Import the libraries

In [15]:

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

**from** tensorflow.keras.models **import** Sequential

**from** tensorflow.keras.layers **import** Dense,Convolution2D,MaxPooling2D,Flatten

**import** numpy **as** np

**from** tensorflow.keras. importload\_model

**from** tensorflow.keras.preprocessing **import** image

Image Augmentation

In [3]:

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

train\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255,zoom\_range**=**0.2,horizontal\_flip**=True**,vertical\_flip**=False**)

test\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255)

In [4]:

x\_train**=**train\_datagen**.**flow\_from\_directory("/content/Dataset Plant Disease/fruit-dataset/fruit dataset/train",target\_size**=**(128,128),class\_mode**=**'categorical',batch\_size**=**24)

Found 5384 images belonging to 6 classes.

In [5]:

x\_test**=**test\_datagen**.**flow\_from\_directory(r"/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test",target\_size**=**(128,128),

class\_mode**=**'categorical',batch\_size**=**24)

Found 1686 images belonging to 6 classes.

Create the model

In [6]:

**From** tensorflow.keras.models **import** Sequential

**From** tensorflow.keras.layers **import** Dense,Convolution2D,MaxPooling2D,Flatten

Add Layers

In [7]:

model**=**Sequential()

In [9]:

model**.**add(Convolution2D(32,(3,3),input\_shape**=**(128,128,3),activation**=**'relu'))

In [10]:

model**.**add(MaxPooling2D(pool\_size**=**(2,2)))

model**.**add(Flatten())

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 [11]:

32**\***(3**\***3**\***3**+**1)

model**.**add(Dense(300,activation**=**'relu'))

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

In [12]:

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

model**.**compile(loss**=**'categorical\_crossentropy',optimizer**=**'adam',metrics**=**['accuracy'])

len(x\_train)

Out[12]:

225

In [13]:

1238**/**24

Out[13]:

51.583333333333336

Fit the model

In [14]:

model**.**fit(x\_train,steps\_per\_epoch**=**len(x\_train),validation\_data**=**x\_test,validation\_steps**=**len(x\_test),epochs**=**10)

Epoch 1/10

225/225 [==============================] - 2639s 12s/step - loss: 1.3354 - accuracy: 0.7652 - val\_loss: 0.4907 - val\_accuracy: 0.8215

Epoch 2/10

225/225 [==============================] - 172s 762ms/step - loss: 0.2829 - accuracy: 0.9008 - val\_loss: 0.1736 - val\_accuracy: 0.9383

Epoch 3/10

225/225 [==============================] - 179s 794ms/step - loss: 0.2056 - accuracy: 0.9296 - val\_loss: 0.1954 - val\_accuracy: 0.9312

Epoch 4/10

225/225 [==============================] - 172s 765ms/step - loss: 0.1694 - accuracy: 0.9383 - val\_loss: 0.2187 - val\_accuracy: 0.9253

Epoch 5/10

225/225 [==============================] - 179s 796ms/step - loss: 0.1539 - accuracy: 0.9461 - val\_loss: 0.1366 - val\_accuracy: 0.9543

Epoch 6/10

225/225 [==============================] - 172s 765ms/step - loss: 0.1428 - accuracy: 0.9491 - val\_loss: 0.1668 - val\_accuracy: 0.9442

Epoch 7/10

225/225 [==============================] - 175s 774ms/step - loss: 0.1333 - accuracy: 0.9538 - val\_loss: 0.1976 - val\_accuracy: 0.9253

Epoch 8/10

225/225 [==============================] - 174s 774ms/step - loss: 0.1172 - accuracy: 0.9590 - val\_loss: 0.0944 - val\_accuracy: 0.9674

Epoch 9/10

225/225 [==============================] - 172s 763ms/step - loss: 0.1143 - accuracy: 0.9569 - val\_loss: 0.1306 - val\_accuracy: 0.9561

Epoch 10/10

225/225 [==============================] - 179s 795ms/step - loss: 0.0913 - accuracy: 0.9673 - val\_loss: 0.1848 - val\_accuracy: 0.9460

Out[14]:

Save the model

In [16]:

model**.**save('fruitdata.h5')

Testing the model

In [17]:

model**=**load\_model('fruitdata.h5')

In [18]:

**Import** numpy **as** np

**From** tensorflow.keras.models **import** load\_model

**From** tensorflow.keras.preprocessing **import** image

In [19]:

img**=**image**.**load\_img( r"/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple\_\_\_healthy/01efa999-757d-487e-8250-27c7854c0ca8\_\_\_RS\_HL 7515.JPG",target\_size**=**(128,128))

