# Fertilizers Recommendation System for Disease Prediction

Ghousiya Begum K, School of EEE, SASTRA University, Thanjavur-613401, India

ghousiyabegum@eie.sastra.edu

## 1. INTRODUCTION

### *1.1 Overview*

Due to variety of bacterial and fungal diseases, a major threat to food productivity has arisen. The diseases in plants yield losses up to 30% every year and early detection of infections is critical. Henceforth, a prognostic and precise identification of plant diseases is indispensable to ensure best quality and quantity. The diseases on plants and its degree of harm caused is due to the presence of pathogen varieties, improper irrigation, and insufficient plant protection techniques and reduce productivity by 10% to 95% [1, 2]. Image analysis have been playing a dynamic role in the identification of images and using automatic recognition of these images can be utilized to detect the plant diseases using machine learning and deep learning techniques. Machine learning techniques deliver low precision and improper classification. So, to improve the classification accuracy and for better prediction deep learning technique is applied for plant disease prediction [3].

So, an intelligent system is presented to identify different diseases on plants especially from leaves of fruits and vegetables. The proposed CNN techniques identify the diseases and suggest the precautions that can be taken for those diseases using Web application server.

### *1.2 Purpose*

The purpose of this project is to:

- To propose a CNN algorithm that gives better accuracy
- To apply CNN algorithm to the pre-processed plant disease image dataset
- To detect the plant disease by building web applications using the Flask framework that can help the farmers to interact with the user interface to upload images of diseased leaf
- To analyse the disease and to suggest the farmer the remedial measures and fertilizers to be used.

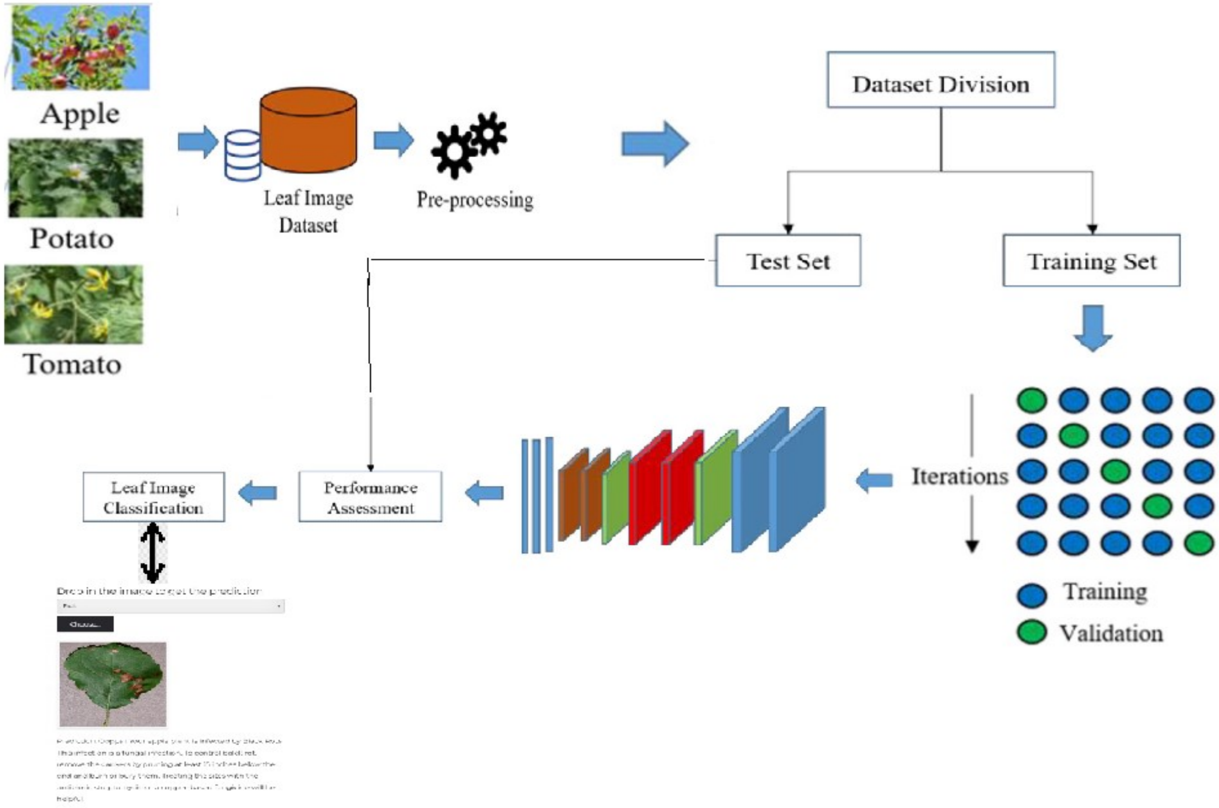
## 2. LITERATURE SURVEY

### *2.1 Existing problem*

It is both time-consuming and error prone to identify the plant disease manually and mitigate the severity of diseases. It entails an exhaustive knowledge of plant pathogens like fungus, bacteria and virus [4]. As an alternative, intelligent methods save both effort and time. Machine learning techniques deliver low precision and improper classification [3]. So deep learning technique has been implemented and the prediction is done using web application service [5].

### *2.2 Proposed solution*

The proposed methodology deals with the implementation of a sequential CNN model that is trained with the training dataset of the fruit and vegetables that were infected. Accordingly, they are categorized under 6 classes (fruit dataset) and 9 classes (vegetable dataset) depending on the type of disease with which the leaves of the fruit and vegetables are being infected.



[illegible] apple plant is infected by Black Rot. This infection is a fungal infection, to control black rot, remove the cankers by pruning at least 15 inches below the end and burn or bury them. Treating the sites with the [illegible] helpful.

**Fig. 1** Block Diagram of the Proposed Methodology

## 3. THEORITICAL ANALYSIS

### *3.1 Block diagram*

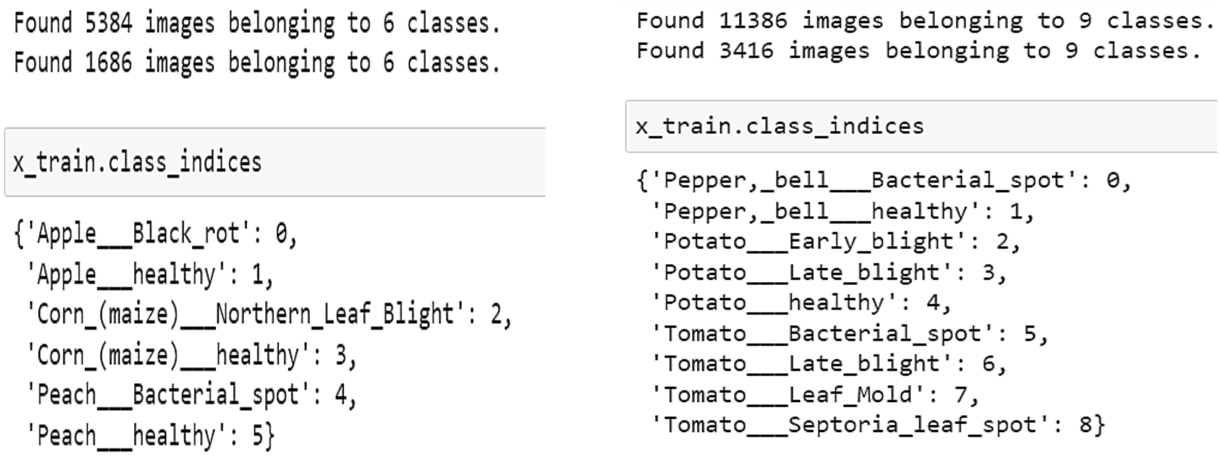
The above Fig. 1 depicts the blocks used in the proposed methodology. UI stands for user interface created using web application Flask app. The data is divided in testing and training dataset. The input images are pre-processed using image augmentation technique. And the classes are defined. A CNN model is developed using convolution layers, max pooling, flatten and dense layers and the model is trained using Adam optimizer. The updated weights and model hyperparameters are saved as .h5 files. The model which is locally deployed is then used to integrate with web application using Flask App where the inputs are fed in to the HTML page and the python script written in Spyder IDE predicts whether the plant leaves have been infected or not.

