Fertilizers Recommendation System for Plant Disease Prediction

SB20220223216

IBM ID : chhaya.gosavi@cumminscollege.in

1 INTRODUCTION

1.1 Overview

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1.2 Purpose

The purpose of this application is to help farmers in detecting the diseases by observing the spots on the leaves, which in turn saves effort and labor costs.

2 LITERATURE SURVEY

2.1 Existing problem

Existing approaches or method: The plant diseases affect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labor and laboratories.

2.2 Proposed solution

What is the method or solution suggested by you?

- Use CNN to detect the disease

- Recommend Fertilizers and Pesticides to be used to prevent this disease

- Build a Web application to interact with the farmers

3   THEORITICAL ANALYSIS

3.1 Block diagram

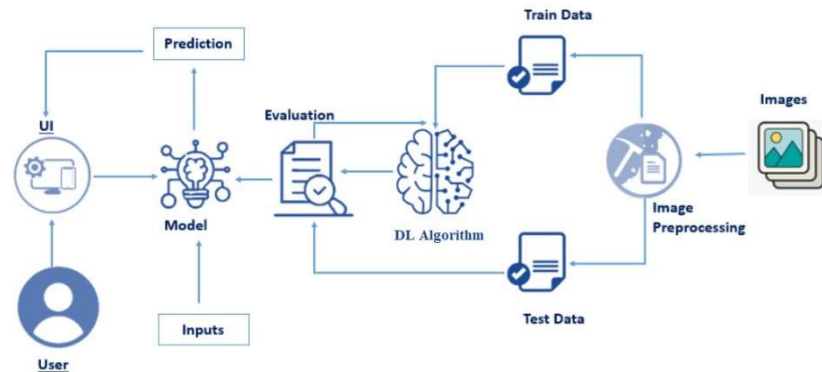Diagrammatic overview of the project.

Figure 1: Block diagram

3.2 Hardware / Software designing

Hardware requirements of the project [1]:

- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.

- System architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86 & M1; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2), 64-bit Power8/Power9, s390x (Linux on IBM Z & LinuxONE).

- Minimum 5 GB disk space to download and install.

Software requirements of the project

Following Tools and Packages needed to complete this project

- Anaconda Navigator - Jupiter notebook and spyder

- Tensor flow

- Keras

- Flask

- IBM Watson studio

- Numpy

- Pandas

4   EXPERIMENTAL INVESTIGATIONS

Analysis or the investigation made while working on the solution.

A web Application is built where:

- Farmers interact with the portal build

- Interacts with the user interface to upload images of diseased leaf

- Our model built analyses the Disease and suggests the farmer with fertilizers are to be used

To accomplish the above task, we did below activities:

- Downloaded the dataset.

- Classified the dataset into train and test sets.

- Added the neural network layers.

- Loaded the trained images and fit the model.

- Tested the model.

- Saved the model and its dependencies.

- Build a Web application using a flask that integrates with the model built.

5   FLOWCHART

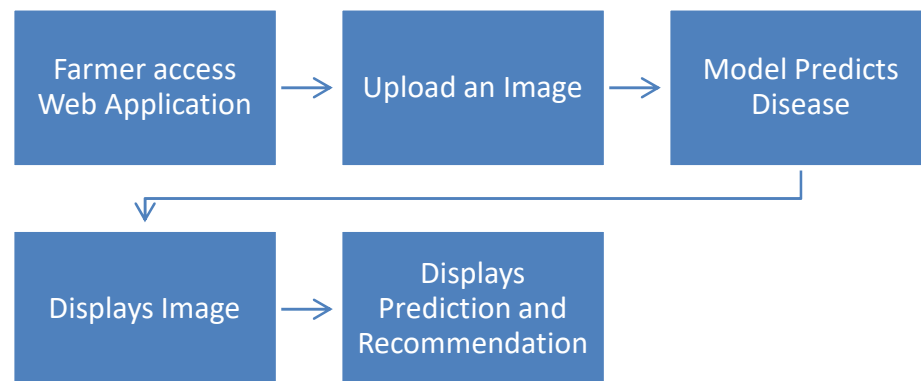Diagram showing the control flow of the solution

