SPRINT-4

STEP-1

Looking at the past prediction stored in the cloud.

Priority is low.

STEP-2

Looking at the user feedback.

Priority is low.

STEP-3

Tweaking the ML Model.

Priority is low.

STEP-4

Make the changes required for the definition of the done.

cd D:\Nalaiya Thiran\Dataset Plant Disease

D:\Nalaiya Thiran\Dataset Plant Disease

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)

x\_train **=** train\_datagen**.**flow\_from\_directory(r'D:\Nalaiya Thiran\Dataset Plant Disease\fruit-dataset\fruit-dataset\train',target\_size**=**(64,64),class\_mode**=**'categorical',batch\_size**=**24)

Found 5384 images belonging to 6 classes.

In [4]:

x\_test **=** train\_datagen**.**flow\_from\_directory(r'D:\Nalaiya Thiran\Dataset Plant Disease\fruit-dataset\fruit-dataset\test',target\_size**=**(64,64),class\_mode**=**'categorical',batch\_size**=**24)

Found 1686 images belonging to 6 classes.

In [5]:

x\_train**.**class\_indices

Out[5]:

{'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 [6]:

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

**from** tensorflow.keras.layers **import** Dense

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

In [7]:

model**=**Sequential()

model**.**add(Convolution2D(32,(3,3),input\_shape**=**(64,64,3),activation**=**'relu'))

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

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

model**.**summary()

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 62, 62, 32) 896

max\_pooling2d (MaxPooling2D (None, 31, 31, 32) 0

)

flatten (Flatten) (None, 30752) 0

=================================================================

Total params: 896

Trainable params: 896

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [8]:

model**.**add(Dense(800,activation**=**'relu'))

model**.**add(Dense(650,activation**=**'relu'))

In [9]:

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

len(x\_train)

Out[9]:

225

In [13]:

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

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 [==============================] - 48s 211ms/step - loss: 0.7575 - accuracy: 0.7834 - val\_loss: 0.2438 - val\_accuracy: 0.9199

Epoch 2/10

225/225 [==============================] - 39s 171ms/step - loss: 0.2540 - accuracy: 0.9120 - val\_loss: 0.3236 - val\_accuracy: 0.8826

Epoch 3/10

225/225 [==============================] - 39s 175ms/step - loss: 0.2118 - accuracy: 0.9248 - val\_loss: 0.1707 - val\_accuracy: 0.9437

Epoch 4/10

225/225 [==============================] - 39s 173ms/step - loss: 0.1760 - accuracy: 0.9389 - val\_loss: 0.1151 - val\_accuracy: 0.9591

Epoch 5/10

225/225 [==============================] - 39s 173ms/step - loss: 0.1273 - accuracy: 0.9569 - val\_loss: 0.1087 - val\_accuracy: 0.9620

Epoch 6/10

225/225 [==============================] - 39s 174ms/step - loss: 0.1197 - accuracy: 0.9577 - val\_loss: 0.1114 - val\_accuracy: 0.9644

Epoch 7/10

225/225 [==============================] - 39s 172ms/step - loss: 0.1029 - accuracy: 0.9649 - val\_loss: 0.1210 - val\_accuracy: 0.9620

Epoch 8/10

225/225 [==============================] - 39s 173ms/step - loss: 0.1202 - accuracy: 0.9569 - val\_loss: 0.1499 - val\_accuracy: 0.9520

Epoch 9/10

225/225 [==============================] - 40s 177ms/step - loss: 0.0819 - accuracy: 0.9729 - val\_loss: 0.1327 - val\_accuracy: 0.9555

Epoch 10/10

225/225 [==============================] - 40s 176ms/step - loss: 0.0755 - accuracy: 0.9742 - val\_loss: 0.0949 - val\_accuracy: 0.9680

Out[13]:

In [14]:

model**.**save('fruit\_dataset.h5')

In [15]:

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

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

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

model**=**load\_model('fruit\_dataset.h5')

In [22]:

img**=**image**.**load\_img(r'D:\Nalaiya Thiran\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Peach\_\_\_Bacterial\_spot\00f87dfc-bf6c-481e-9022-bfb98224d4b7\_\_\_Rutg.\_Bact.S 1965.JPG')

In [23]:

img

Out[23]:



In [29]:

img**=**image**.**load\_img(r'D:\Nalaiya Thiran\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Peach\_\_\_Bacterial\_spot\00f87dfc-bf6c-481e-9022-bfb98224d4b7\_\_\_Rutg.\_Bact.S 1965.JPG',target\_size**=**(64,64))

img

Out[29]:



In [30]:

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

x

Out[30]:

array([[[118., 114., 115.],

[118., 114., 115.],

[118., 114., 115.],

...,

[100., 94., 96.],

[ 97., 91., 93.],

[ 92., 86., 88.]],

[[128., 124., 125.],

[126., 122., 123.],

[117., 113., 114.],

...,

[102., 96., 98.],

[ 96., 90., 92.],

[103., 97., 99.]],

[[133., 129., 130.],

[125., 121., 122.],

[127., 123., 124.],

...,

[101., 95., 97.],

[ 92., 86., 88.],

[ 97., 91., 93.]],

...,

[[147., 143., 144.],

[146., 142., 143.],

[145., 141., 142.],

...,

[124., 120., 121.],

[122., 118., 119.],

[122., 118., 119.]],

[[147., 143., 144.],

[145., 141., 142.],

[141., 137., 138.],

...,

[125., 121., 122.],

[124., 120., 121.],

[118., 114., 115.]],

[[146., 142., 143.],

[147., 143., 144.],

[149., 145., 146.],

...,

[124., 120., 121.],

[120., 116., 117.],

[121., 117., 118.]]], dtype=float32)

In [31]:

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

x

Out[31]:

array([[[[118., 114., 115.],

[118., 114., 115.],

[118., 114., 115.],

...,

[100., 94., 96.],

[ 97., 91., 93.],

[ 92., 86., 88.]],

[[128., 124., 125.],

[126., 122., 123.],

[117., 113., 114.],

...,

[102., 96., 98.],

[ 96., 90., 92.],

[103., 97., 99.]],

[[133., 129., 130.],

[125., 121., 122.],

[127., 123., 124.],

...,

[101., 95., 97.],

[ 92., 86., 88.],

[ 97., 91., 93.]],

...,

[[147., 143., 144.],

[146., 142., 143.],

[145., 141., 142.],

...,

[124., 120., 121.],

[122., 118., 119.],

[122., 118., 119.]],

[[147., 143., 144.],

[145., 141., 142.],

[141., 137., 138.],

...,

[125., 121., 122.],

[124., 120., 121.],

[118., 114., 115.]],

[[146., 142., 143.],

[147., 143., 144.],

[149., 145., 146.],

...,

[124., 120., 121.],

[120., 116., 117.],

[121., 117., 118.]]]], dtype=float32)

In [32]:

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

y

1/1 [==============================] - 2s 2s/step

Out[32]:

array([1], dtype=int64)

In [33]:

x\_train**.**class\_indices

Out[33]:

{'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 [34]:

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

In [35]:

index[y[0]]

Out[35]:

'Apple\_\_\_healthy'