### *3.2 Hardware / Software designing*

The hardware requirement of the project is Intel core i5 processor, and the software tools needed are Anaconda Jupyter notebook or Watson Studio, Spyder IDE for Flask App integration, IBM Cloud.

## 4. EXPERIMENTAL INVESTIGATIONS

The dataset is image dataset comprising of plant diseases categorized for fruits and vegetables. Analysis or the investigation made while working on the solution. For fruits has the following data as mentioned below in Fig. 2a,



```
Found 5384 images belonging to 6 classes.
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}
```

**(a)**

```
Found 11386 images belonging to 9 classes.
Found 3416 images belonging to 9 classes.
```

```
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}
```

**(b)**

**Fig. 2** a) Fruit dataset class labels b) Vegetable dataset class labels

Here, under train dataset totally 5384 images are present and 1686 images for testing. The labels (6 classes) are given as above for multiclass classification. Similarly, for vegetables the details are mentioned are indicated in Fig. 2b.

## 5. FLOW CHART

The flowchart explicates the flow of the proposed methodology as shown in Fig. 3.

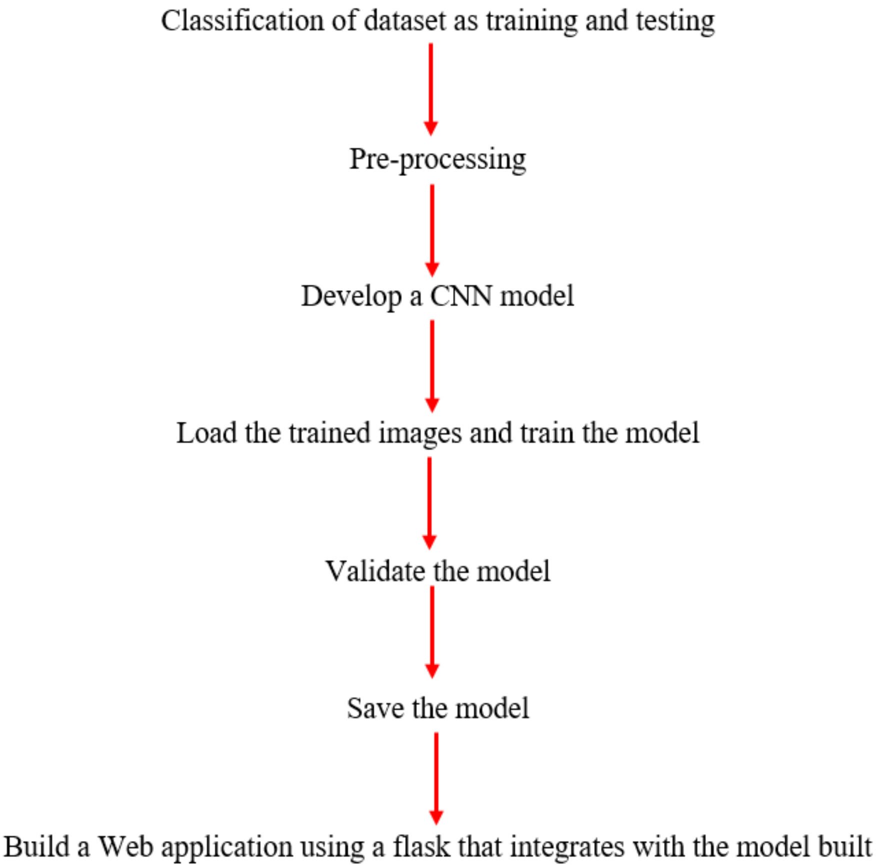
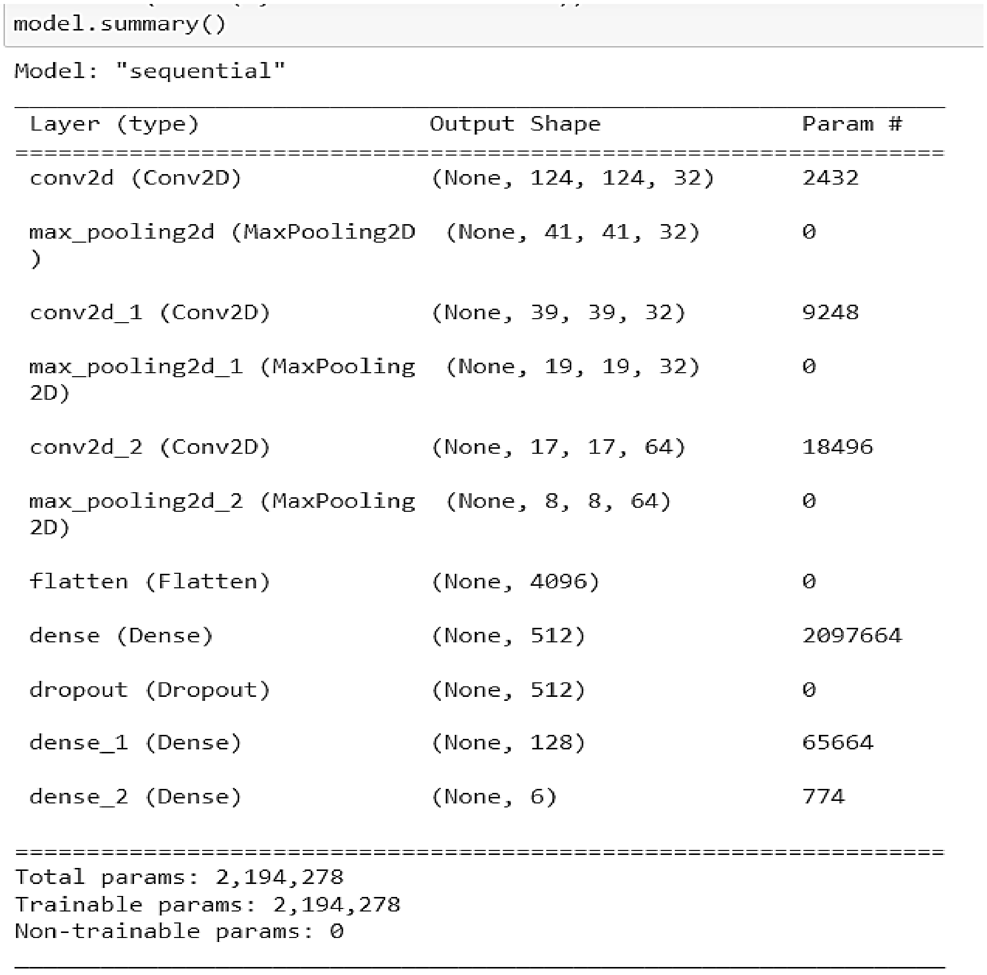

Classification of dataset as training and testing
Pre-processing
Develop a CNN model
Load the trained images and train the model
Validate the model
Save the model
Build a Web application using a flask that integrates with the model built