Figure 2: Flow Diagram

6   RESULT

Final findings (Output) of the project along with screenshots.
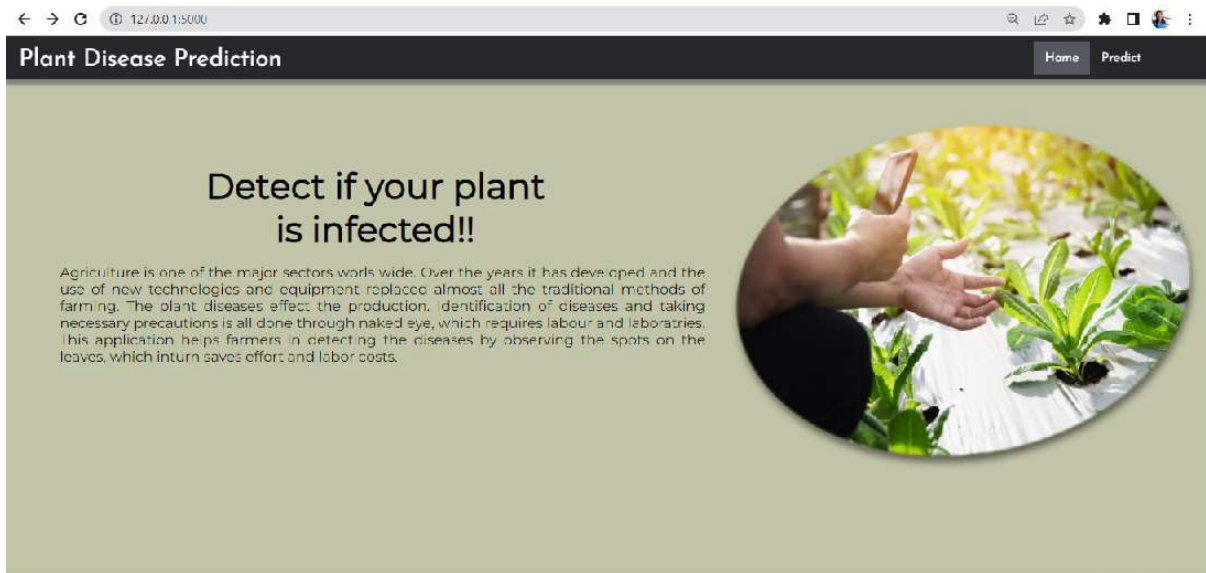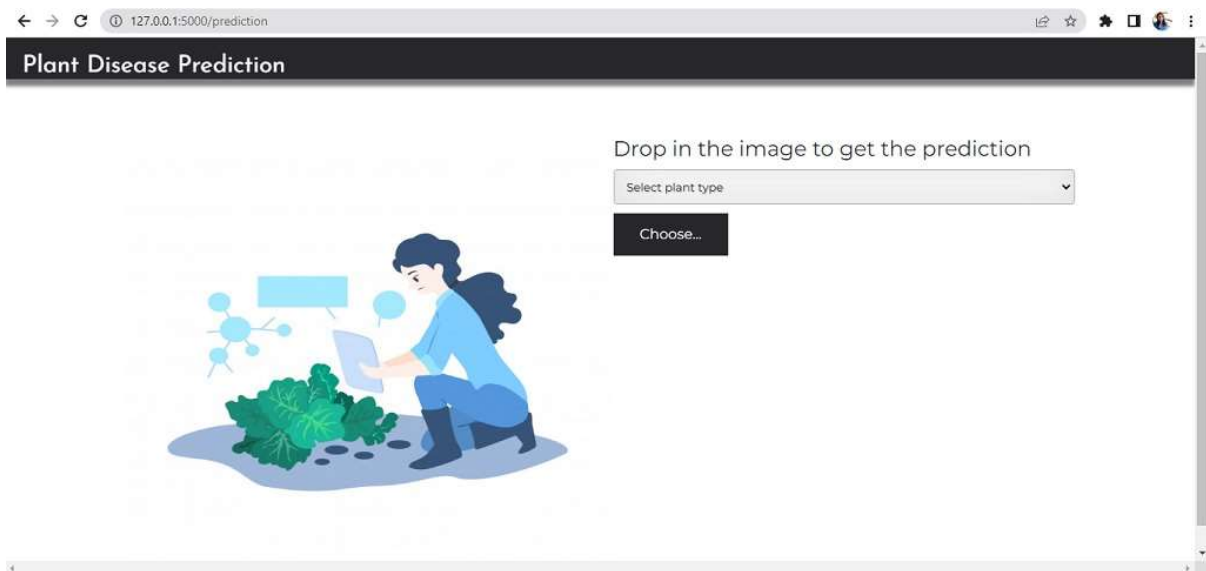

