

In [1]:

```
ls
```

Volume in drive C is Local disk :
Volume Serial Number is EE22-D61B

Directory of C:\Users\LonelyDinesh

```
11/03/2022  01:47 AM    <DIR>          .
07/30/2022  09:28 AM    <DIR>          ..
10/25/2022  01:24 PM                6,329 .bash_history
10/25/2022  12:29 AM                212 .gitconfig
11/03/2022  01:39 AM    <DIR>          .ipynb_checkpoints
11/03/2022  01:16 AM    <DIR>          .ipython
11/03/2022  01:12 AM    <DIR>          .keras
09/22/2022  11:10 PM                20 .lessht
11/03/2022  01:06 AM    <DIR>          .matplotlib
11/02/2022  09:06 PM    <DIR>          .node-red
11/03/2022  01:02 AM    <DIR>          .spyder-py3
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  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  01:47 AM      697 Untitled1.ipynb
08/12/2022  09:37 PM    <DIR>          Videos
       7 File(s)      339,198,266 bytes
      20 Dir(s)   204,871,598,080 bytes free
```

In [2]:

```
pwd
```

Out[2]:

```
'C:\\Users\\LonelyDinesh'
```

Image Augmentation

In [3]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [5]:

```
train_datagen = ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255,)
```

In [6]:

```
x_train = train_datagen.flow_from_directory(r'E:\IBM\Fertilizers_Recommendation_System_F
or_Disease_Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\train', target_si
ze = (128,128), batch_size = 32, class_mode = 'categorical')
```

Found 5384 images belonging to 6 classes.

In [7]:

```
x_test = test_datagen.flow_from_directory(r'E:\IBM\Fertilizers_Recommendation_System_For_Disease_Prediction\Dataset_Plant_Disease\fruit-dataset\fruit-dataset\test',target_size=(128,128),batch_size = 32,class_mode = 'categorical')
```

Found 1686 images belonging to 6 classes.

In [8]:

```
x_train.class_indices
```

Out[8]:

```
{'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 [9]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

In [10]:

```
model=Sequential()
```

In [11]:

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

In [12]:

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [13]:

```
model.add(Flatten())
```

In [14]:

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
flatten (Flatten)	(None, 127008)	0
=====		
Total params: 896		
Trainable params: 896		
Non-trainable params: 0		
=====		

In [15]:

```
32*(3*3*3+1)
```

Out[15]:

Hidden Layers

In [16]:

```
model.add(Dense(6, activation='softmax'))
```

In [17]:

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

In [18]:

```
len(x_train)
```

Out[18]:

169

In [19]:

```
1238/24
```

Out[19]:

51.583333333333336

In [20]:

```
model.fit_generator(x_train, steps_per_epoch=len(x_train), validation_data=x_test, validation_steps=len(x_test), epochs=10)
```

C:\Users\LonelyDinesh\AppData\Local\Temp\ipykernel_13816\1645096295.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=len(x_train), validation_data=x_test, validation_steps=len(x_test), epochs=10)
```

Epoch 1/10

169/169 [=====] - 105s 612ms/step - loss: 1.1001 - accuracy: 0.7903 - val_loss: 0.3263 - val_accuracy: 0.8968

Epoch 2/10

169/169 [=====] - 52s 306ms/step - loss: 0.2627 - accuracy: 0.9118 - val_loss: 0.3388 - val_accuracy: 0.8962

Epoch 3/10

169/169 [=====] - 51s 299ms/step - loss: 0.2164 - accuracy: 0.9303 - val_loss: 0.2605 - val_accuracy: 0.9045

Epoch 4/10

169/169 [=====] - 52s 306ms/step - loss: 0.2177 - accuracy: 0.9268 - val_loss: 0.2080 - val_accuracy: 0.9241

Epoch 5/10

169/169 [=====] - 51s 304ms/step - loss: 0.1832 - accuracy: 0.9398 - val_loss: 0.1579 - val_accuracy: 0.9425