**Fig. 3** Flowchart

## 6. RESULT

### a) Fruit Dataset CNN Sequential Model Building

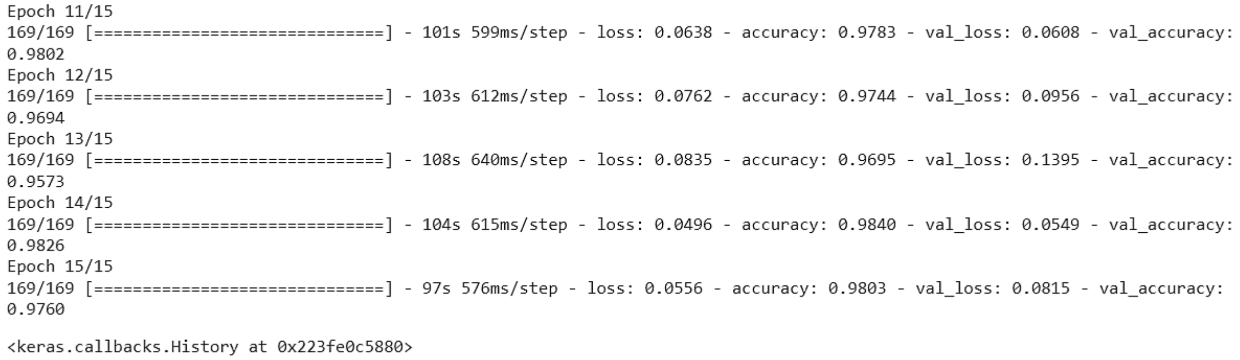
Here the image target_size is (128,128) and batch_size is 32 and image is RGB image. The Fig. 4 depicts the sequential CNN model structure and the number of parameters used.



```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 124, 124, 32)      2432      
                                                                 
 max_pooling2d (MaxPooling2D  (None, 41, 41, 32)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 39, 39, 32)        9248      
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 19, 19, 32)       0         
 2D)                                                             
                                                                 
 conv2d_2 (Conv2D)           (None, 17, 17, 64)        18496     
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 8, 8, 64)         0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 4096)              0         
                                                                 
 dense (Dense)               (None, 512)               2097664   
                                                                 
 dropout (Dropout)           (None, 512)               0         
                                                                 
 dense_1 (Dense)             (None, 128)               65664     
                                                                 
 dense_2 (Dense)             (None, 6)                 774       
                                                                 
=================================================================
Total params: 2,194,278
Trainable params: 2,194,278
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 4** Model summary

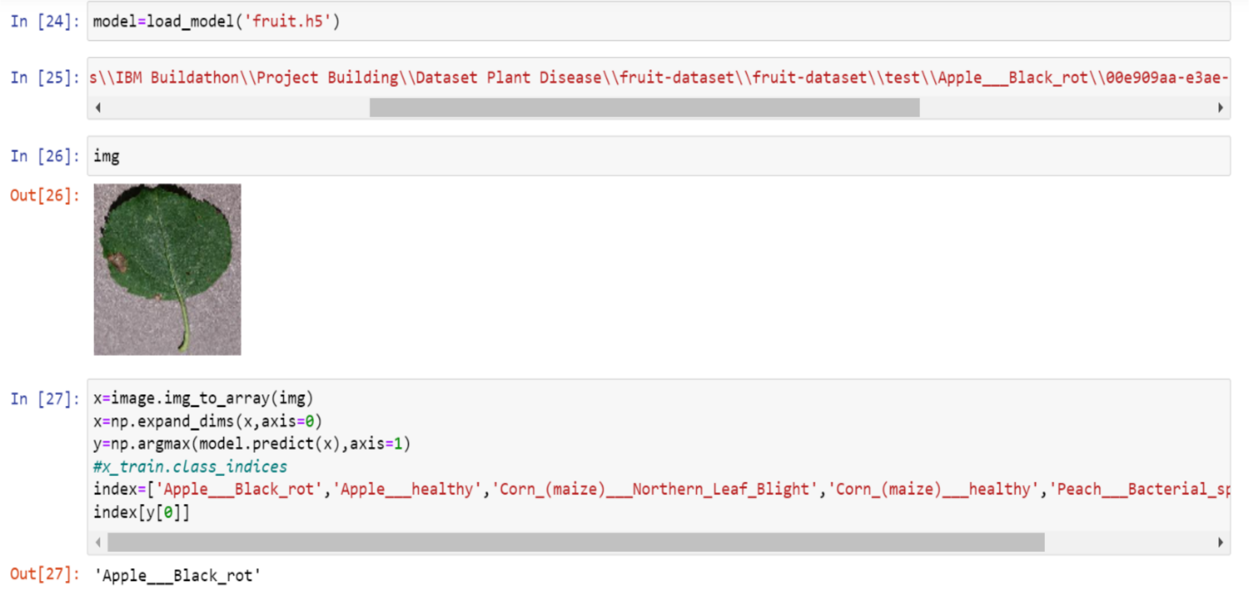
Fig. 5 shows the training of the CNN model and the accuracy obtained is 98% indicating that the model is very well trained.



```
Epoch 11/15
169/169 [==============================] - 101s 599ms/step - loss: 0.0638 - accuracy: 0.9783 - val_loss: 0.0608 - val_accuracy:
0.9802
Epoch 12/15
169/169 [==============================] - 103s 612ms/step - loss: 0.0762 - accuracy: 0.9744 - val_loss: 0.0956 - val_accuracy:
0.9694
Epoch 13/15
169/169 [==============================] - 108s 640ms/step - loss: 0.0835 - accuracy: 0.9695 - val_loss: 0.1395 - val_accuracy:
0.9573
Epoch 14/15
169/169 [==============================] - 104s 615ms/step - loss: 0.0496 - accuracy: 0.9840 - val_loss: 0.0549 - val_accuracy:
0.9826
Epoch 15/15
169/169 [==============================] - 97s 576ms/step - loss: 0.0556 - accuracy: 0.9803 - val_loss: 0.0815 - val_accuracy:
0.9760

<keras.callbacks.History at 0x223fe0c5880>
```

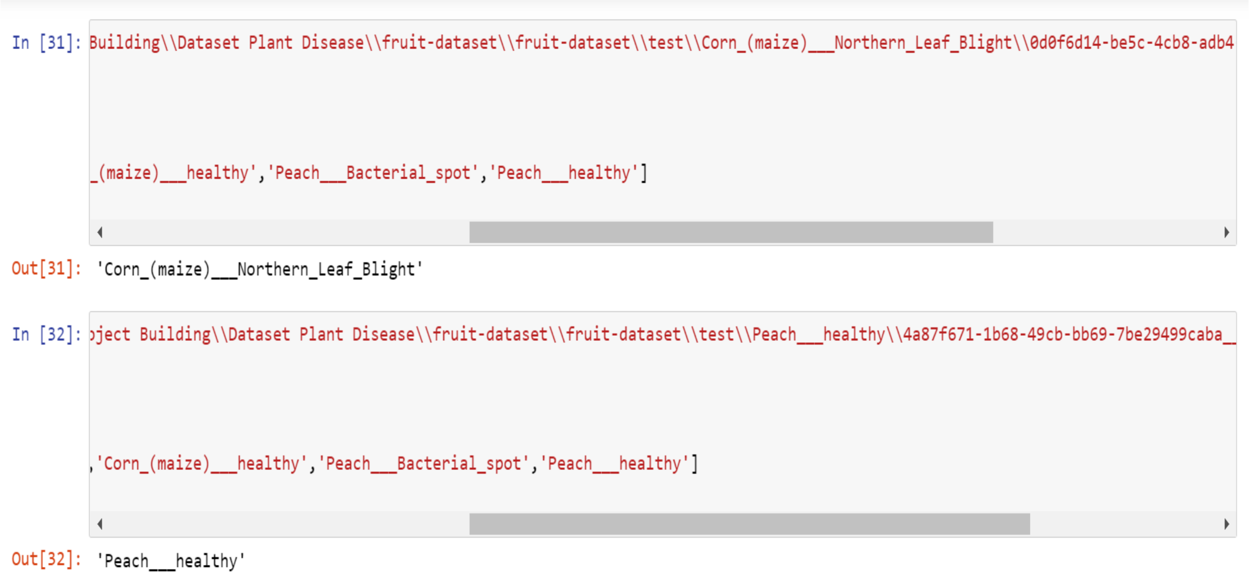
**Fig. 5** Trained model metrics

Figs. 6 & 7 represents that the model has correctly identified the class to which the input test leaves belong to. The model identifies the plant disease correctly as it is trained well.