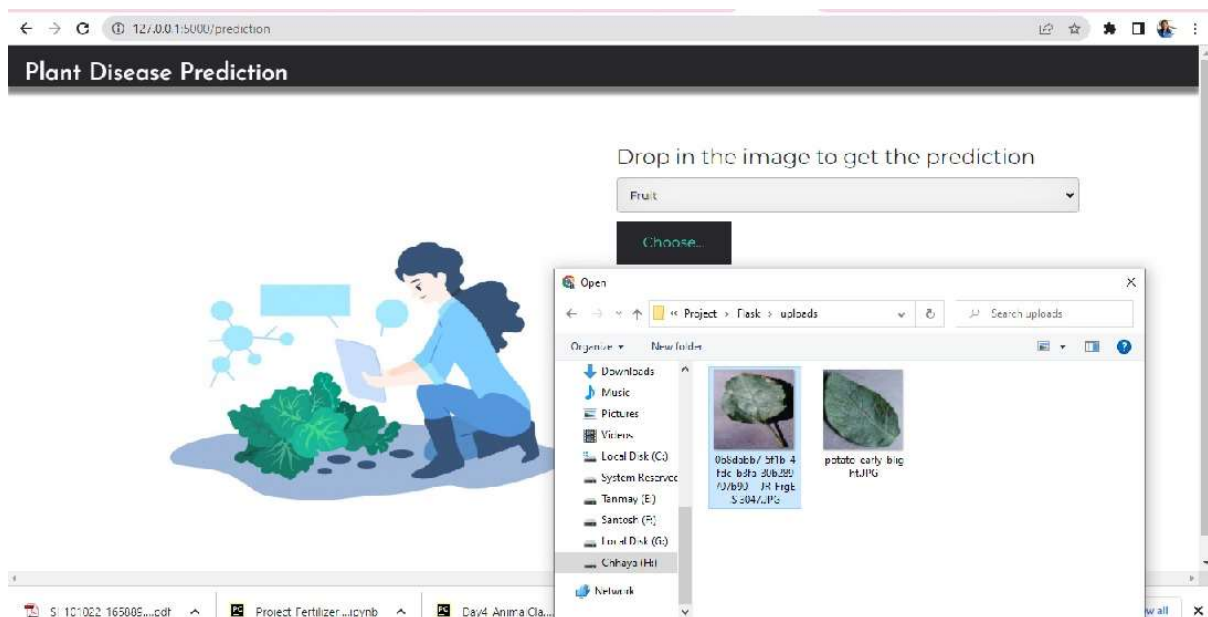Figure 3: Home Page

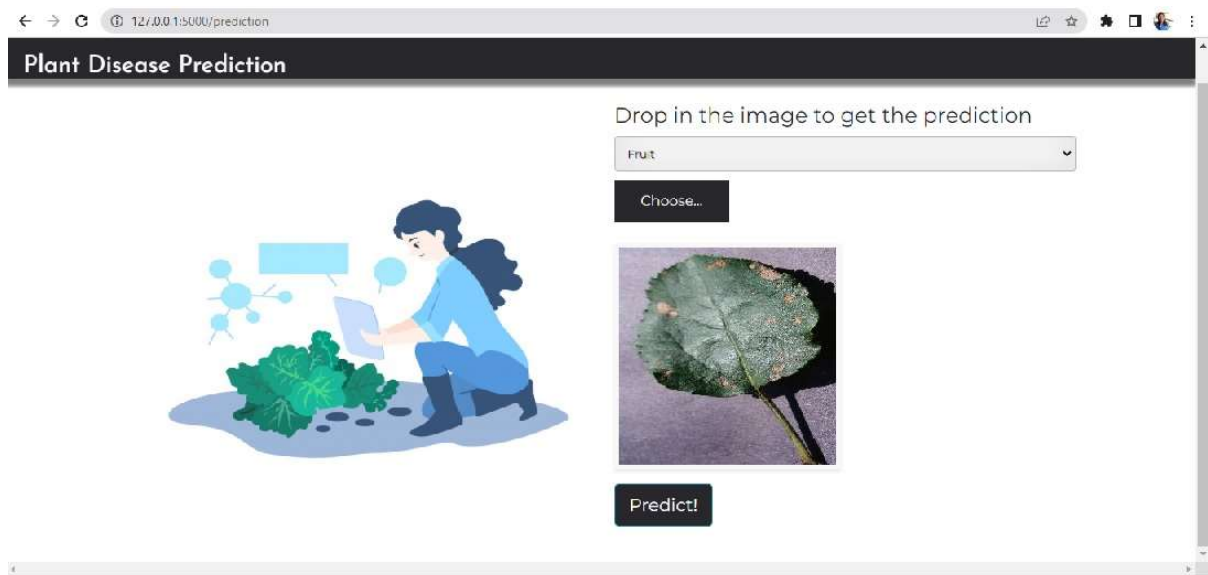Figure 4: Predict Page

Figure 5: Accepting Input

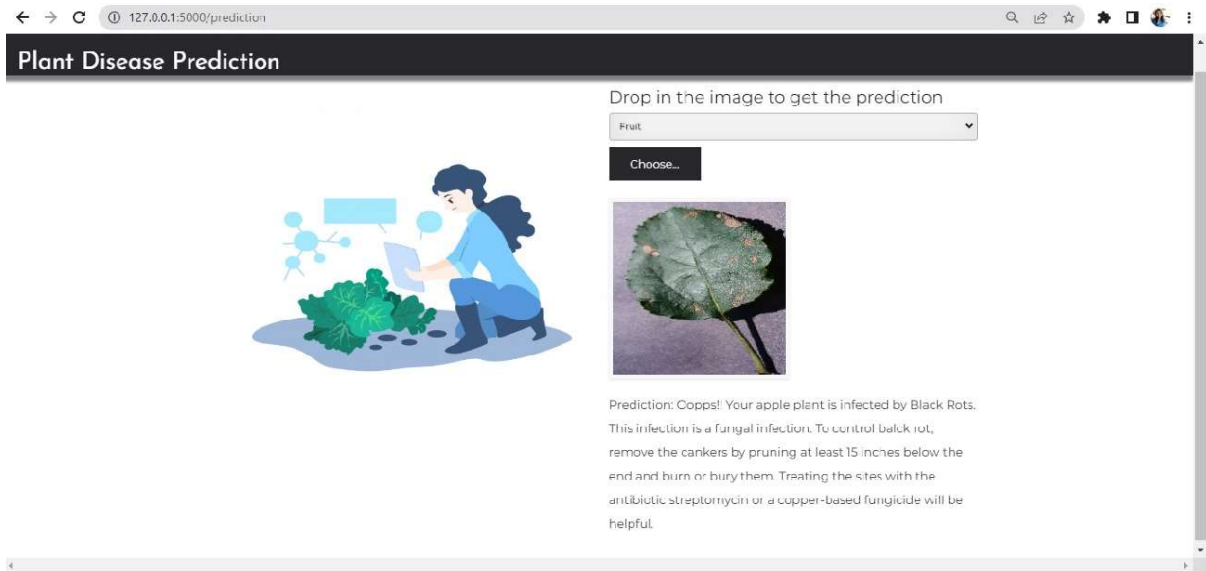Figure 6: Loaded Input Image

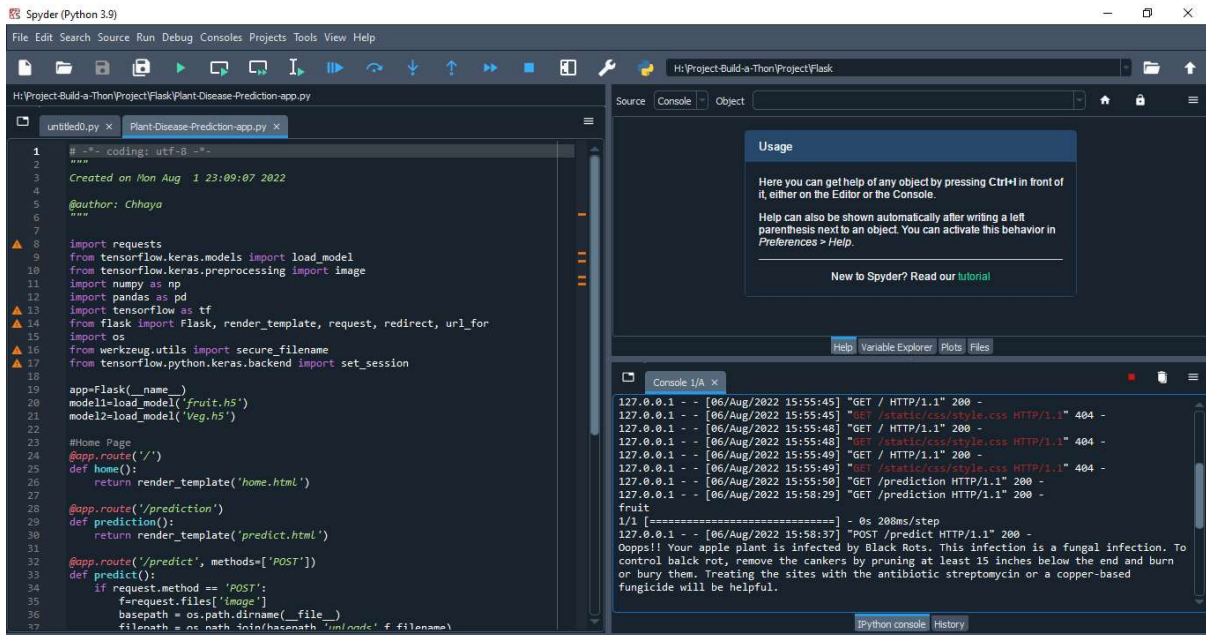Figure 7: Prediction and Recommendation

Figure 8: Program Execution Screenshot

7   ADVANTAGES & DISADVANTAGES

Advantages of the proposed solution:

-    User friendly GUI

-    Simple Web Application to predict the disease just by uploading an image of the leaf

-    Good accuracy

- Useful recommendations for the predicted disease

Disadvantages of the proposed solution:

- Usability is limited as model is trained for sample vegetables and fruits

8   APPLICATIONS

The areas where this solution can be applied:

- Mainly in Agriculture - By the farmers

- Nurseries

- House hold

- Gardens

9   CONCLUSION

Conclusion summarizing the entire work and findings.

Plant disease affects the crop production. This application is a small effort to in early disease detection. This will help to save the plant from the disease and increase the production.

10   FUTURE SCOPE

Enhancements that can be made in the future.

- More fruitful recommendations such as type and quantity of the fertilizers and pesticides can be given

