SI-101022-1658899310

Fertilizers Recommendation System For Disease Prediction

1. Introduction:

a. 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 diseaseson plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest precautions that can be taken for those diseases.

b. Purpose:

To Detect and recognize the plant diseases and to **recommend fertilizer**,it is necessary to provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new **fertilizers Recommendation System** for crop **disease prediction**.

2. LITERATURE SURVEY:

a. Existing problem:

Adequate mineral nutrition is central to crop production. However, it can also exert considerable Influence on disease development.Fertilizer application can increase or decrease development of diseases caused by different pathogens, and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen. The effects of mineral nutrition on plant disease and the mechanisms responsible for those effectshave been dealt with comprehensively elsewhere.In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.

Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an external display or an Android application. The application was created to measure the approximate values of temperature, humidity,and moisture sensors that were programmed into a micro-controller to manage the amount of water.

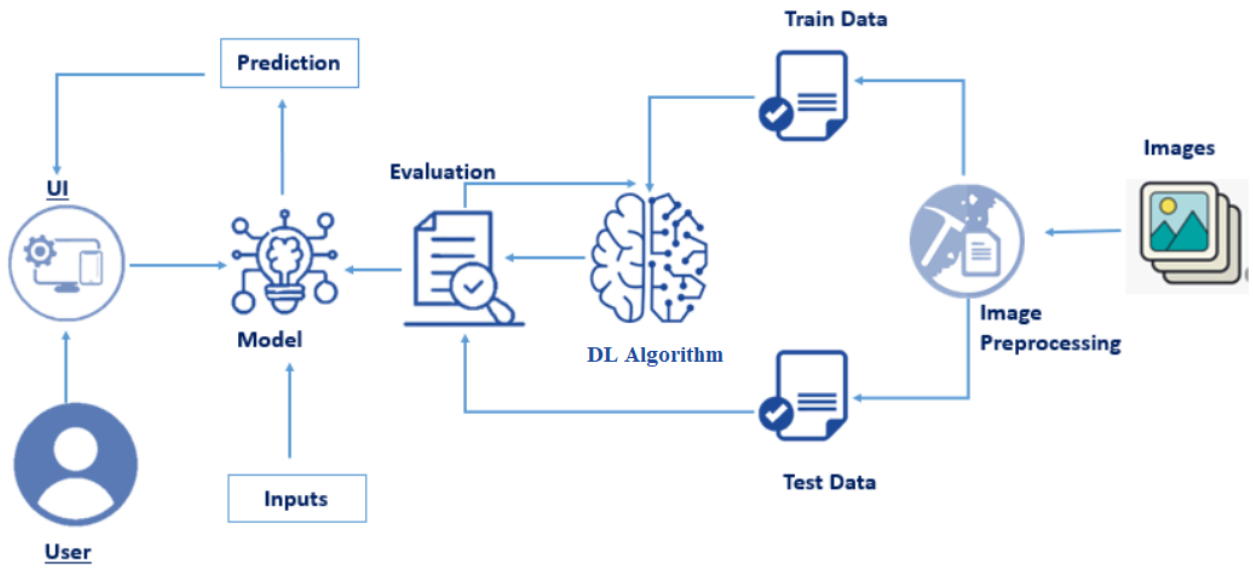
b. Proposed solution:

Web Application is built where :

1.	Farmers interact with the portal build
2.	Interacts with the user interface to upload images of diseased leaf
3.	Our model built analyses the Disease and suggests the farmer with fertilizers that are to be used.

1. THEORITICAL ANALYSIS:

a. **Block diagram :**



b. ***Hardware / Software designing:***

To complete this project you should have the following software and packages.