```
In [24]: model=load_model('fruit.h5')

In [25]: s\\IBM Buildathon\\Project Building\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Apple___Black_rot\\00e909aa-e3ae-

In [26]: img

Out[26]:

In [27]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         y=np.argmax(model.predict(x),axis=1)
         #x_train.class_indices
         index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp
         index[y[0]]

Out[27]: 'Apple___Black_rot'
```

**Fig. 6** Predicted output for Image 1

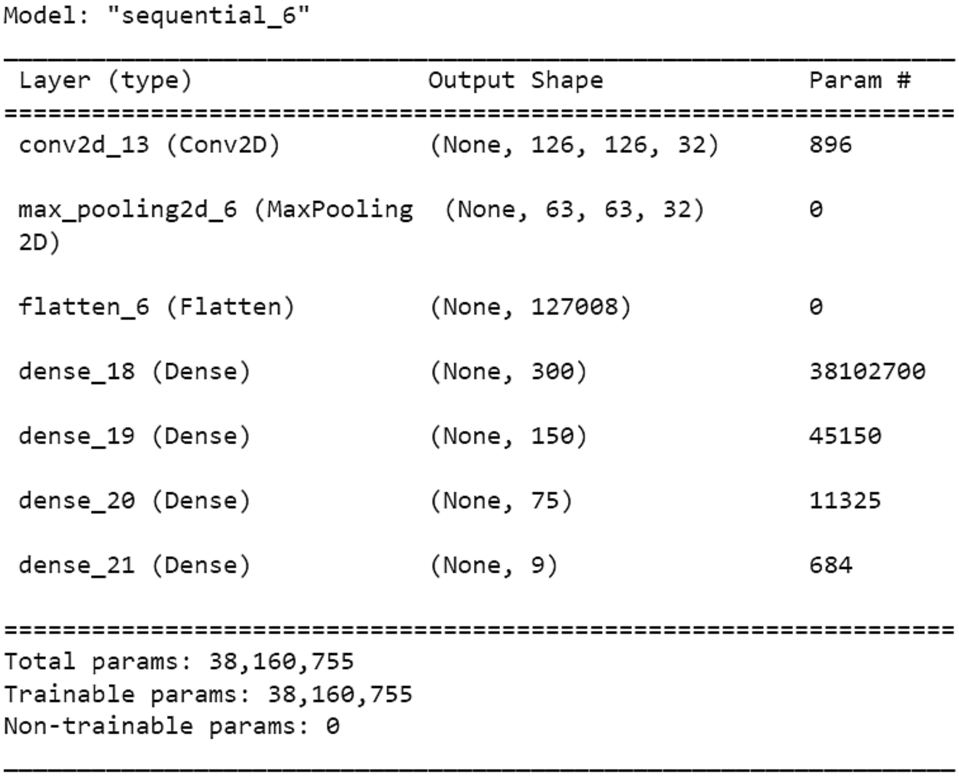


```
In [31]: Building\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Corn_(maize)___Northern_Leaf_Blight\\0d0f6d14-be5c-4cb8-adb4



         _(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']

Out[31]: 'Corn_(maize)___Northern_Leaf_Blight'

In [32]: oject Building\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Peach___healthy\\4a87f671-1b68-49cb-bb69-7be29499caba_



         ,'Corn_(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']

Out[32]: 'Peach___healthy'
```

**Fig. 7** Predicted output for Image 2 & 3

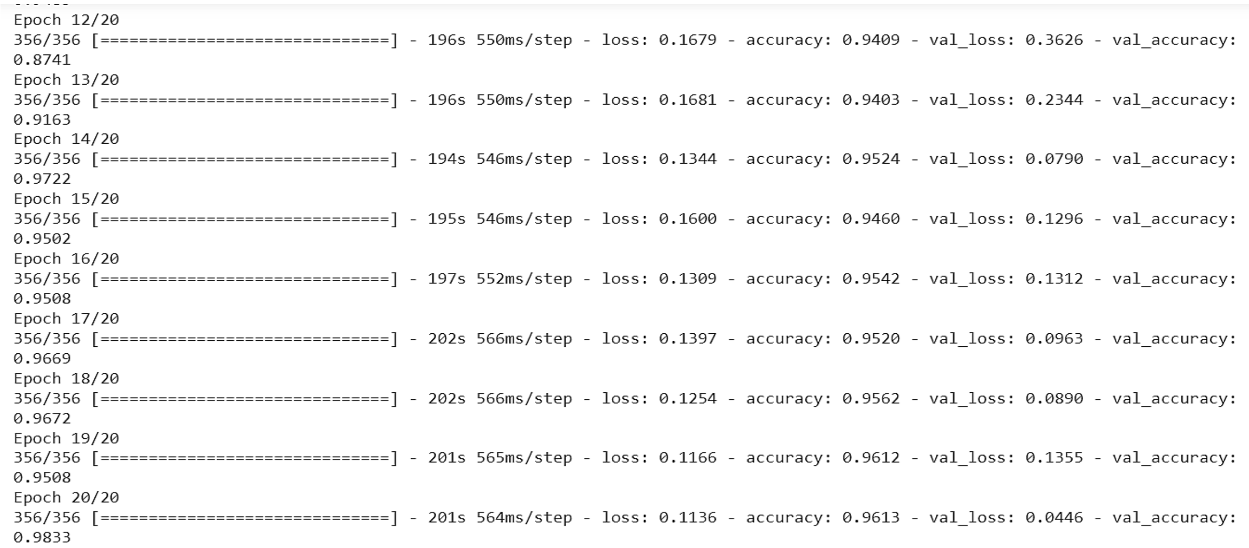
### b) Vegetable Dataset- CNN Sequential Model Building

Similarly, Figs. 8-11 displays the model, fitting of the model and accuracy, class labels, and predicted results for vegetable dataset.



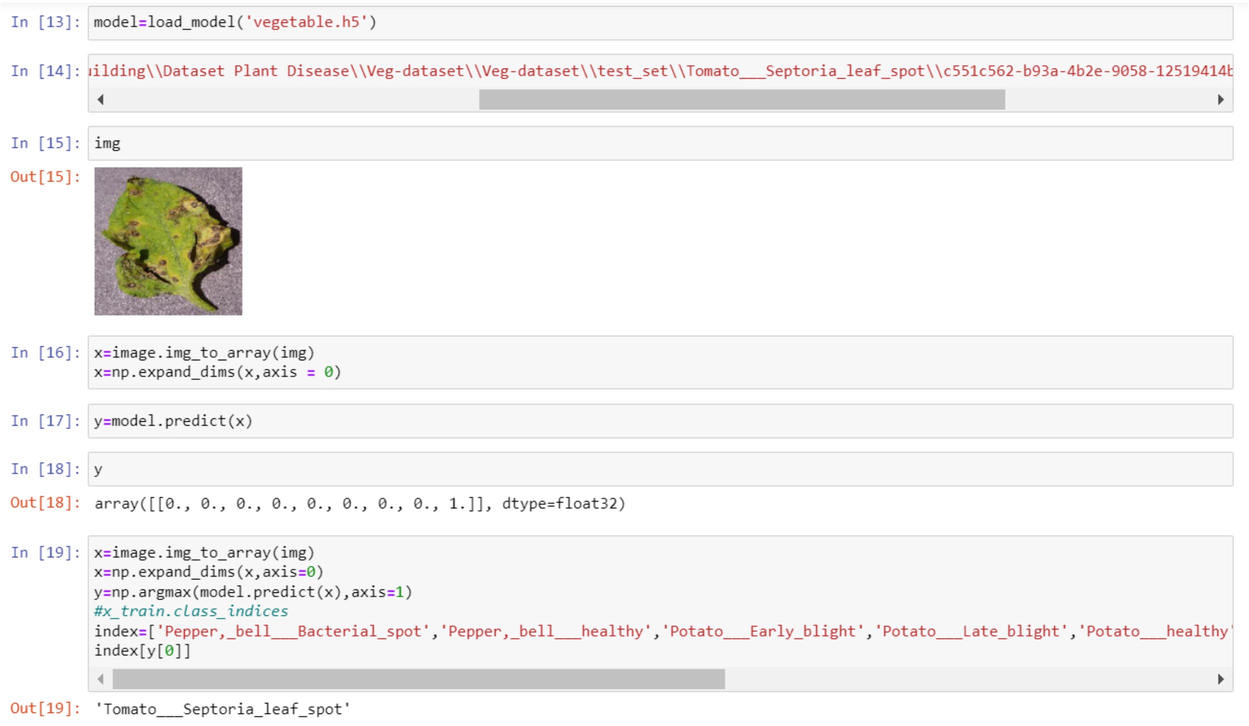
```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_13 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_6 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_6 (Flatten)         (None, 127008)            0

 dense_18 (Dense)            (None, 300)               38102700

 dense_19 (Dense)            (None, 150)               45150

 dense_20 (Dense)            (None, 75)                11325

 dense_21 (Dense)            (None, 9)                 684

=================================================================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 8** Model summary



