```
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('fruit-dataset/f_train',
                                          target_size=(128,128),batch_size=24,class_mode='categorical')
x_test=test_datagen.flow_from_directory('fruit-dataset/f_test',
                                        target_size=(128,128),batch_size=24,class_mode='categorical')
Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.
x_train.class_indices
{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
Model: "sequential"
_________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 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

_____

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

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

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

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 [==============================] - 158s 701ms/step - loss: 1.1387 - accuracy: 0.7624 - val_loss: 0.3235 - val_accuracy: 0.8843

Epoch 2/10

225/225 [==============================] - 111s 491ms/step - loss: 0.2539 - accuracy: 0.9116 - val_loss: 0.2671 - val_accuracy: 0.9039

Epoch 3/10

225/225 [==============================] - 118s 525ms/step - loss: 0.1975 - accuracy: 0.9329 - val_loss: 0.1724 - val_accuracy: 0.9478

Epoch 4/10

225/225 [==============================] - 127s 565ms/step - loss: 0.1653 - accuracy: 0.9421 - val_loss: 0.2982 - val_accuracy: 0.9004

Epoch 5/10

225/225 [==============================] - 127s 565ms/step - loss: 0.1533 - accuracy: 0.9461 - val_loss: 0.1494 - val_accuracy: 0.9531

Epoch 6/10

225/225 [==============================] - 113s 500ms/step - loss: 0.1154 - accuracy: 0.9573 - val_loss: 0.1461 - val_accuracy: 0.9520

Epoch 7/10

225/225 [==============================] - 114s 508ms/step - loss: 0.1154 - accuracy: 0.9625 - val_loss: 0.2850 - val_accuracy: 0.9211

Epoch 8/10

225/225 [==============================] - 112s 496ms/step - loss: 0.1047 - accuracy: 0.9627 - val_loss: 0.1464 - val_accuracy: 0.9591

Epoch 9/10

225/225 [==============================] - 107s 475ms/step - loss: 0.1242 - accuracy: 0.9567 - val_loss: 0.1058 - val_accuracy: 0.9656

Epoch 10/10

225/225 [==============================] - 106s 470ms/step - loss: 0.1006 - accuracy: 0.9651 - val_loss: 0.2158 - val_accuracy: 0.9312

score,acc=model.evaluate(x_test,batch_size=128,verbose=2)

acc

71/71 - 5s - loss: 0.2158 - accuracy: 0.9312 - 5s/epoch - 67ms/step

0.9311981201171875

model.save('fruit.h5')