Software:

1.Anaconda Navigator

2.py charm

3.Visual studio code

4.Jupiter notebook

5.IBM Watson studio

Packages:

1.Tensor flow

2. Keras

3. Flask

4.numpy

5.Pandas

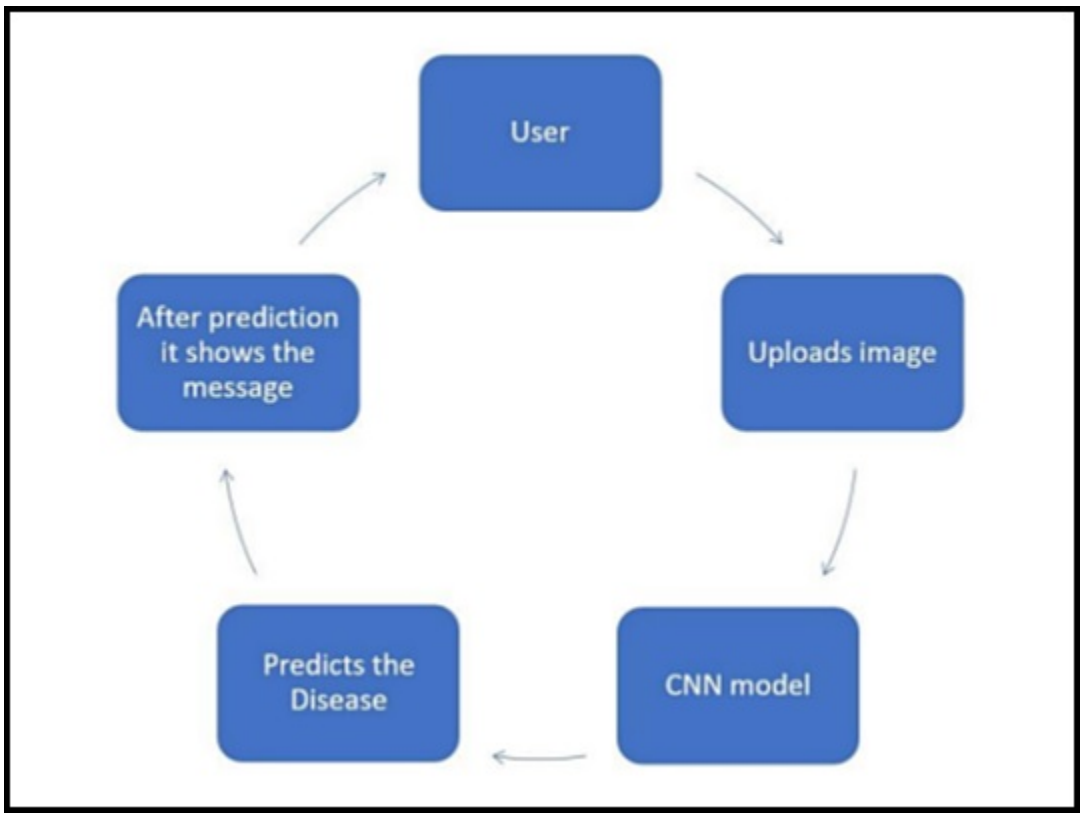
By using the above listed software and packages ,we build this application to take the input (image) from the farmer and detects whether theplant is infected or not. Here we use Deep learning techniques and give the out put to the user (Farmer).

2. EXPERIMENTAL INVESTIGATIONS :

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. Analysis or the investigation made while working on the solution.

3. FLOWCHART :



To accomplish the above task you must complete the below activities and tasks :