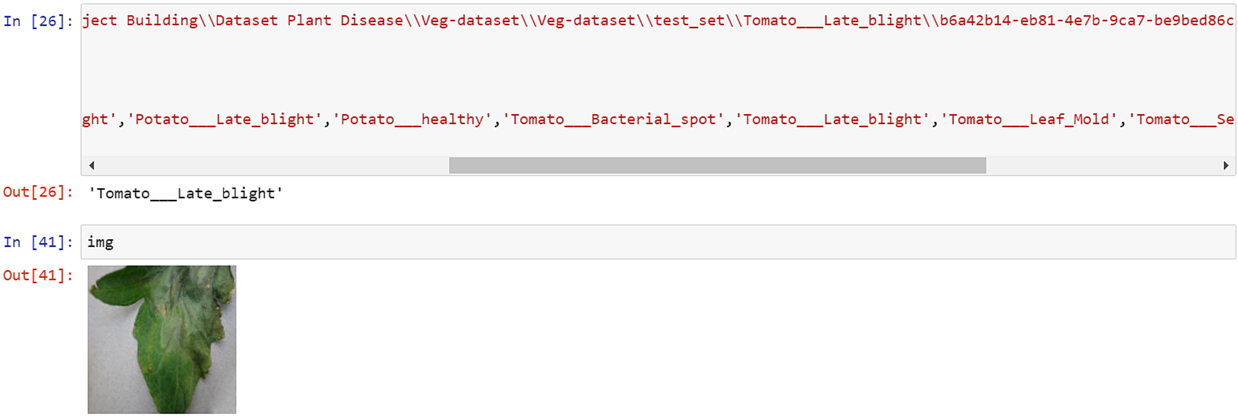
```
Epoch 12/20
356/356 [==============================] - 196s 550ms/step - loss: 0.1679 - accuracy: 0.9409 - val_loss: 0.3626 - val_accuracy:
0.8741
Epoch 13/20
356/356 [==============================] - 196s 550ms/step - loss: 0.1681 - accuracy: 0.9403 - val_loss: 0.2344 - val_accuracy:
0.9163
Epoch 14/20
356/356 [==============================] - 194s 546ms/step - loss: 0.1344 - accuracy: 0.9524 - val_loss: 0.0790 - val_accuracy:
0.9722
Epoch 15/20
356/356 [==============================] - 195s 546ms/step - loss: 0.1600 - accuracy: 0.9460 - val_loss: 0.1296 - val_accuracy:
0.9502
Epoch 16/20
356/356 [==============================] - 197s 552ms/step - loss: 0.1309 - accuracy: 0.9542 - val_loss: 0.1312 - val_accuracy:
0.9508
Epoch 17/20
356/356 [==============================] - 202s 566ms/step - loss: 0.1397 - accuracy: 0.9520 - val_loss: 0.0963 - val_accuracy:
0.9669
Epoch 18/20
356/356 [==============================] - 202s 566ms/step - loss: 0.1254 - accuracy: 0.9562 - val_loss: 0.0890 - val_accuracy:
0.9672
Epoch 19/20
356/356 [==============================] - 201s 565ms/step - loss: 0.1166 - accuracy: 0.9612 - val_loss: 0.1355 - val_accuracy:
0.9508
Epoch 20/20
356/356 [==============================] - 201s 564ms/step - loss: 0.1136 - accuracy: 0.9613 - val_loss: 0.0446 - val_accuracy:
0.9833
```

**Fig. 9** Training accuracy



```
In [13]: model=load_model('vegetable.h5')

In [14]: ilding\\Dataset Plant Disease\\Veg-dataset\\Veg-dataset\\test_set\\Tomato___Septoria_leaf_spot\\c551c562-b93a-4b2e-9058-12519414b

In [15]: img
Out[15]:

In [16]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis = 0)

In [17]: y=model.predict(x)

In [18]: y
Out[18]: array([[0., 0., 0., 0., 0., 0., 0., 0., 1.]], dtype=float32)

In [19]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         y=np.argmax(model.predict(x),axis=1)
         #x_train.class_indices
         index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___Late_blight','Potato___healthy'
         index[y[0]]
Out[19]: 'Tomato___Septoria_leaf_spot'
```

**Fig. 10** Predicted output for Image 4

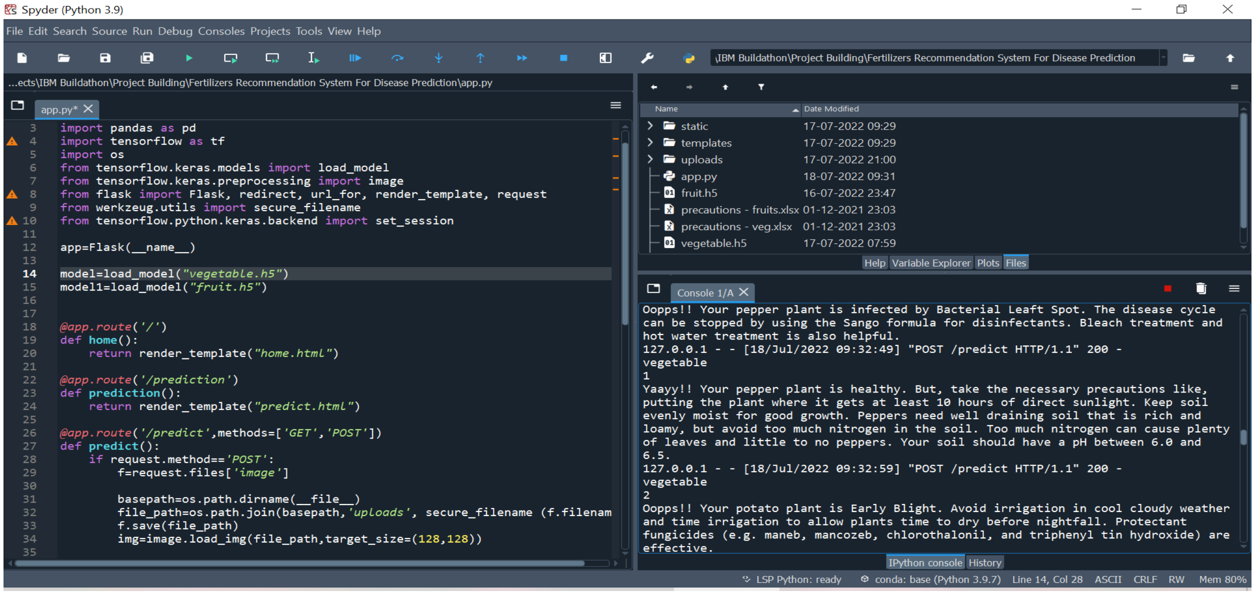


```
In [26]: ject Building\\Dataset Plant Disease\\Veg-dataset\\Veg-dataset\\test_set\\Tomato___Late_blight\\b6a42b14-eb81-4e7b-9ca7-be9bed86c



         ght','Potato___Late_blight','Potato___healthy','Tomato___Bacterial_spot','Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Se

Out[26]: 'Tomato___Late_blight'

In [41]: img
Out[41]:
```

**Fig. 11** Predicted output for Image 5

### c) Flask App python script on Spyder IDE

For web application building, Flask App is used where the HTML page as well as python script are needed to display the predicted results to the farmers. Fig. 12 shows the python script app.py written at the back end which redirects to the url http://127.0.0.1:5000, where the farmers will load the fruit and vegetable leaves images so as to predict their condition.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
\IBM Buildathon\Project Building\Fertilizers Recommendation System For Disease Prediction
...ects\IBM Buildathon\Project Building\Fertilizers Recommendation System For Disease Prediction\app.py
app.py*
Help Variable Explorer Plots Files
Console 1/A
IPython console History
LSP Python: ready conda: base (Python 3.9.7) Line 14, Col 28 ASCII CRLF RW Mem 80%

```python
import pandas as pd
import tensorflow as tf
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask, redirect, url_for, render_template, request
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app=Flask(__name__)

model=load_model("vegetable.h5")
model1=load_model("fruit.h5")


@app.route('/')
def home():
    return render_template("home.html")

@app.route('/prediction')
def prediction():
    return render_template("predict.html")

@app.route('/predict',methods=['GET','POST'])
def predict():
    if request.method=='POST':
        f=request.files['image']

        basepath=os.path.dirname(__file__)
        file_path=os.path.join(basepath,'uploads', secure_filename (f.filenam
        f.save(file_path)
        img=image.load_img(file_path,target_size=(128,128))

