**ASSIGNMENT-03**

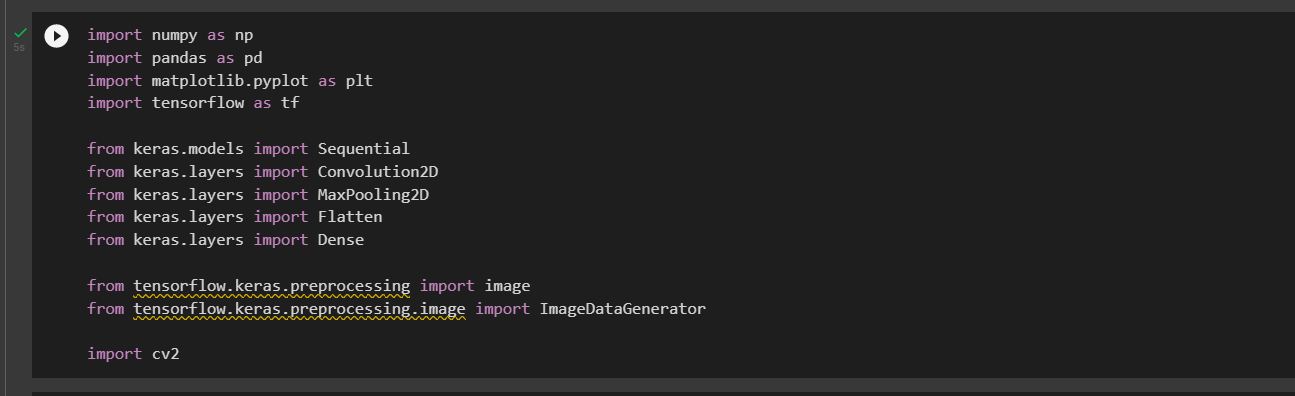
**Build CNN Model for Classification Of Flowers**

|  |  |
| --- | --- |
| **Assignment Date** | 5 October 2022 |
| **Student Name** | S.Nivitha |
| **Student Roll Number** | 113219071027 |
| **Maximum Marks** | 2 Marks |

QUESTION 1:

Download the Dataset

Dataset is downloaded and uploaded



QUESTION 2:

Image Augmentation

data\_path **=** '/content/drive/MyDrive/CNN/flowers'

batch\_size **=** 32

target\_size **=** (64, 64)

train\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255,

shear\_range**=**0.2,

zoom\_range**=**0.2,

width\_shift\_range**=**0.1,

height\_shift\_range**=**0.1,

horizontal\_flip**=True**,

validation\_split**=**0.2)

test\_datagen **=** ImageDataGenerator(rescale**=**1. **/** 255, validation\_split**=**0.2)

X\_train **=** train\_datagen**.**flow\_from\_directory(data\_path,

target\_size**=**target\_size,

batch\_size**=**batch\_size,

subset**=**"training",

class\_mode **=** 'categorical')

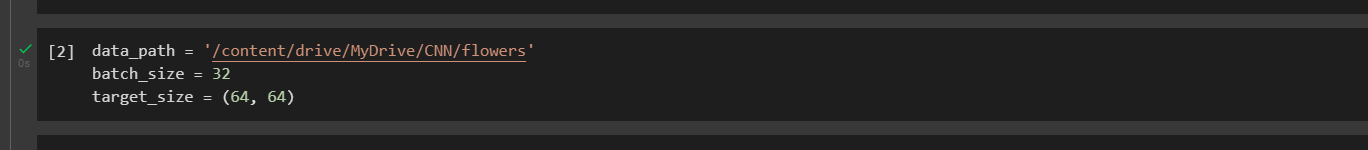
X\_test **=** test\_datagen**.**flow\_from\_directory(data\_path,

