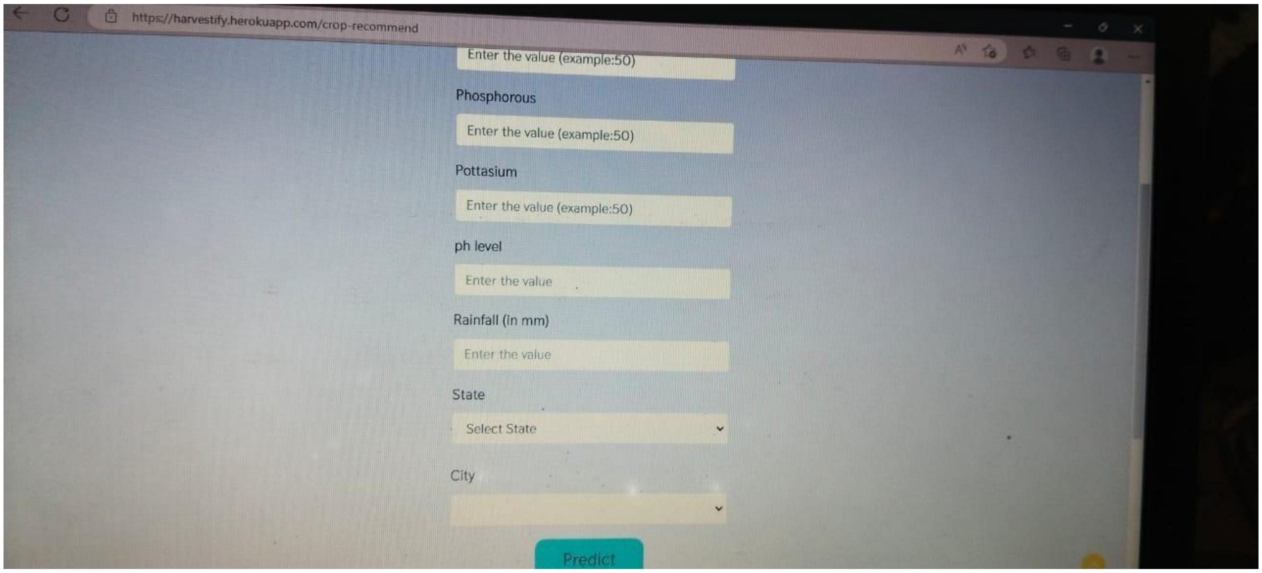
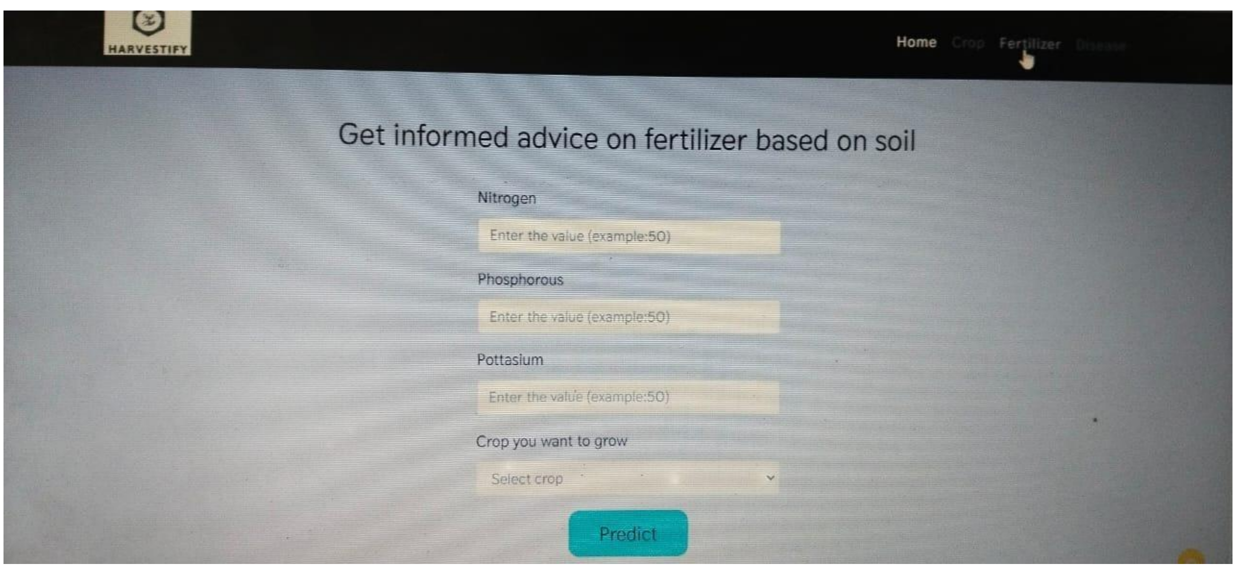