- Mobile Application can be built instead of Web Application

- An IoT application cab be done like: automatic image capture, disease detection and alarm to the farmer

11   BIBILOGRAPHY

References of previous works or websites visited/books referred for

analysis about the project, solution previous findings etc.

[1]. https://docs.anaconda.com/anaconda/install/ [Hardware Requirements]

APPENDIX

A. Source Code

Attach the code for the solution built.

```
pwd
'/content'
cd /content/drive/MyDrive/IBM-Project/
```

Data Collection
```
cd /content/drive/MyDrive/IBM-Project/Project/
/content/drive/MyDrive/IBM-Project/Project

!unzip FRS.zip
```

Image Augmentation
```
[69]
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2
,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
[70]
pwd

'/content'
[71]
ls
```

```
_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM-
Project/Project/dataset/fruit-dataset/fruit-
dataset/train',target_size=(128,128),class_mode='categorical',batch_size=32)
Found 5384 images belonging to 6 classes.

x_test=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM-
Project/Project/dataset/fruit-dataset/fruit-
dataset/test',target_size=(128,128),class_mode='categorical',batch_size=32)
Found 1686 images belonging to 6 classes.

x_train.class_indices
{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
```

```
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}
```

CNN
```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Flatten())
```

ADD Dense Layers
```python
model.add(Dense(units = 40, activation='relu', kernel_initializer = 'uniform'))
model.add(Dense(units=20, activation='relu', kernel_initializer = 'random_uniform'))
model.add(Dense(units=6, activation='softmax', kernel_initializer = 'random_uniform'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)
```

```python
model.save('fruit.h5')
```

Test the Model
```
[74]
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
[75]
model=load_model('fruit.h5')
[76]
img=image.load_img("/content/drive/MyDrive/IBM-Project/Project/dataset/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/09e1e0ed-2535-496e-885d-a51999b51820___Rut._Bact.S 0988.JPG")
[77]
img
```

[78]
```
img=image.load_img("/content/drive/MyDrive/IBM-Project/Project/dataset/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/09e1e0ed-2535-496e-885d-a51999b51820___Rut._Bact.S 0988.JPG", target_size=(128,128))
```
[79]
```
Img
```