```

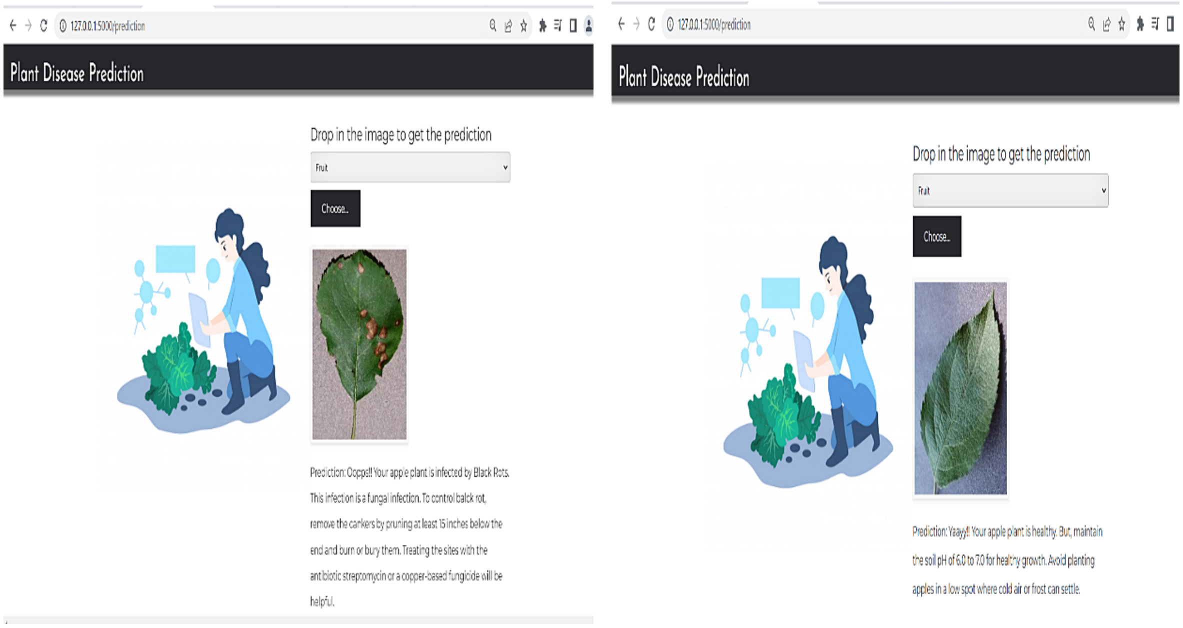
| Name | Date Modified |
|---|---|
| static | 17-07-2022 09:29 |
| templates | 17-07-2022 09:29 |
| uploads | 17-07-2022 21:00 |
| app.py | 18-07-2022 09:31 |
| fruit.h5 | 16-07-2022 23:47 |
| precautions - fruits.xlsx | 01-12-2021 23:03 |
| precautions - veg.xlsx | 01-12-2021 23:03 |
| vegetable.h5 | 17-07-2022 07:59 |

```
Oopps!! Your pepper plant is infected by Bacterial Leaft Spot. The disease cycle
can be stopped by using the Sango formula for disinfectants. Bleach treatment and
hot water treatment is also helpful.
127.0.0.1 - - [18/Jul/2022 09:32:49] "POST /predict HTTP/1.1" 200 -
vegetable
1
Yaayy!! Your pepper plant is healthy. But, take the necessary precautions like,
putting the plant where it gets at least 10 hours of direct sunlight. Keep soil
evenly moist for good growth. Peppers need well draining soil that is rich and
loamy, but avoid too much nitrogen in the soil. Too much nitrogen can cause plenty
of leaves and little to no peppers. Your soil should have a pH between 6.0 and
6.5.
127.0.0.1 - - [18/Jul/2022 09:32:59] "POST /predict HTTP/1.1" 200 -
vegetable
2
Oopps!! Your potato plant is Early Blight. Avoid irrigation in cool cloudy weather
and time irrigation to allow plants time to dry before nightfall. Protectant
fungicides (e.g. maneb, mancozeb, chlorothalonil, and triphenyl tin hydroxide) are
effective.
```

**Fig. 12** Python script for Flask App

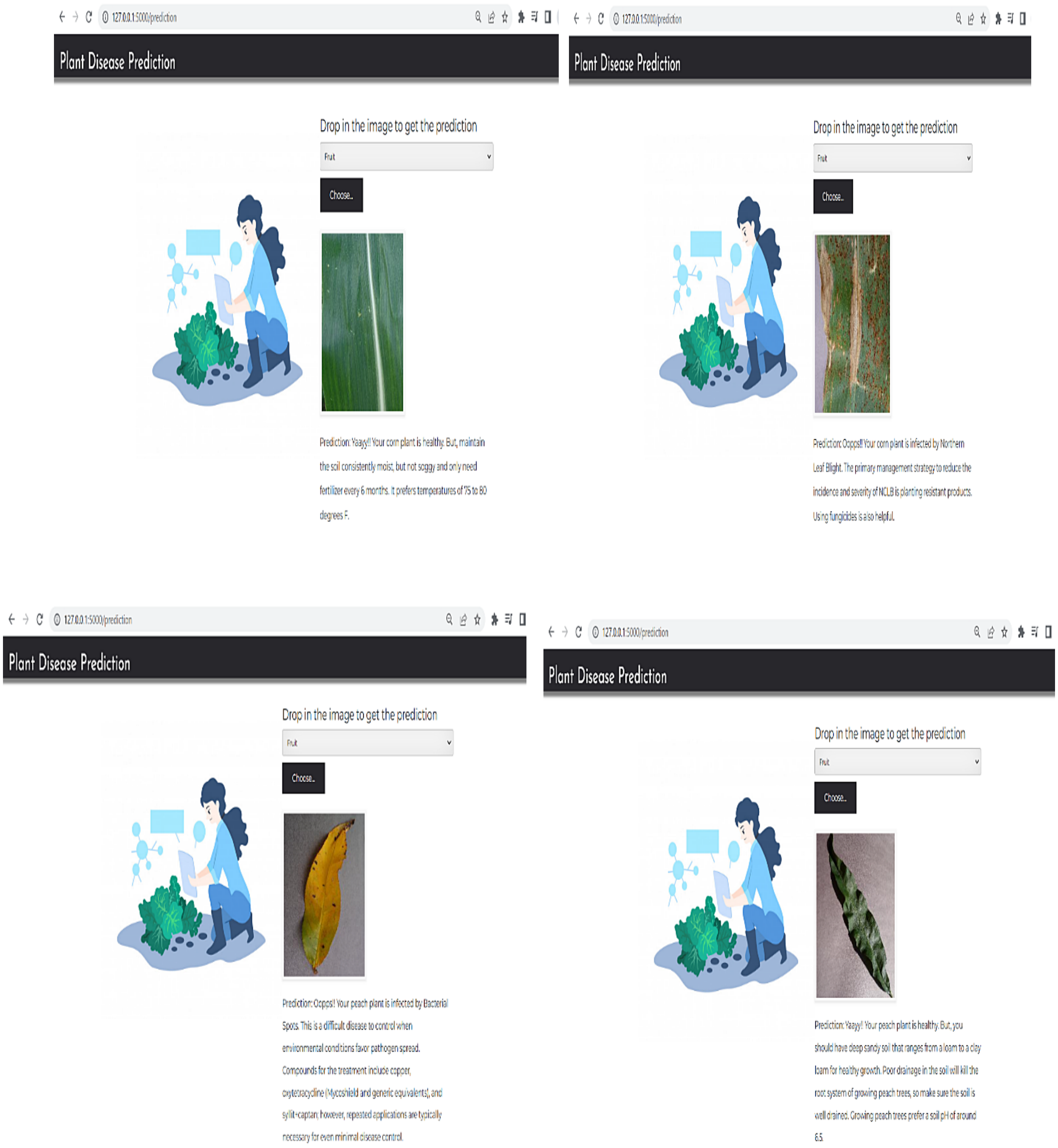
## d) Prediction results on HTML page

The prediction results displayed on the HTML page can be visualized as shown in Figs. 13 for Fruit dataset.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Fruit
Choose..
127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Fruit
Choose..