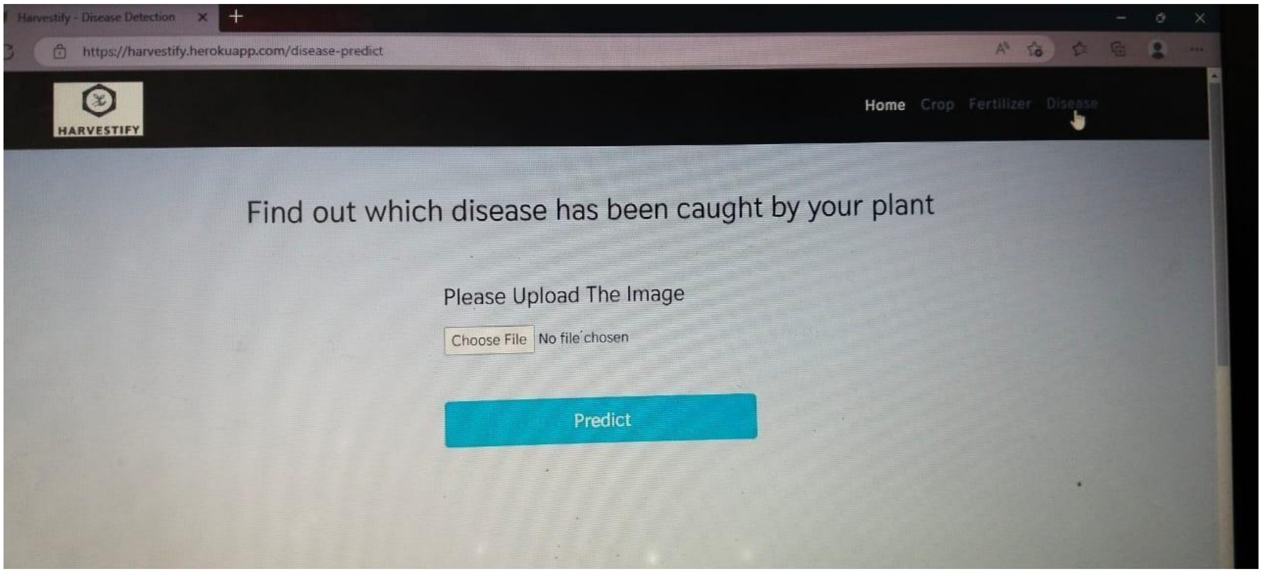
Figure.6.6. Training Vegetable Dataset in IBM Cloud