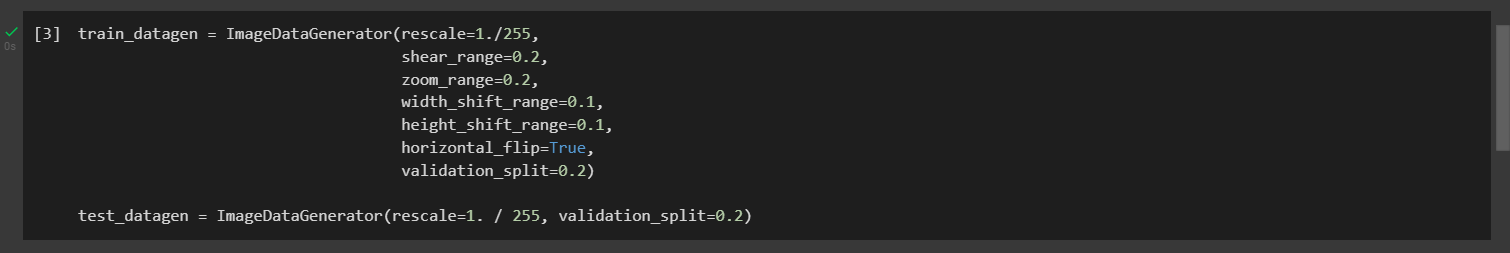
target\_size**=**target\_size,

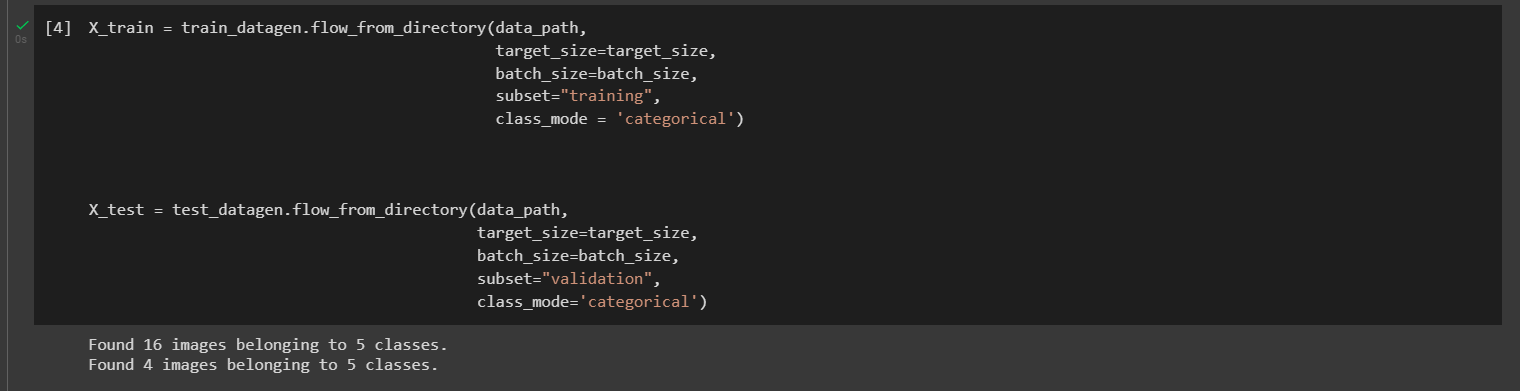
batch\_size**=**batch\_size,

subset**=**"validation",

class\_mode**=**'categorical')



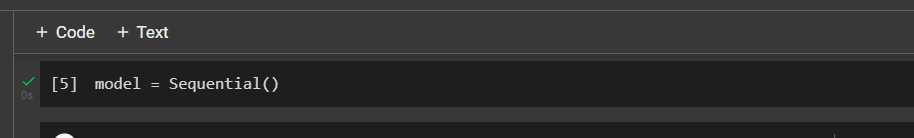




QUESTION 3:

Create Model

model **=** Sequential()



QUESTION 4:

Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output

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

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

model**.**add(Convolution2D(32, (3, 3), activation**=**'relu'))

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

model**.**add(Convolution2D(64, (3, 3), activation**=**'relu'))