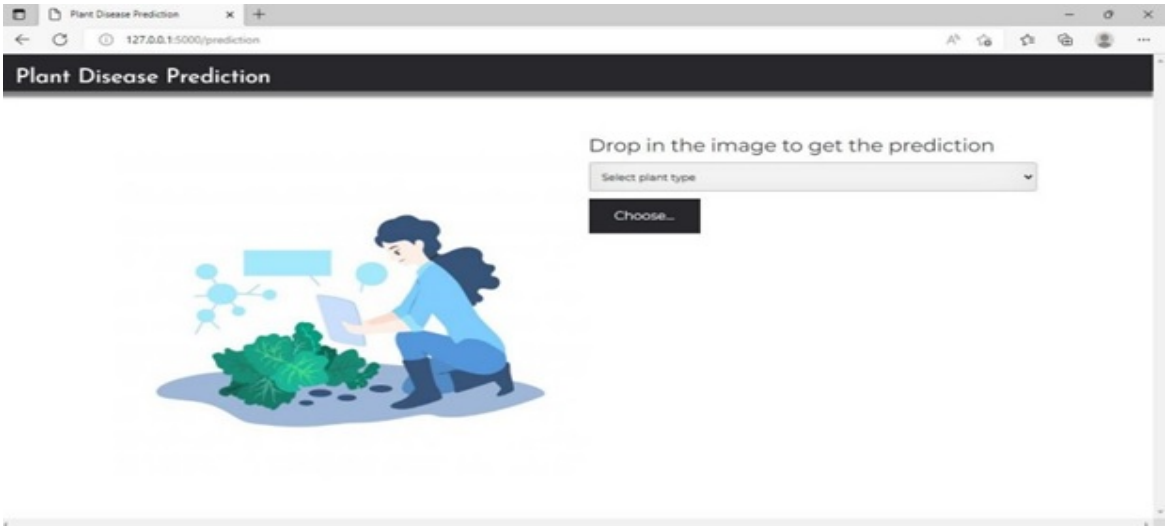
1. Download the dataset.
2. Classify the dataset into train and test sets.
3. Add the neural network layers.

4. Load the trained images and fit the model.
5. Test the model.
6. Save the model and its dependencies.
7. Build a Web application using a flask that integrates with the model built.

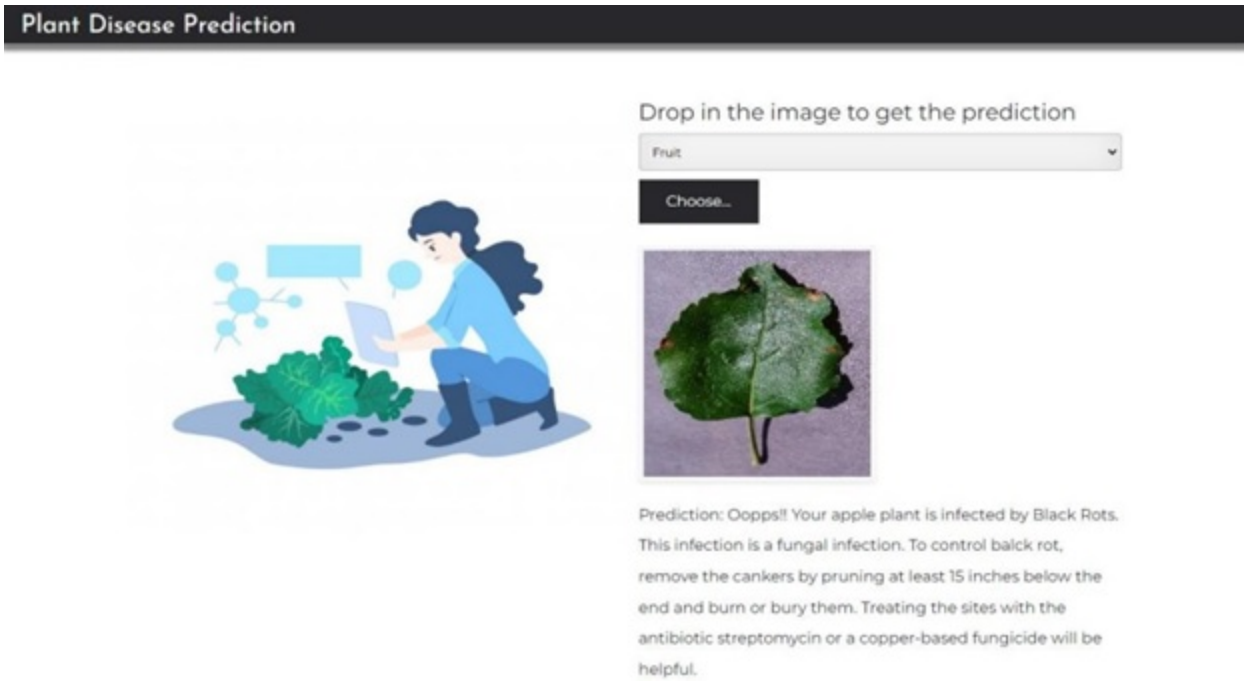
4. RESULT:

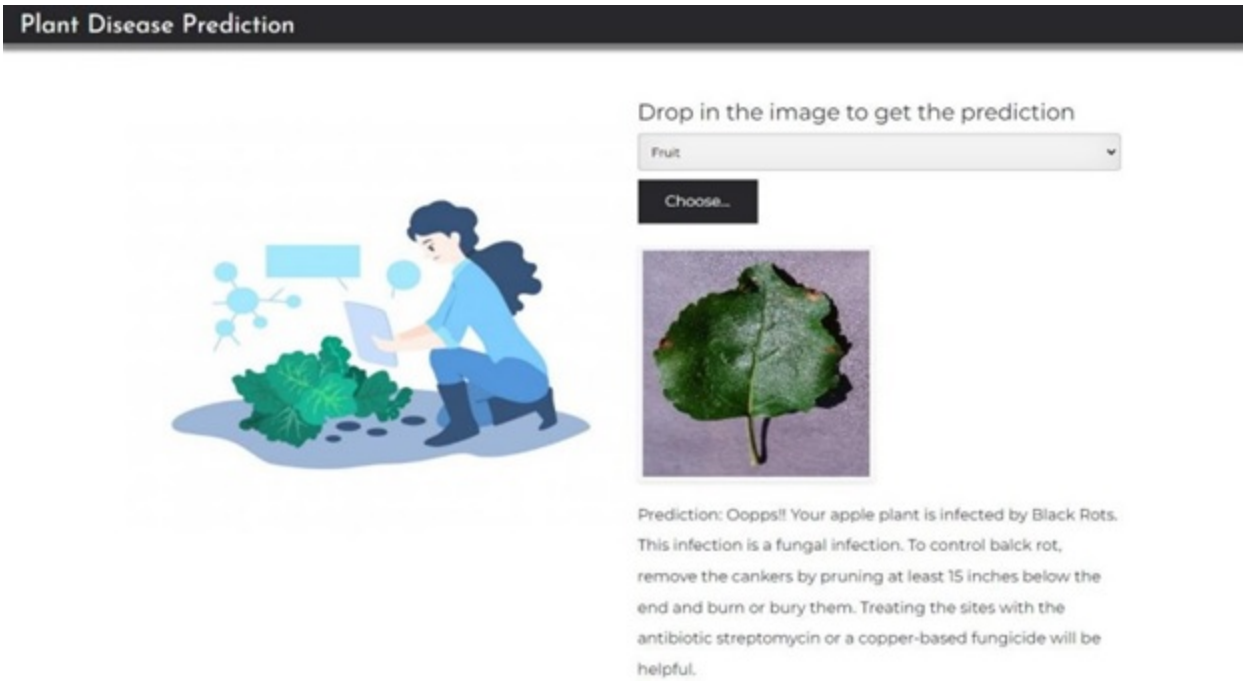
1. **`Home Page:`**

2. **Prediction Page:**



1. **Result Page:**





5. *ADVANTAGES & DISADVANTAGES:*

ADVANTAGES:

- The proposed model could predict the disease just from the image of a particular plant
- Easy to use UI
- Model has some good accuracy in detecting the plant just by taking theinput(leaf).

DISADVANTAGES:

- Prediction is limited to few plants as we havent trained all the plants .

6. *APPLICATIONS :*

This web application can be used by farmers or users to check whether their plant is infected or not and can also show the remedy so that the user can take necessary precautions.

These kind of web applications can be used in the agricultural sector as well as for small house hold plants as well.

7. **CONCLUSION :**

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 numberof 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. Usage of such applications could help the farmers to necessary precautions so that they dont face any loss as such.

8. FUTURE SCOPE :

As of now we have just build the web application which apparently takes the input as an image and then predict the out in the near future wecan develop an application which computer vision and AI techniques to predict the infection once you keep the camera near

the plant or leaf this could make our projecteven more usable.

