import the libraries

In [18]:

**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.models **import** load\_model

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

image augmentation

In [1]:

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

In [2]:

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

In [3]:

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

In [4]:

x\_train**=**train\_datagen**.**flow\_from\_directory(r"/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/Veg-dataset/train\_set",target\_size**=**(128,128),

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

Found 11386 images belonging to 9 classes.

In [5]:

x\_test**=**test\_datagen**.**flow\_from\_directory(r'/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/Veg-dataset/test\_set',target\_size**=**(128,128),class\_mode**=**'categorical',batch\_size**=**24)

Found 3416 images belonging to 9 classes.

add layers

In [6]:

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

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

In [7]:

model**=**Sequential()

In [8]:

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

In [9]:

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

In [10]:

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

In [11]:

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 [12]:

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

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

In [13]:

model**.**add(Dense(9,activation**=**'softmax'))

In [14]:

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

In [15]:

len(x\_train)

Out[15]:

475

In [16]:

1238**/**24

Out[16]:

51.583333333333336

fit the model

In [17]:

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

Epoch 1/10

475/475 [==============================] - 3275s 7s/step - loss: 1.4156 - accuracy: 0.6092 - val\_loss: 0.8031 - val\_accuracy: 0.7178

Epoch 2/10

475/475 [==============================] - 375s 789ms/step - loss: 0.5598 - accuracy: 0.8004 - val\_loss: 0.5123 - val\_accuracy: 0.8150

Epoch 3/10

475/475 [==============================] - 362s 762ms/step - loss: 0.4634 - accuracy: 0.8402 - val\_loss: 0.3696 - val\_accuracy: 0.8718

Epoch 4/10

475/475 [==============================] - 364s 765ms/step - loss: 0.3832 - accuracy: 0.8669 - val\_loss: 0.2470 - val\_accuracy: 0.9183

Epoch 5/10

475/475 [==============================] - 364s 765ms/step - loss: 0.3210 - accuracy: 0.8873 - val\_loss: 0.3565 - val\_accuracy: 0.8741

Epoch 6/10

475/475 [==============================] - 360s 758ms/step - loss: 0.2831 - accuracy: 0.9040 - val\_loss: 0.4657 - val\_accuracy: 0.8352

Epoch 7/10

475/475 [==============================] - 371s 781ms/step - loss: 0.2453 - accuracy: 0.9135 - val\_loss: 0.1700 - val\_accuracy: 0.9458

Epoch 8/10

475/475 [==============================] - 361s 760ms/step - loss: 0.2693 - accuracy: 0.9062 - val\_loss: 0.3006 - val\_accuracy: 0.8967

Epoch 9/10

475/475 [==============================] - 373s 786ms/step - loss: 0.2177 - accuracy: 0.9257 - val\_loss: 0.1678 - val\_accuracy: 0.9429

Epoch 10/10

475/475 [==============================] - 373s 786ms/step - loss: 0.2082 - accuracy: 0.9288 - val\_loss: 0.1483 - val\_accuracy: 0.9494

Out[17]:

save the model

In [19]:

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

test the model

In [23]:

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

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

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

In [25]:

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

In [26]:

img**=**image**.**load\_img(r"/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/Veg-dataset/test\_set/Potato\_\_\_Early\_blight/b7883606-5157-4dc1-b965-fc10f8fe1796\_\_\_RS\_Early.B 7598.JPG")

img

Out[26]:



In [27]:

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

img**=**image**.**load\_img(r"/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/Veg-dataset/test\_set/Potato\_\_\_Early\_blight/b7883606-5157-4dc1-b965-fc10f8fe1796\_\_\_RS\_Early.B 7598.JPG",target\_size**=**(128,128))

img

Out[27]:



In [28]:

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

x

Out[28]:

array([[[156., 163., 191.],

[158., 165., 193.],

[155., 162., 190.],

...,

[109., 113., 140.],

[109., 113., 140.],

[113., 117., 144.]],

[[166., 173., 201.],

[166., 173., 201.],

[159., 166., 194.],

...,

[110., 114., 141.],

[104., 108., 135.],

[109., 113., 140.]],

[[168., 175., 203.],

[160., 167., 195.],

[152., 159., 187.],

...,

[110., 114., 141.],

[101., 105., 132.],

[110., 114., 141.]],

...,

[[160., 161., 181.],

[162., 163., 183.],

[155., 156., 176.],

...,

[103., 101., 122.],

[ 99., 97., 118.],

[105., 103., 124.]],

[[155., 156., 176.],

[150., 151., 171.],

[152., 153., 173.],

...,

[109., 107., 128.],

[102., 100., 121.],

[107., 105., 126.]],

[[157., 158., 178.],

[156., 157., 177.],

[149., 150., 170.],

...,

[ 92., 90., 111.],

[119., 117., 138.],

[ 96., 94., 115.]]], dtype=float32)

In [29]:

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

In [30]:

x

Out[30]:

array([[[[156., 163., 191.],

[158., 165., 193.],

[155., 162., 190.],

...,

[109., 113., 140.],

[109., 113., 140.],

[113., 117., 144.]],

[[166., 173., 201.],

[166., 173., 201.],

[159., 166., 194.],

...,

[110., 114., 141.],

[104., 108., 135.],

[109., 113., 140.]],

[[168., 175., 203.],

[160., 167., 195.],

[152., 159., 187.],

...,

[110., 114., 141.],

[101., 105., 132.],

[110., 114., 141.]],

...,

[[160., 161., 181.],

[162., 163., 183.],

[155., 156., 176.],

...,

[103., 101., 122.],

[ 99., 97., 118.],

[105., 103., 124.]],

[[155., 156., 176.],

[150., 151., 171.],

[152., 153., 173.],

...,

[109., 107., 128.],

[102., 100., 121.],

[107., 105., 126.]],

[[157., 158., 178.],

[156., 157., 177.],

[149., 150., 170.],

...,

[ 92., 90., 111.],

[119., 117., 138.],

[ 96., 94., 115.]]]], dtype=float32)

In [31]:

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

1/1 [==============================] - 0s 208ms/step

In [32]:

x\_train**.**class\_indices

Out[32]:

{'Pepper,\_bell\_\_\_Bacterial\_spot': 0,

'Pepper,\_bell\_\_\_healthy': 1,

'Potato\_\_\_Early\_blight': 2,

'Potato\_\_\_Late\_blight': 3,

'Potato\_\_\_healthy': 4,

'Tomato\_\_\_Bacterial\_spot': 5,

'Tomato\_\_\_Late\_blight': 6,

'Tomato\_\_\_Leaf\_Mold': 7,

'Tomato\_\_\_Septoria\_leaf\_spot': 8}

In [35]:

index**=**['Pepper,\_bell\_\_\_Bacterial\_spot','Pepper,\_bell\_\_\_healthy','Potato\_\_\_Early\_blight','Potato\_\_\_Late\_blight','Potato\_\_\_healthy','Tomato\_\_\_Bacterial\_spot','Tomato\_\_\_Late\_blight','Tomato\_\_\_Leaf\_Mold','Tomato\_\_\_Septoria\_leaf\_spot']

In [36]:

index[y[0]]

Out[36]:

'Potato\_\_\_Early\_blight'

In [37]:

img**=**image**.**load\_img(r"/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/Veg-dataset/test\_set/Potato\_\_\_Early\_blight/b7883606-5157-4dc1-b965-fc10f8fe1796\_\_\_RS\_Early.B 7598.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**=**['Pepper,\_bell\_\_\_Bacterial\_spot','Pepper,\_bell\_\_\_healthy','Potato\_\_\_Early\_blight','Potato\_\_\_Late\_blight','Potato\_\_\_healthy','Tomato\_\_\_Bacterial\_spot','Tomato\_\_\_Leaf\_Mold','Tomato\_\_\_Septoria\_leaf\_spot']

index[y[0]]

1/1 [==============================] - 0s 60ms/step

Out[37]:

'Potato\_\_\_Early\_blight'