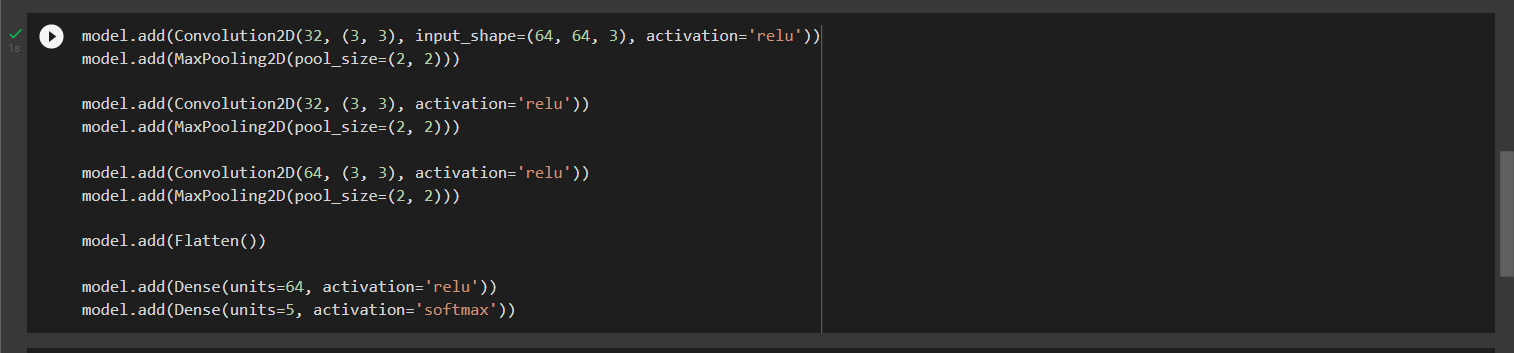
model**.**add(MaxPooling2D(pool\_size**=**(2, 2)))

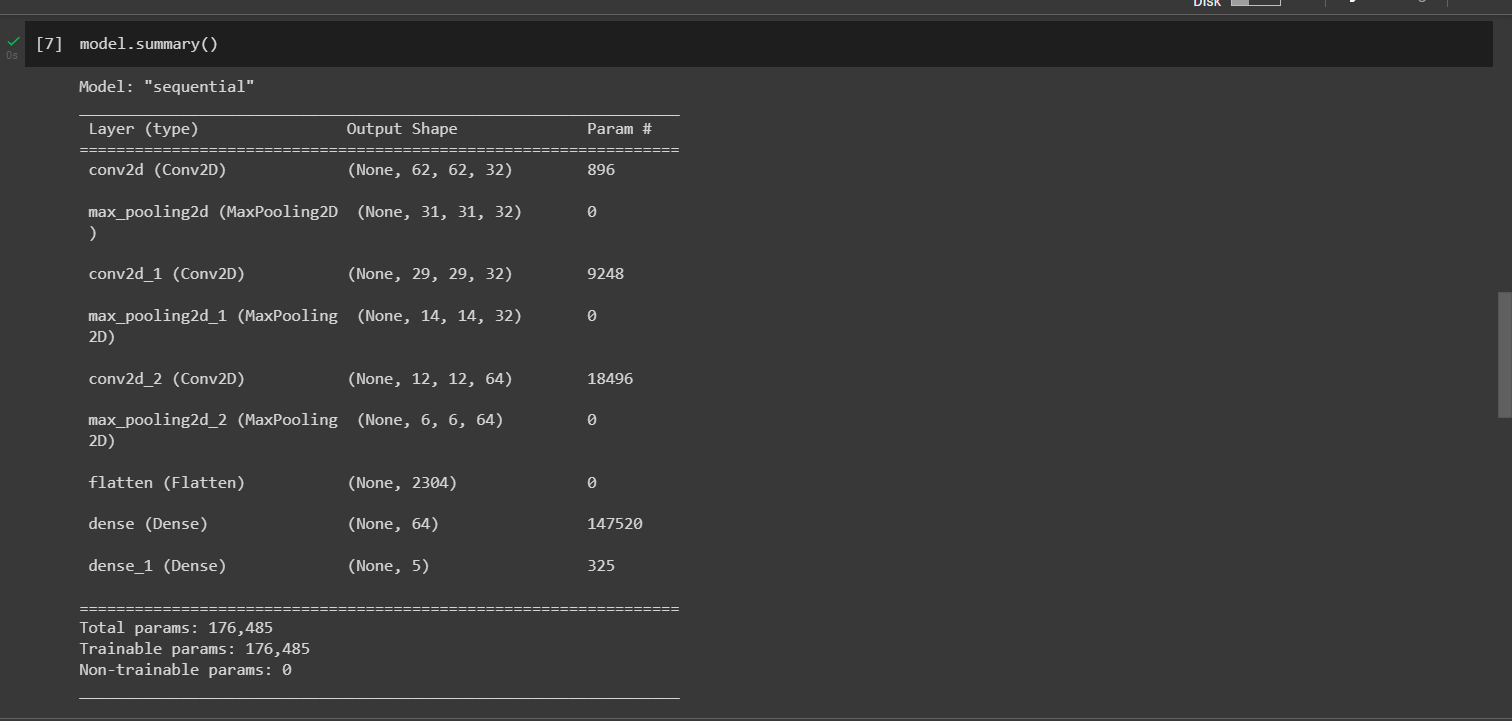
model**.**add(Flatten())