9. BIBILOGRAPHY:

http://www.ijstr.org/final-print/nov2019/Fertilizers-Recommendation-SystemFor-Disease-Prediction-In-Tree-Leave.pdf

https://www.sciencedirect.com/science/article/pii/S0168169921004245

http://www.ijetajournal.org/volume-8/issue-2/IJETA-V8I2P1.pdf

https://www.semanticscholar.org/paper/Fertilizers-Recommendation-SystemFor-Disease-In-Neela-Nithya/495379d3ef2b461fabd2de8d0605c164cb1e396f

https://ieeexplore.ieee.org/document/8878781

https://www.irjet.net/archives/V7/i10/IRJET-V7I1004.pdf

APPENDIX:

A.Source Code:

Image Augmentation

In [1]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [2]:
```
train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

In [3]:
```
x_train = train_datagen.flow_from_directory('D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\frui
x_test = test_datagen.flow_from_directory('D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-
```

```
Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.
```

In [4]:
```
x_train.class_indices
```

Out[4]:
```
{'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

In [5]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

In [6]:
```
model = Sequential()
```

In [7]:
```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

In [8]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [9]:
```
model.add(Flatten())
```

Hidden Layers

In [10]:
```
model.add(Dense(128 ,activation='relu'))
model.add(Dense(64,activation = 'relu'))
```

Output Layer

In [11]:
```
model.add(Dense(6,activation = 'softmax'))
```

In [12]:
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

In [13]:
```
model.summary()
```

In [14]:
```
model.fit_generator(x_train,steps_per_epoch=5384//8,validation_data=x_test,validation_steps=1686//8,epochs=10)
```

```
C:\Users\VISHWANTH BAVIREDDY\AppData\Local\Temp\ipykernel_8872\1515256012.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=5384//8,validation_data=x_test,validation_steps=1686//8,epochs=10)
Epoch 1/10
673/673 [==============================] - 175s 258ms/step - loss: 0.4570 - accuracy: 0.8423 - val_loss: 0.2609 - val_accuracy: 0.9167
Epoch 2/10
673/673 [==============================] - 121s 179ms/step - loss: 0.2536 - accuracy: 0.9120 - val_loss: 0.2470 - val_accuracy: 0.9220
Epoch 3/10
673/673 [==============================] - 124s 184ms/step - loss: 0.2021 - accuracy: 0.9318 - val_loss: 0.1411 - val_accuracy: 0.9571
Epoch 4/10
673/673 [==============================] - 128s 190ms/step - loss: 0.1801 - accuracy: 0.9411 - val_loss: 0.1480 - val_accuracy: 0.9565
Epoch 5/10
673/673 [==============================] - 131s 195ms/step - loss: 0.1693 - accuracy: 0.9426 - val_loss: 0.1298 - val_accuracy: 0.9643
Epoch 6/10
673/673 [==============================] - 130s 193ms/step - loss: 0.1496 - accuracy: 0.9487 - val_loss: 0.1553 - val_accuracy: 0.9548
Epoch 7/10
673/673 [==============================] - 131s 195ms/step - loss: 0.1366 - accuracy: 0.9547 - val_loss: 0.1290 - val_accuracy: 0.9643
Epoch 8/10
673/673 [==============================] - 131s 194ms/step - loss: 0.1050 - accuracy: 0.9627 - val_loss: 0.1366 - val_accuracy: 0.9565
Epoch 9/10
673/673 [==============================] - 130s 192ms/step - loss: 0.1193 - accuracy: 0.9582 - val_loss: 0.1769 - val_accuracy: 0.9506
Epoch 10/10
673/673 [==============================] - 137s 204ms/step - loss: 0.0994 - accuracy: 0.9658 - val_loss: 0.1493 - val_accuracy: 0.9607
```

Out[14]:
```
<keras.callbacks.History at 0x1a82019afd0>
```

Saving The Model

In [15]:
```
model.save("fruit.h5")
```

Test the model

In [35]:
```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