Prediction: Oopps!! Your apple plant is infected by Black Rots. This infection is a fungal infection. To control balck rot, remove the cankers by pruning at least 15 inches below the end and burn or bury them. Treating the sites with the antibiotic streptomycin or a copper-based fungicide will be helpful.

Prediction: Yaayy!! Your apple plant is healthy. But, maintain the soil pH of 6.0 to 7.0 for healthy growth. Avoid planting apples in a low spot where cold air or frost can settle.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Fruit
Choose...

Prediction: Yaayy!! Your corn plant is healthy. But, maintain the soil consistently moist, but not soggy and only need fertilizer every 6 months. It prefers temperatures of 75 to 80 degrees F.

Prediction: Oopps!! Your corn plant is infected by Northern Leaf Blight. The primary management strategy to reduce the incidence and severity of NCLB is planting resistant products. Using fungicides is also helpful.

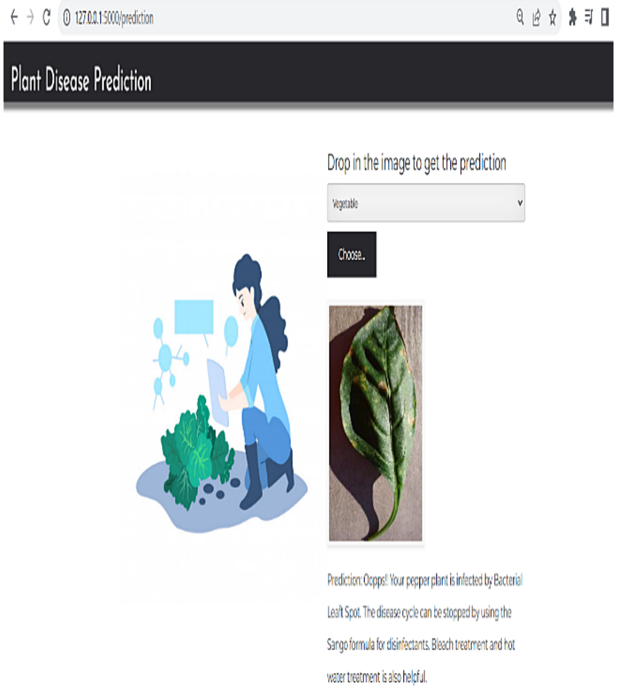
Prediction: Oopps! Your peach plant is infected by Bacterial Spots. This is a difficult disease to control when environmental conditions favor pathogen spread. Compounds for the treatment include copper, oxytetracycline (Mycoshield and generic equivalents), and syllit+captan; however, repeated applications are typically necessary for even minimal disease control.

Prediction: Yaayy! Your peach plant is healthy. But, you should have deep sandy soil that ranges from a loam to a clay loam for healthy growth. Poor drainage in the soil will kill the root system of growing peach trees, so make sure the soil is well drained. Growing peach trees prefer a soil pH of around 6.5.

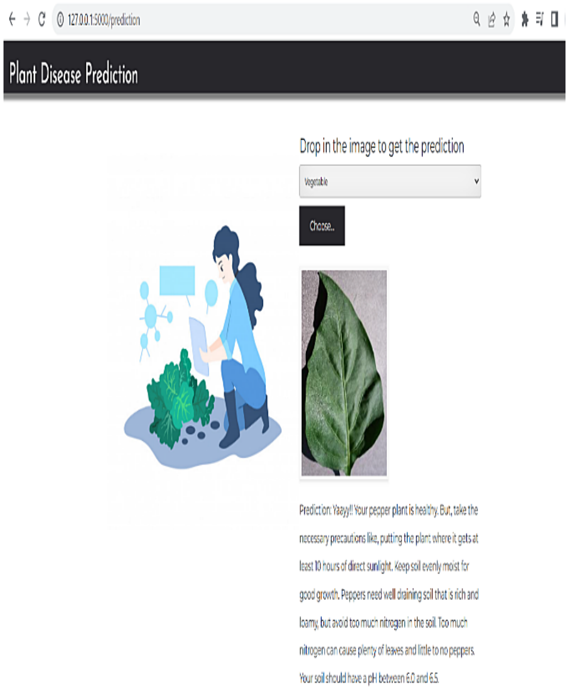
**Fig. 13** Predicted outputs on HTML page for fruit dataset

*For Vegetables:*

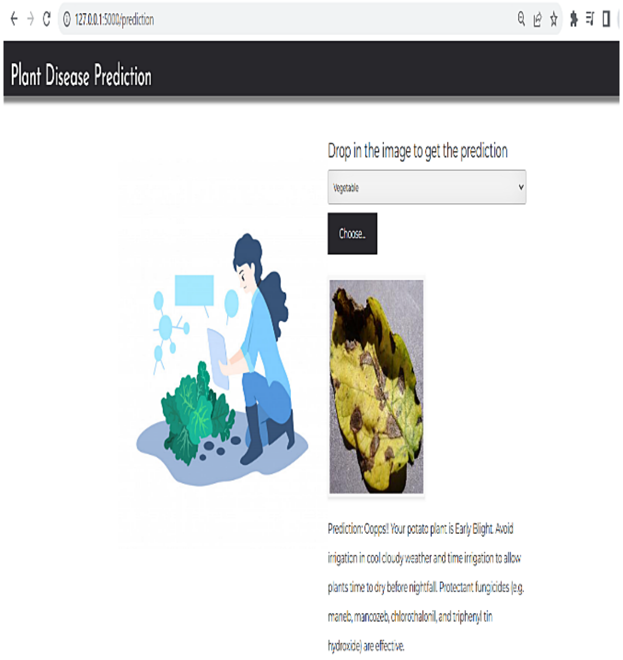
Similarly for vegetables the predicted results are displayed below in Fig. 14 for 6 classes they belong to.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose...

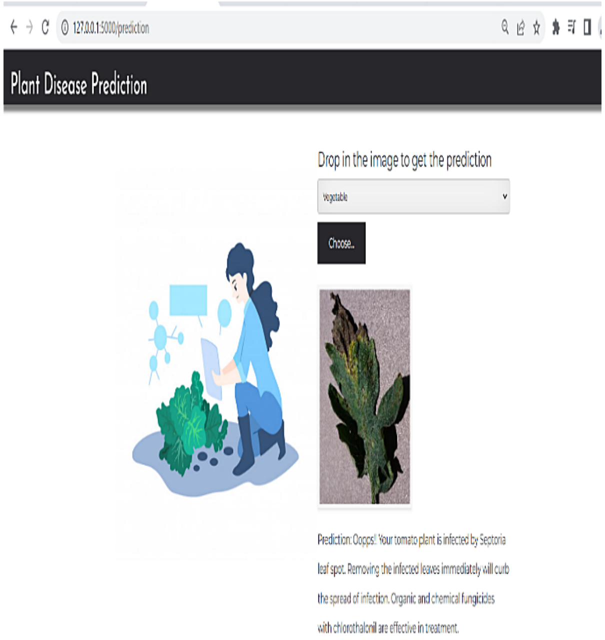
Prediction: Oopps!! Your pepper plant is infected by Bacterial Leaft Spot. The disease cycle can be stopped by using the Sango formula for disinfectants. Bleach treatment and hot water treatment is also helpful.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose...