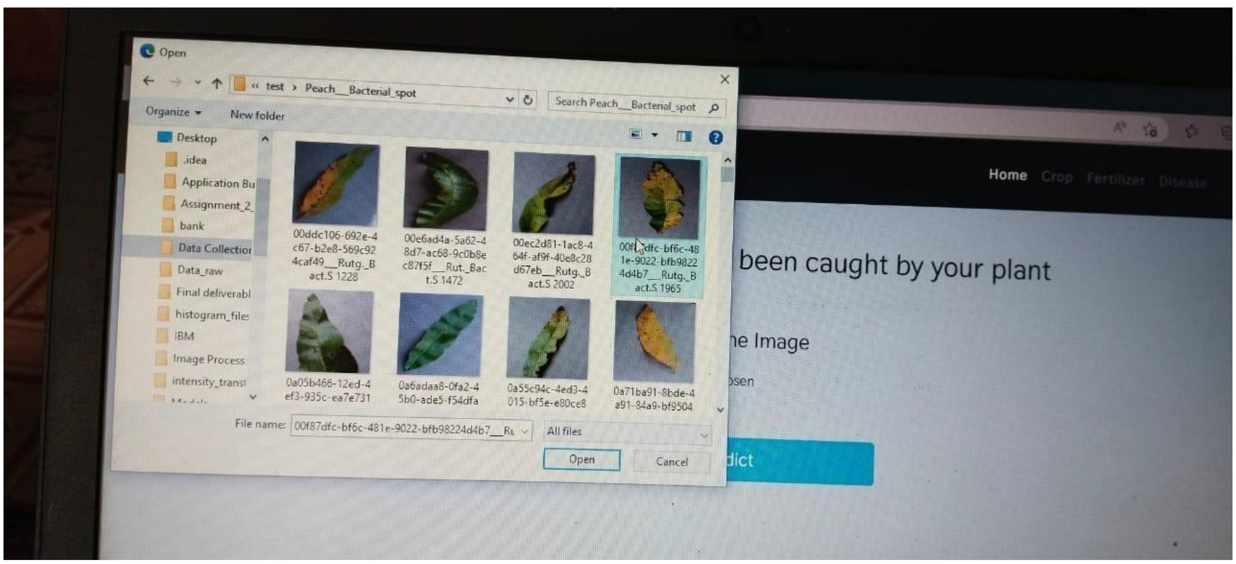
Out put

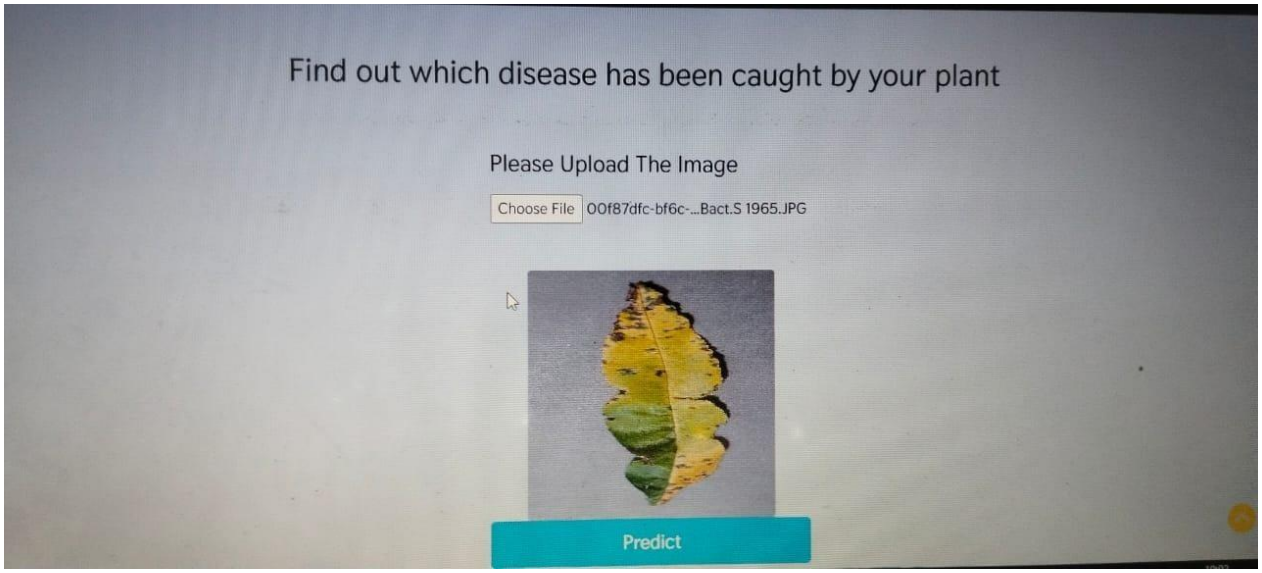


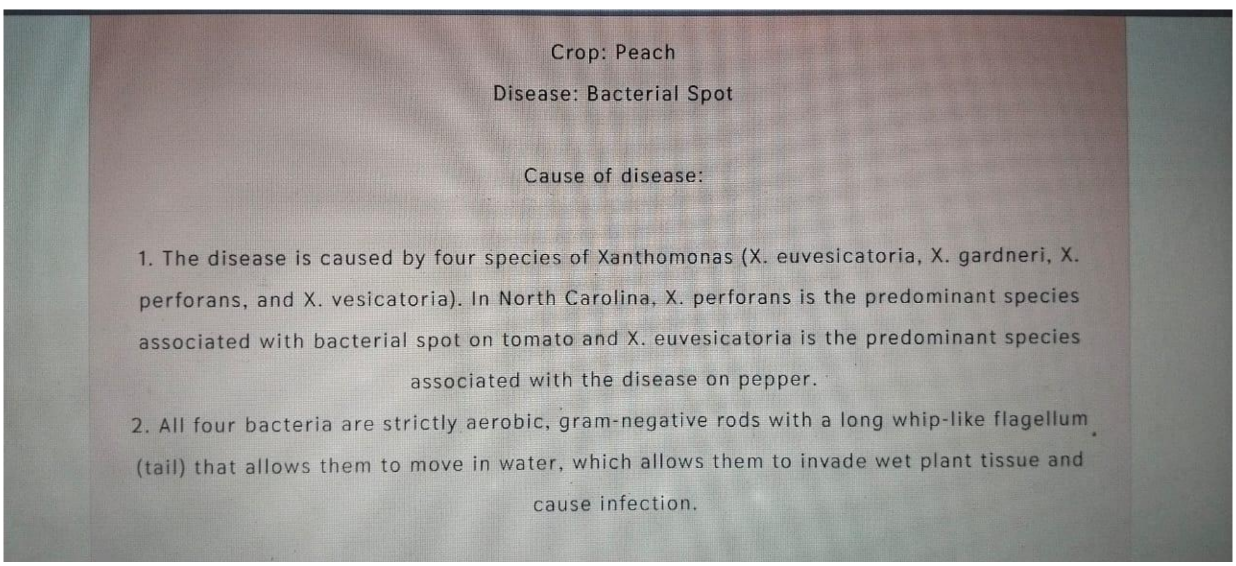

https://harvestify.herokuapp.com/crop-recommend
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
ph level
Enter the value
Rainfall (in mm)
Enter the value
State
Select State
City
Predict