Epoch 6/10

169/169 [=====] - 52s 307ms/step - loss: 0.1619 - accuracy: 0.9447 - val_loss: 0.2502 - val_accuracy: 0.9217

Epoch 7/10

169/169 [=====] - 52s 306ms/step - loss: 0.1424 - accuracy: 0.9528 - val_loss: 0.1692 - val_accuracy: 0.9466

Epoch 8/10

169/169 [=====] - 51s 302ms/step - loss: 0.1467 - accuracy: 0.9467 - val_loss: 0.1125 - val_accuracy: 0.9597

Epoch 9/10

169/169 [=====] - 52s 307ms/step - loss: 0.1301 - accuracy: 0.9549 - val_loss: 0.1007 - val_accuracy: 0.9662

Epoch 10/10

169/169 [=====] - 52s 308ms/step - loss: 0.1170 - accuracy: 0.9586 - val_loss: 0.1241 - val_accuracy: 0.9561

Out [20]:

<keras.callbacks.History at 0x1eb808a0430>

Training Model

In [23]:

```
ls

Volume in drive C is Local disk :
Volume Serial Number is EE22-D61B

Directory of C:\Users\LonelyDinesh

11/03/2022  02:55 AM      <DIR>          .
07/30/2022  09:28 AM      <DIR>          ..
10/25/2022  01:24 PM                6,329 .bash_history
10/25/2022  12:29 AM                212 .gitconfig
11/03/2022  01:39 AM      <DIR>          .ipynb_checkpoints
11/03/2022  01:16 AM      <DIR>          .ipython
11/03/2022  01:12 AM      <DIR>          .keras
09/22/2022  11:10 PM                20 .lessht
11/03/2022  01:06 AM      <DIR>          .matplotlib
11/02/2022  09:06 PM      <DIR>          .node-red
11/03/2022  01:02 AM      <DIR>          .spyder-py3
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  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  02:55 AM            12,835 Untitled1.ipynb
08/12/2022  09:37 PM      <DIR>          Videos
      7 File(s)      339,210,404 bytes
     20 Dir(s)   206,038,216,704 bytes free
```

In [41]:

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

In [42]:

```
ls

Volume in drive C is Local disk :
Volume Serial Number is EE22-D61B

Directory of C:\Users\LonelyDinesh

11/03/2022  03:11 AM      <DIR>          .
07/30/2022  09:28 AM      <DIR>          ..
10/25/2022  01:24 PM                6,329 .bash_history
10/25/2022  12:29 AM                212 .gitconfig
11/03/2022  01:39 AM      <DIR>          .ipynb_checkpoints
11/03/2022  01:16 AM      <DIR>          .ipython
11/03/2022  01:12 AM      <DIR>          .keras
09/22/2022  11:10 PM                20 .lessht
11/03/2022  01:06 AM      <DIR>          .matplotlib
11/02/2022  09:06 PM      <DIR>          .node-red
11/03/2022  01:02 AM      <DIR>          .spyder-py3
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  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  02:55 AM            12,835 Untitled1.ipynb
08/12/2022  09:37 PM      <DIR>          Videos
      7 File(s)      339,210,404 bytes
     20 Dir(s)   206,038,216,704 bytes free
```

```

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]:

```

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

```

In [44]:

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

In [45]:

```

img=image.load_img(r"E:\IBM\Fertilizers_Recommendation_System_For_Disease_Prediction\Da
taset Plant Disease\fruit-dataset\fruit-dataset\test\Apple___healthy\0adclc5b-8958-47c0-a
152-f28078c214f1___RS_HL_7825.JPG",target_size=(128,128))
img

```

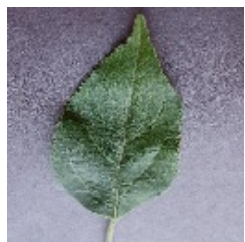
Out[45]:



In [46]:

```
img
```

Out[46]:



In [47]:

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

In [48]:

```
x
```

Out[48]:

```
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
```

Out[50]:

```
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\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Peach__healthy\0a2ed402-5d23-4e8d-bc98-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'