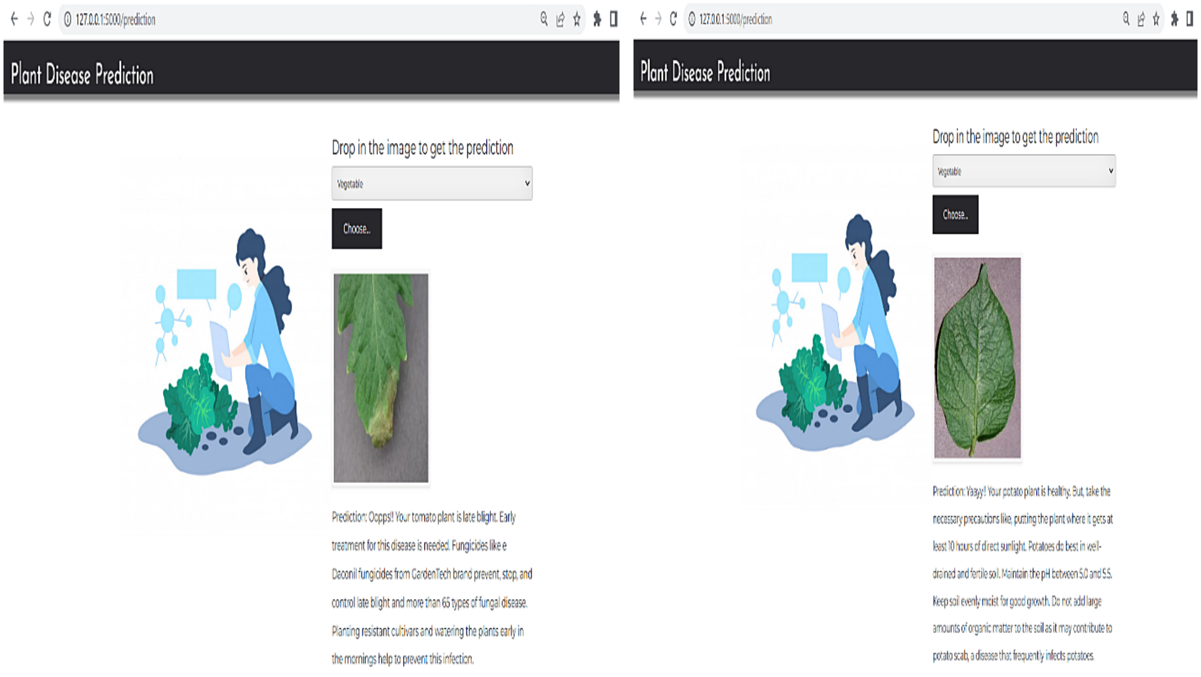
Prediction: Yaayy!! Your pepper plant is healthy. But, take the necessary precautions like, putting the plant where it gets at least 10 hours of direct sunlight. Keep soil evenly moist for good growth. Peppers need well draining soil that is rich and loamy, but avoid too much nitrogen in the soil. Too much nitrogen can cause plenty of leaves and little to no peppers. Your soil should have a pH between 6.0 and 6.5.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose...

Prediction: Oopps!! Your potato plant is Early Blight. Avoid irigation in cool cloudy weather and time irrigation to allow plants time to dry before nightfall. Protectant fungicides (e.g. maneb, mancozeb, chlorothalonil, and tripheryl tin hydroxide) are effective.


127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose...

Prediction: Oopps!! Your tomato plant is infected by Septoria leaf spot. Removing the infected leaves immediately will curb the spread of infection. Organic and chemical fungicides with chlorothalonil are effective in treatment.



127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose..

Prediction: Oopps!! Your tomato plant is late blight. Early treatment for this disease is needed. Fungicides like e Daconil fungicides from GardenTech brand prevent, stop, and control late blight and more than 65 types of fungal disease. Planting resistant cultivars and watering the plants early in the mornings help to prevent this infection.

127.0.0.1:5000/prediction
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose..

Prediction: Yaayy! Your potato plant is healthy. But, take the necessary precautions like, putting the plant where it gets at least 10 hours of direct sunlight. Potatoes do best in well-drained and fertile soil. Maintain the pH between 5.0 and 5.5. Keep soil evenly moist for good growth. Do not add large amounts of organic matter to the soil as it may contribute to potato scab, a disease that frequently infects potatoes.

**Fig. 14** Predicted outputs on HTML page for vegetable dataset

As CUH is 0, instead of IBM cloud deployment, local deployment is done and integrated with Flask App for prediction.

## 7. ADVANTAGES & DISADVANTAGES

The advantages of the proposed solution are

- It intelligently detects the plant diseases without any manual supervision.
- It analyses the disease and suggests the farmer which fertilizer to be used.

The disadvantages of the proposed solution are

- If the CNN has numerous layers, and the image dataset is huge, then the training process encounters computational burden if the processing unit doesn't possess a good GPU.
- Deficiency in dataset leads to poor training of the CNN model.

## 8. APPLICATIONS

The areas where this solution can be applied are,

- Crop prediction
- Plant disease prediction
- Agriculture
- Smart farming

## 9. CONCLUSIONS

As to recognize the diverse plant diseases prognostically and to reduce the severity of diseases, an intelligent disease identification technique is presented in this work. The proposed project gives insights about automatic plant disease detection in terms of symptoms presented by diseases like bacterial spots, color change in images. Convolutional neural network (CNN) models have been considered to be tremendously powerful and capable in locating visual patterns in images. Hence, a simple CNN model is implemented which after training is used to detect the class in which the plant image belongs to and thereby predicting the disease the plant was subjected. Based on the degree and type of disease suggestion are given to the farmers using a web application app done using Flask. With the increase in available computing power, it has been made effectual, precise, low-cost, and instantaneous system to identify the plant diseases. Thus, the suggested system is integrated with web application, that triggers an alert and suggestions to the farmer as soon as he uploads the images of the plant leaves that are infected.

## 10. FUTURE SCOPE

The future scope of the project will be to,

- Develop better CNN models for accurate prediction
- Increase the plant image dataset for better training.
- To deploy the model on IBM cloud with huge dataset and to have a real time prediction using camera
- To develop a chatbot for the farmers for decision making

- A camera can be used to capture the images of the leaves thereby automatically triggering an alert to the farmer to intercede.

## 11 BIBILOGRAPHY

1. Lucas, G.B.; Campbell, C.L.; Lucas, L.T. Causes of plant diseases. In Introduction to Plant Diseases; Springer: Berlin/Heidelberg, Germany, 1992; pp. 9–14.

2. Shirahatti, J.; Patil, R.; Akulwar, P. A survey paper on plant disease identification using machine learning approach. In Proceedings of the 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 15–16 October 2018; pp. 1171–1174.

3. Rajathi, N., Parameswari, P. (2022). Early-Stage Prediction of Plant Leaf Diseases Using Deep Learning Models. In: Uddin, M.S., Bansal, J.C. (eds) Computer Vision and Machine Learning in Agriculture, Volume 2. Algorithms for Intelligent Systems. Springer, Singapore.

4. Orchi H, Sadik M, Khaldoun M. On Using Artificial Intelligence and the Internet of Things for Crop Disease Detection: A Contemporary Survey. *Agriculture*. 2022; 12(1):9.

5. V. Tiwari, R. C. Joshi, M. K. Dutta, Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images, Ecological Informatics, Volume 63, 2021, 101289.

APPENDIX A.

Source Code

```
x_train=train_datagen.flow_from_directory('C:\\Users\\91978\\Desktop\\Python
Projects\\IBM Buildathon\\Project Building\\Dataset Plant Disease\\fruit-
dataset\\fruit-dataset\\train',


target_size=(128,128),class_mode='categorical',batch_size=32)

x_test=test_datagen.flow_from_directory('C:\\Users\\91978\\Desktop\\Python
Projects\\IBM Buildathon\\Project Building\\Dataset Plant Disease\\fruit-
dataset\\fruit-dataset\\test',
target_size=(128,128),class_mode='categorical',batch_size=32)
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, (5, 5),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Convolution2D(32, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(64, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(128,activation='relu'))
model.add(Dense(6,activation='softmax'))
model.summary()
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics='accuracy')
len(x_train)
model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=52,epochs=15)
model.save('fruit.h5')
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("C:\\Users\\91978\\Desktop\\Python Projects\\IBM Buildathon\\Project Building\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Apple___Black_rot\\00e909aa-e3ae-4558-9961-336bb0f35db3___JR_FrgE.S 8593.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]]
```

Similarly, for vegetable data set the same codings have been written with different CNN model

```
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.add(Dense(300,kernel_initializer = 'uniform',activation ='relu'))

model.add(Dense(150, kernel_initializer ='uniform',activation ='relu'))

model.add(Dense(75, kernel_initializer ='uniform',activation ='relu'))

model.add(Dense(9, kernel_initializer ='uniform',activation ='softmax'))

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

len(x_train)

model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=20)
```