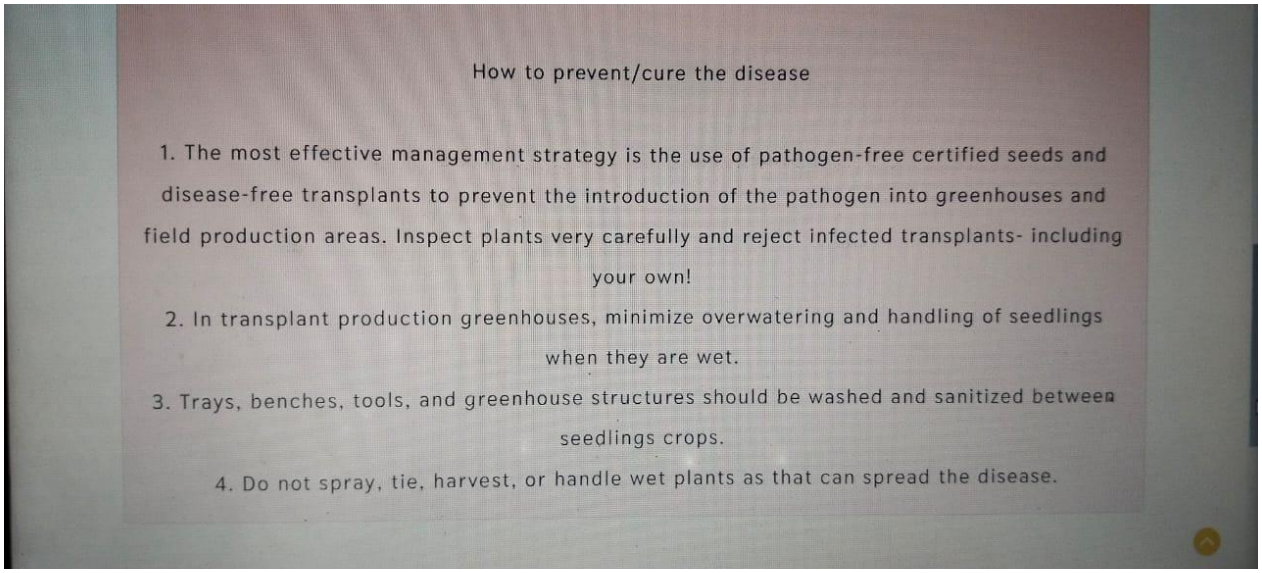

HARVESTIFY
Home
Fertilizer
Get informed advice on fertilizer based on soil
Nitrogen
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
Crop you want to grow
Select crop
Predict


Harvestify - Disease Detection
https://harvestify.herokuapp.com/disease-predict
HARVESTIFY
Home
Crop
Fertilizer
Disease
Find out which disease has been caught by your plant
Please Upload The Image
Choose File
No file chosen
Predict


Open
test
Peach__Bacterial_spot
Organize
New folder
Desktop
Home
Crop
Fertilizer
been caught by your plant
File name:
All files
Open
Cancel


Find out which disease has been caught by your plant
Please Upload The Image
Choose File
00f87dfc-bf6c-...Bact.S 1965.JPG
Predict


Crop: Peach
Disease: Bacterial Spot
Cause of disease:
1. The disease is caused by four species of Xanthomonas (X. euvesicatoria, X. gardneri, X. perforans, and X. vesicatoria). In North Carolina, X. perforans is the predominant species associated with bacterial spot on tomato and X. euvesicatoria is the predominant species associated with the disease on pepper.
2. All four bacteria are strictly aerobic, gram-negative rods with a long whip-like flagellum (tail) that allows them to move in water, which allows them to invade wet plant tissue and cause infection.



How to prevent/cure the disease

1. The most effective management strategy is the use of pathogen-free certified seeds and disease-free transplants to prevent the introduction of the pathogen into greenhouses and field production areas. Inspect plants very carefully and reject infected transplants- including your own!
2. In transplant production greenhouses, minimize overwatering and handling of seedlings when they are wet.
3. Trays, benches, tools, and greenhouse structures should be washed and sanitized between seedlings crops.
4. Do not spray, tie, harvest, or handle wet plants as that can spread the disease.

7. ADVANTAGES & DISADVANTAGES

List of advantages

- The model that is being suggested here achieves extremely high categorization accuracy.
- It is also possible to train on and test very big datasets.\
- Very high resolution images can be scaled inside the proposal itself.

List of disadvantages

- The suggested model has a very high computational time requirement for both training and testing.
- This project's utilisation of a neural network design involves great levels of complexity.

8. APPLICATIONS

1. The image patterns were accurately classified using the trained network model.
2. The suggested model is employed not only for the classification of plant diseases but also for the classification of other image patterns, such as animal classification.

3. This project work application uses pattern recognition in addition to image classification.

9. CONCLUSIONS

The model here involves classifying images from datasets of fruits and vegetables. Observations made during model testing and training include the following:

- The number of epochs was increased to boost categorization accuracy.
- Different classification accuracies are found for various batch sizes.
- By adding more convolution layers, the accuracy is increased.
- The use of several dense layers significantly improved categorization accuracy.
- By adjusting the size of the kernel utilised in the convolution layer output, different accuracies can be achieved.
- The accuracy varies with the size of the train and test datasets.

