**8.TESTING**

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

model=load\_model('train.h5')

model=load\_model('dataset.h5')

model=load\_model('nutrition.h5')

img=image.load\_img(r"/content/drive/MyDrive

/CNN/Dataset/TEST\_SET/WATERMELON/3\_100.jpg")

img



img=image.load\_img(r"/content/drive/MyDrive

/CNN/Dataset/TEST\_SET/WATERMELON/3\_100.jpg",

target\_size=(64,64))

img



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

x

array([[[[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],

...,  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]]]], dtype=float32)

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

[[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]],  
  
 [[255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.],  
 ...,  
 [255., 255., 255.],  
 [255., 255., 255.],  
 [255., 255., 255.]]], dtype=float32)

pred = model.predict

pred

array

([[0.25227112, 0.17414774, 0.15219809, 0.20493415, 0.21644896],  
 [0.26760292, 0.1759095 , 0.15206912, 0.19424875, 0.21016978],  
 [0.26474723, 0.165203 , 0.14452063, 0.20434381, 0.2211853 ],  
 ...,

[0.24550524, 0.1721549 , 0.16282505, 0.21065485, 0.20885986],  
 [0.25395462, 0.1735253 , 0.16055605, 0.20655352, 0.20541045],  
 [0.24495909, 0.15889102, 0.16927534, 0.20705006, 0.21982446]],  
 dtype=float32

<bound method Model.predict of <keras.engine.

sequential.Sequential object at 0x7f94abfd7c10>>

predict\_x=model.predict(x\_test)

classes\_x=np.argmax(predict\_x,axis=1)

classes\_x

array([0, 0, 0, ..., 0, 0, 0])

x\_test.class\_indices

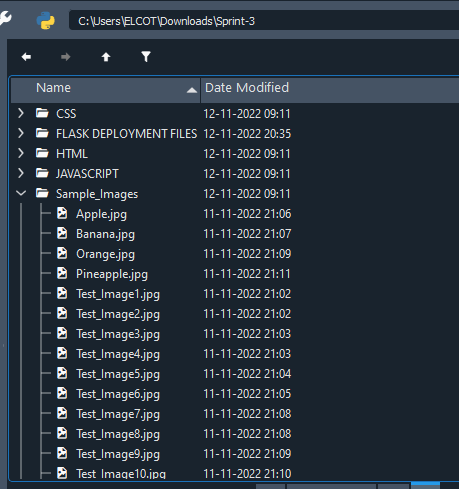
index=['APPLE','BANANA','ORANGE','WATERMELON','PINEAPPLE']

result=str(index[classes\_x[0]])

result

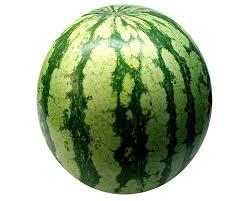
'Watermelon’'

**8.1 TEST CASES**

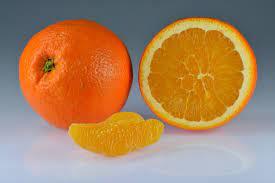


**8.2 USER ACCEPTANCE TESTING**

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is a kind of black box testing where two or more end-users will be involved. Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.



APPLE BANANA ORANGE PINEAPPLE WATERMELON



TEST\_IMAGE1 TEST\_IMAGE2 TEST\_IMAGE3 TEST\_IMAGE4 TEST\_IMAGE5