[80]
```
x=image.img_to_array(img)
```
[82]
```
x.shape
(128, 128, 3)
```
[83]
```
x=np.expand_dims(x,axis=0)
```
[84]
```
x.shape
(1, 128, 128, 3)
```
[85]
```
y=np.argmax(model.predict(x),axis=1)
```
[86]
```
y
array([4])
```
[88]
```
index = ['Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___Northern_Leaf_Blight', 'Corn_(maize)___healthy', 'Peach___Bacterial_spot', 'Peach___healthy']
```
[89]
```
index[y[0]]
'Peach___Bacterial_spot'
```

<mark>Model Building for Vegetable Disease Prediction</mark>

[2]
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2
,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
```
[3]
```
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/IBM-
Project/Project/dataset/Veg-dataset/Veg-
dataset/train_set',target_size=(128,128),class_mode='categorical',batch_size=16)
Found 11386 images belonging to 9 classes.
```

[4]
```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/IBM-
Project/Project/dataset/Veg-dataset/Veg-
dataset/test_set',target_size=(128,128),class_mode='categorical',batch_size=16)
Found 3416 images belonging to 9 classes.
```

[5]
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten
```
[6]
```
model=Sequential()
```
[7]
```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```
[8]
```
model.add(MaxPool2D(pool_size=(2,2)))
```
[9]
```
model.add(Flatten())
```
[10]
```
model.add(Dense(units = 300, activation='relu', kernel_initializer = 'uniform'))
model.add(Dense(units=150, activation='relu', kernel_initializer = 'uniform'))
model.add(Dense(units=75, activation='relu', kernel_initializer = 'uniform'))
model.add(Dense(units=9, activation='softmax', kernel_initializer = 'uniform'))
```
[11]
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accurac
y'])
```
[12]
```
model.fit_generator(x_train,steps_per_epoch=89,validation_data=x_test,validation
_steps=27,epochs=20)
```


```
model.save('veg.h5')
```
[14]
```
ls
drive/  sample_data/  veg.h5
```

Test the Model
[22]
```
pwd
```

```
'/content'
```
[15]

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```
[17]
```
model=load_model('veg.h5')
```
[52]
```
img=image.load_img("/content/drive/MyDrive/IBM-Project/Project/dataset/Veg-
dataset/Veg-dataset/test_set/Pepper,_bell___Bacterial_spot/ff7d08df-7fca-4a2e-
a2f4-b196a19e6442___JR_B.Spot 3237.JPG")
```
[53]
```
Img
```


[54]
```
img=image.load_img("/content/drive/MyDrive/IBM-Project/Project/dataset/Veg-
dataset/Veg-dataset/test_set/Pepper,_bell___Bacterial_spot/ff7d08df-7fca-4a2e-
a2f4-b196a19e6442___JR_B.Spot 3237.JPG", target_size=(128,128))
```
[55]
```
img
```


[60]
```
x=image.img_to_array(img)
```
[61]
```
x.shape
(128, 128, 3)
```
[62]
```
x=np.expand_dims(x,axis=0)
```
[63]
```
x.shape
(1, 128, 128, 3)
```
[64]

```
y=np.argmax(model.predict(x),axis=1)
[65]
y
array([0])
[66]
x_train.class_indices
{'Pepper,_bell___Bacterial_spot': 0,
 'Pepper,_bell___healthy': 1,
 'Potato___Early_blight': 2,
 'Potato___Late_blight': 3,
 'Potato___healthy': 4,
 'Tomato___Bacterial_spot': 5,
 'Tomato___Late_blight': 6,
 'Tomato___Leaf_Mold': 7,
 'Tomato___Septoria_leaf_spot': 8}
[67]
index=['Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy', 'Potato___Earl
y_blight',  'Potato___Late_blight', 'Potato___healthy', 'Tomato___Bacterial_spot
',
 'Tomato___Late_blight', 'Tomato___Leaf_Mold', 'Tomato___Septoria_leaf_spot']
[68]
index[y[0]]
'Pepper,_bell___Bacterial_spot'
[90]
ls
 Apple___Black_rot/                     fruit.h5
 Apple___healthy/                       Peach___Bacterial_spot/
'Corn_(maize)___healthy'/               Peach___healthy/
'Corn_(maize)___Northern_Leaf_Blight'/  veg.h5
```