10. FUTURE SCOPE

The model that is being provided in this project work can be expanded to recognise images. Using python to exe software, the complete model may be turned into application software. With the aid of the OpenCV Python library, real-time image classification, image recognition, and video processing are all made possible. This project's work can be expanded to include security applications like face, iris, and figure print recognition.

11. BIBILOGRAPHY

[1]. R Indumathi Leaf Disease Detection and Fertilizer Suggestion, IEEE Conference on System, Computation, Automation and Networking (ICSCAN), March 29–30, 2019, DOI: 10.1109/ICSCAN.2019.8878781.

[2]. International Journal of Engineering Trends and Applications (IJETA) - Volume 8 Issue 2, Mar-Apr 2021, P. Pandi Selvi and P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System."

[3]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395- 0056.

APPENDIX

A. Source Code (Jupyter notebook python code)

fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

```
#!/usr/bin/env python

# coding: utf-8

# In[1]: pwd

# In[2]: cd E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset

# # Apply ImageDataGenerator functionality to Train and Test set

# # Preprocessing # In[3]: from keras.preprocessing.image import ImageDataGenerator train_datagen =
```

ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_fli p=True) test_datagen = ImageDataGenerator(rescale=1) # In[4]: pwd

In[5]: x_train = train_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruitdataset/train',target_size=(128,128),batch_size=32,class_mode='categorical')

In[6]: x_test=test_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128), batch_size=32,class_mode='categorical') # # Import the models

In[7]: from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten

Initializing the models 10

In[8]: model=Sequential()

Add CNN Layers

In[9]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In[10]: x_train.class_indices

Add Pooling layer

In[11]: model.add(MaxPool2D(pool_size=(2,2)))

Add Flatten layer # In[12]: model.add(Flatten())

Add Dense Layer

In[21]: model.add(Dense(40, kernel_initializer='uniform',activation='relu')) model.add(Dense(20, kernel_initializer='random_uniform',activation='relu'))

Add Output Layer # In[24]: model.add(Dense(6,activation='softmax', kernel_initializer='random_uniform'))

Compile the model # In[25]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy']) # In[26]: len(x_train)

In[27]: 5384/32

Fit the Model

In[28]: model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_st eps=52,epochs=3)

Save the Model

In[29]: model.save("fruit.h5")

In[30]: ls

Test the Model

In[32]: from keras.preprocessing import image from tensorflow.keras.preprocessing.image import img_to_array from tensorflow.keras.models import load_model import numpy as np

In[33]: model = load_model("fruit.h5")

Test Apple_Healthy Class images

In[37]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a9b67bb7b105c___RS_HL 7708.JPG',target_size=(128,128)) 11

In[39]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0)

In[40]: pred = model.predict_classes(x)

In[41]: pred

In[45]: index =['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']

In[46]: print('the given image belogs to=',index[pred[0]])

Test Apple Black Rot class images # In[54]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Apple___Black_rot/0f3d45f4-e121-42cd-a5b6- be2f866a0574___JR_FrgE.S 2870.JPG',target_size=(128,128))

In[55]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

Test Corn Northern leaf Blight class images

In[56]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Corn_(maize)___Northern_Leaf_Blight/00a14441-7a62- 4034-bc40-b196aeab2785___RS_NLB 3932.JPG',target_size=(128,128))

In[57]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

Test Corn Healthy class images # In[58]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Corn_(maize)___healthy/0a68ef5a-027c-41ae-b227- 159dae77d3dd___R.S_HL 7969 copy.jpg',target_size=(128,128))

In[59]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]]) # # Test Peach Bacterial spot class images # In[60]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach___Bacterial_spot/00ddc106-692e-4c67-b2e8- 569c924caf49___Rutg._Bact.S 1228.JPG',target_size=(128,128)) 12 # In[61]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

Test Peach Healthy class images

In[62]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach___healthy/1a07ce54-f4fd-41cf-b088- 144f6bf71859___Rutg._HL 3543.JPG',target_size=(128,128))

In[63]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])