In [21]:

img

Out[21]:



In [22]:

img**=**image**.**load\_img(r"/content/drive/MyDrive/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple\_\_\_healthy/01efa999-757d-487e-8250-27c7854c0ca8\_\_\_RS\_HL 7515.JPG",target\_size**=**(128,128))

img

Out[22]:



In [23]:

x**=**image**.**img\_to\_array(img)

x

Out[23]:

array([[[111., 122., 152.],

[110., 121., 151.],

[118., 129., 159.],

...,

[184., 198., 225.],

[187., 201., 228.],

[180., 194., 221.]],

[[124., 135., 165.],

[117., 128., 158.],

[121., 132., 162.],

...,

[178., 192., 219.],

[191., 205., 232.],

[179., 193., 220.]],

[[123., 134., 164.],

[117., 128., 158.],

[114., 125., 155.],

...,

[180., 194., 221.],

[189., 203., 230.],

[179., 193., 220.]],

...,

[[114., 128., 157.],

[118., 132., 161.],

[125., 139., 168.],

...,

[177., 191., 217.],

[187., 199., 223.],

[180., 192., 216.]],

[[120., 134., 163.],

[125., 139., 168.],

[122., 136., 165.],

...,

[189., 197., 216.],

[188., 197., 214.],

[186., 195., 210.]],

[[127., 141., 170.],

[118., 132., 161.],

[118., 132., 161.],

...,

[160., 167., 183.],

[172., 180., 191.],

[190., 199., 208.]]], dtype=float32)

In [24]:

x**=**np**.**expand\_dims(x,axis**=**0)

x

Out[24]:

array([[[[111., 122., 152.],

[110., 121., 151.],

[118., 129., 159.],

...,

[184., 198., 225.],

[187., 201., 228.],

[180., 194., 221.]],

[[124., 135., 165.],

[117., 128., 158.],

[121., 132., 162.],

...,

[178., 192., 219.],

[191., 205., 232.],

[179., 193., 220.]],

[[123., 134., 164.],

[117., 128., 158.],

[114., 125., 155.],

...,

[180., 194., 221.],

[189., 203., 230.],

[179., 193., 220.]],

...,

[[114., 128., 157.],

[118., 132., 161.],

[125., 139., 168.],

...,

[177., 191., 217.],

[187., 199., 223.],

[180., 192., 216.]],

[[120., 134., 163.],

[125., 139., 168.],

[122., 136., 165.],

...,

[189., 197., 216.],

[188., 197., 214.],

[186., 195., 210.]],

[[127., 141., 170.],

[118., 132., 161.],

[118., 132., 161.],

...,

[160., 167., 183.],

[172., 180., 191.],

[190., 199., 208.]]]], dtype=float32)

In [25]:

x

Out[25]:

array([[[[111., 122., 152.],

[110., 121., 151.],

[118., 129., 159.],

...,

[184., 198., 225.],

[187., 201., 228.],

[180., 194., 221.]],

[[124., 135., 165.],

[117., 128., 158.],

[121., 132., 162.],

...,

[178., 192., 219.],

[191., 205., 232.],

[179., 193., 220.]],

[[123., 134., 164.],

[117., 128., 158.],

[114., 125., 155.],

...,

[180., 194., 221.],

[189., 203., 230.],

[179., 193., 220.]],

...,

[[114., 128., 157.],

[118., 132., 161.],

[125., 139., 168.],

...,

[177., 191., 217.],

[187., 199., 223.],

[180., 192., 216.]],

[[120., 134., 163.],

[125., 139., 168.],

[122., 136., 165.],

...,

[189., 197., 216.],

[188., 197., 214.],

[186., 195., 210.]],

[[127., 141., 170.],

[118., 132., 161.],

[118., 132., 161.],

...,

[160., 167., 183.],

[172., 180., 191.],

[190., 199., 208.]]]], dtype=float32)

In [26]:

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

1/1 [==============================] - 0s 201ms/step

In [27]:

index**=**['Apple\_\_\_Black\_rot','Apple\_\_\_healthy','Corn\_(maize)\_\_\_Northern\_Leaf\_Blight','Corn\_(maize)\_\_\_healthy','Peach\_\_\_Bacterial\_spot','Peach\_\_\_healthy']

index[y[0]]

Out[27]:

'Apple\_\_\_healthy'

In [28]:

img**=**image**.**load\_img(r r"/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple\_\_\_healthy/01efa999-757d-487e-8250-27c7854c0ca8\_\_\_RS\_HL 7515.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 48ms/step

Out[28]:

'Apple\_\_\_healthy'