In [36]:
```
#Load the model

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

In [37]:
```
img = image.load_img("D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Apple_
```

In [38]:
```
img
```

Out[38]:

In [39]:
```
img = image.load_img("D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Apple_
img
```

Out[39]:

In [40]:
```
x=image.img_to_array(img)
```

In [41]:
```
x
```

Out[41]:
```
array([[[165., 153., 189.],
        [165., 153., 189.],
        [165., 153., 189.],
        ...,
        [176., 170., 206.],
        [176., 170., 206.],
        [176., 170., 206.]],

       [[164., 152., 188.],
        [164., 152., 188.],
        [164., 152., 188.],
        ...,
        [173., 167., 203.],
        [172., 166., 202.],
        [172., 166., 202.]],

       [[163., 151., 187.],
        [163., 151., 187.],
        [163., 151., 187.],
        ...,
        [172., 166., 202.],
        [170., 164., 200.],
        [169., 163., 199.]],

       ...,

       [[135., 119., 156.],
        [139., 123., 160.],
        [134., 118., 155.],
        ...,
        [143., 133., 168.],
        [138., 128., 163.],
        [141., 131., 166.]],

       [[136., 120., 157.],
        [134., 118., 155.],
        [134., 118., 155.],
        ...,
        [141., 131., 166.],
        [141., 131., 166.],
        [146., 136., 171.]],

       [[135., 119., 156.],
        [140., 124., 161.],
        [143., 127., 164.],
        ...,
        [145., 135., 170.],
        [151., 141., 176.],
        [140., 130., 165.]]], dtype=float32)
```

In [42]:
```
x=np.expand_dims(x,axis=0)
```

In [43]:
```
x
```

Out[43]:
```
array([[[[165., 153., 189.],
         [165., 153., 189.],
         [165., 153., 189.],
         ...,
         [176., 170., 206.],
         [176., 170., 206.],
         [176., 170., 206.]],

        [[164., 152., 188.],
         [164., 152., 188.],
         [164., 152., 188.],
         ...,
         [173., 167., 203.],
         [172., 166., 202.],
         [172., 166., 202.]],

        [[163., 151., 187.],
         [163., 151., 187.],
         [163., 151., 187.],
         ...,
         [172., 166., 202.],
         [170., 164., 200.],
         [169., 163., 199.]],

        ...,

        [[135., 119., 156.],
         [139., 123., 160.],
         [134., 118., 155.],
         ...,
         [143., 133., 168.],
         [138., 128., 163.],
         [141., 131., 166.]],

        [[136., 120., 157.],
         [134., 118., 155.],
         [134., 118., 155.],
         ...,
         [141., 131., 166.],
         [141., 131., 166.],
         [146., 136., 171.]],

        [[135., 119., 156.],
         [140., 124., 161.],
         [143., 127., 164.],
         ...,
         [145., 135., 170.],
         [151., 141., 176.],
         [140., 130., 165.]]], dtype=float32)
```

In [46]:
```
pred = np.argmax(model.predict(x),axis=1)
```

In [47]:
```
pred
```

Out[47]:
```
array([1], dtype=int64)
```

In [48]:
```
x_test.class_indices
```

Out[48]:
```
{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}
```

In [49]:
```
index=['Apple___Black_rot',
 'Apple___healthy',
 'Corn_(maize)___Northern_Leaf_Blight',
 'Corn_(maize)___healthy',
 'Peach___Bacterial_spot',
 'Peach___healthy']
```

In [50]:
```
index[pred[0]]
```

Out[50]:
```
'Apple___healthy'
```

In [65]:
```
img = image.load_img("D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\test\\Apple_
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = 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[pred[0]]
```

Out[65]:
```
'Apple___healthy'
```

In [63]:
```
img = image.load_img("D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\train\\Apple
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = 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[pred[0]]
```

Out[63]:
```
'Apple___healthy'
```

In [60]:
```
img = image.load_img("D:\\Project Buildathon\\Fertilizers_Recommendation_ System_For_Disease_ Prediction\\Dataset Plant Disease\\fruit-dataset\\fruit-dataset\\train\\Peach
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = 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[pred[0]]
```

Out[60]:
```
'Peach___Bacterial_spot'
```