model**.**add(Dense(units**=**64, activation**=**'relu'))

model**.**add(Dense(units**=**5, activation**=**'softmax'))

model**.**summary()

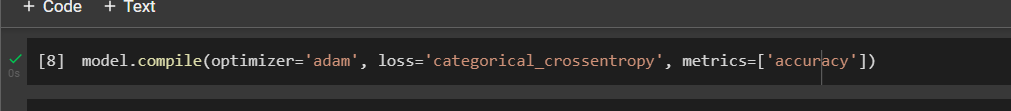




QUESTION 5:

Compile the Model

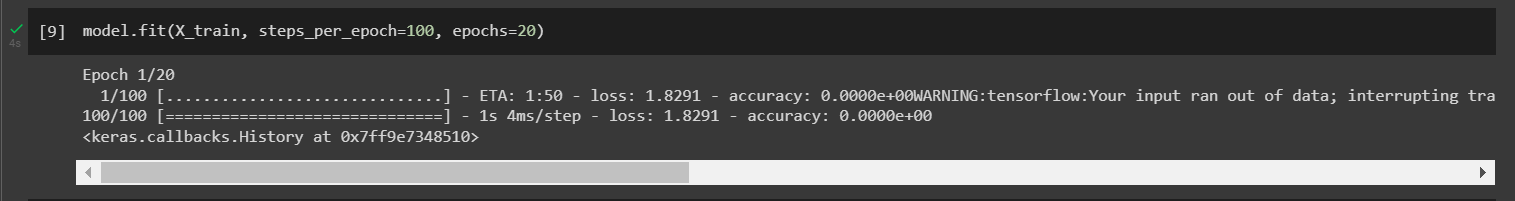
model**.**compile(optimizer**=**'adam', loss**=**'categorical\_crossentropy', metrics**=**['accuracy'])



QUESTION 6:

Fit the model

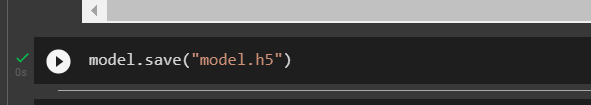
model**.**fit(X\_train, steps\_per\_epoch**=**100, epochs**=**20)



QUESTION 7:

Save the Model

model**.**save("model.h5")



QUESTION 8:

Test the Model

**def** predict():

img **=** image**.**load\_img("/content/drive/MyDrive/CNN/flowers/rose/118974357\_0faa23cce9\_n.jpg", target\_size**=**target\_size)

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

x **=** tf**.**expand\_dims(x,0)

labels **=** ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']

pred **=** model**.**predict(x)

prediction **=** labels[np**.**argmax(pred[0])]

print(f'The given image is a {prediction}')

plt**.**imshow(plt**.**imread("/content/drive/MyDrive/CNN/flowers/rose/118974357\_0faa23cce9\_n.jpg"))

plt**.**axis('off')

plt**.**show()

predict()



