```
09/30/2022  07:10 PM    <DIR>          .ssh
11/03/2022  12:57 AM    <DIR>          anaconda3
05/14/2022  09:14 PM    <DIR>          Contacts
08/14/2022  10:41 PM    <DIR>          Documents
09/21/2022  02:05 PM    <DIR>          Dropbox
09/18/2022  10:51 PM    <DIR>          Favorites
11/03/2022  01:16 AM       339,185,106 Fertilizers_Recommendation_ System_For_Disease_ Pr
ediction (2).zip
11/03/2022  03:11 AM         9,184,528 fruit.h5
11/03/2022  01:29 AM             2,951 ImagePreProcessing for Fruit and veg dataset.ipynb
09/18/2022  10:51 PM    <DIR>          Links
08/12/2022  09:36 PM    <DIR>          Music
09/20/2022  09:20 PM    <DIR>          OneDrive
05/14/2022  09:14 PM    <DIR>          Saved Games
09/24/2022  08:15 PM    <DIR>          Searches
11/03/2022  01:28 AM             2,951 Untitled.ipynb
11/03/2022  03:11 AM            17,369 Untitled1.ipynb
08/12/2022  09:37 PM    <DIR>          Videos
               8 File(s)    348,399,466 bytes
              20 Dir(s)  205,759,488,000 bytes free
```

Test the model

In [43]:

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

In [44]:

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

In [45]:

```python
img=image.load_img(r"E:\IBM\Fertilizers_Recommendation_ System_For_Disease_ Prediction\Da
taset Plant Disease\fruit-dataset\fruit-dataset\test\Apple___healthy\0adc1c5b-8958-47c0-a
152-f28078c214f1___RS_HL 7825.JPG",target_size=(128,128))
img
```

Out[45]:

In [46]:

```python
img
```

Out[46]:

In [47]:

```python
x=image.img_to_array(img)
```

In [48]:

```
x
```

```
array([[[ 99.,  86., 106.],
        [101.,  88., 108.],
        [118., 105., 125.],
        ...,
        [ 92.,  83., 102.],
        [ 93.,  84., 103.],
        [ 89.,  80.,  99.]],

       [[ 96.,  83., 103.],
        [ 87.,  74.,  94.],
        [102.,  89., 109.],
        ...,
        [ 88.,  79.,  98.],
        [ 89.,  80.,  99.],
        [ 83.,  74.,  93.]],

       [[ 86.,  73.,  93.],
        [ 88.,  75.,  95.],
        [ 98.,  85., 105.],
        ...,
        [107.,  98., 117.],
        [ 96.,  87., 106.],
        [ 96.,  87., 106.]],

       ...,

       [[172., 175., 194.],
        [173., 176., 195.],
        [175., 178., 197.],
        ...,
        [179., 180., 198.],
        [184., 185., 203.],
        [179., 180., 198.]],

       [[172., 175., 194.],
        [170., 173., 192.],
        [173., 176., 195.],
        ...,
        [178., 179., 197.],
        [182., 183., 201.],
        [178., 179., 197.]],

       [[169., 172., 191.],
        [166., 169., 188.],
        [168., 171., 190.],
        ...,
        [187., 188., 206.],
        [185., 186., 204.],
        [186., 187., 205.]]], dtype=float32)
```

In [49]:

```
x=np.expand_dims(x,axis=0)
```

In [50]:

```
x
```

```
array([[[[ 99.,  86., 106.],
         [101.,  88., 108.],
         [118., 105., 125.],
         ...,
         [ 92.,  83., 102.],
         [ 93.,  84., 103.],
         [ 89.,  80.,  99.]],
```

```
       [[ 96.,   83.,  103.],
        [ 87.,   74.,   94.],
        [102.,   89.,  109.],
        ...,
        [ 88.,   79.,   98.],
        [ 89.,   80.,   99.],
        [ 83.,   74.,   93.]],

       [[ 86.,   73.,   93.],
        [ 88.,   75.,   95.],
        [ 98.,   85.,  105.],
        ...,
        [107.,   98.,  117.],
        [ 96.,   87.,  106.],
        [ 96.,   87.,  106.]],

       ...,

       [[172.,  175.,  194.],
        [173.,  176.,  195.],
        [175.,  178.,  197.],
        ...,
        [179.,  180.,  198.],
        [184.,  185.,  203.],
        [179.,  180.,  198.]],

       [[172.,  175.,  194.],
        [170.,  173.,  192.],
        [173.,  176.,  195.],
        ...,
        [178.,  179.,  197.],
        [182.,  183.,  201.],
        [178.,  179.,  197.]],

       [[169.,  172.,  191.],
        [166.,  169.,  188.],
        [168.,  171.,  190.],
        ...,
        [187.,  188.,  206.],
        [185.,  186.,  204.],
        [186.,  187.,  205.]]]], dtype=float32)
```

In [51]:

```
y=np.argmax(model.predict(x),axis=1)
```

```
1/1 [==============================] - 0s 244ms/step
```

In [52]:

```
x_train.class_indices
```

Out[52]:

```
{'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 [53]:

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

In [60]:

```
img=image.load_img(r"E:\IBM\Fertilizers_Recommendation_ System_For_Disease_ Prediction\Da
taset Plant Disease\fruit-dataset\fruit-dataset\test\Peach___healthy\0a2ed402-5d23-4e8d-b
c98-b264aea9c3fb___Rutg._HL 2471.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]]
```

1/1 [==============================] - 0s 33ms/step

Out[60]:

'Corn_(maize)___healthy'

In [ ]:

```
# Predicting a Second Model Just For The Example
```

In [54]:

```
index[y[0]]
```

Out[54]:

'Apple___healthy'