ASSIGNMENT-3

|  |  |
| --- | --- |
| Assignment Date | 30 September 2022 |
| Student Name | MUTHU LAKSHMI.P |
| Student Roll Number | 962719104025 |
| Maximum Marks | 2 Marks |

# Download the Dataset

Link: <https://drive.google.com/file/d/1zZ87e7GDpN90-Sa_AKbvMm3EEfQkEQ_R/view>

# Image Augmentation Solution:

pwd

from tensorflow.keras.preprocessing.image import ImageDataGenerator train\_datagen=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal\_flip=True,ve rtical\_flip=False)

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

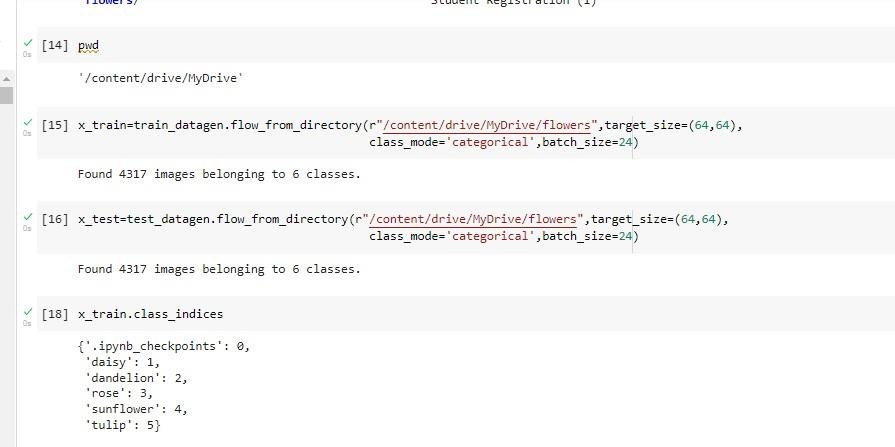
pwd x\_train=train\_datagen.flow\_from\_directory(r"/content/drive/MyDrive/flowers",target\_siz e=(64,64),

class\_mode='categorical',batch\_size=24) x\_test=test\_datagen.flow\_from\_directory(r"/content/drive/MyDrive/flowers",target\_size= (64,64),

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

x\_train.class\_indices



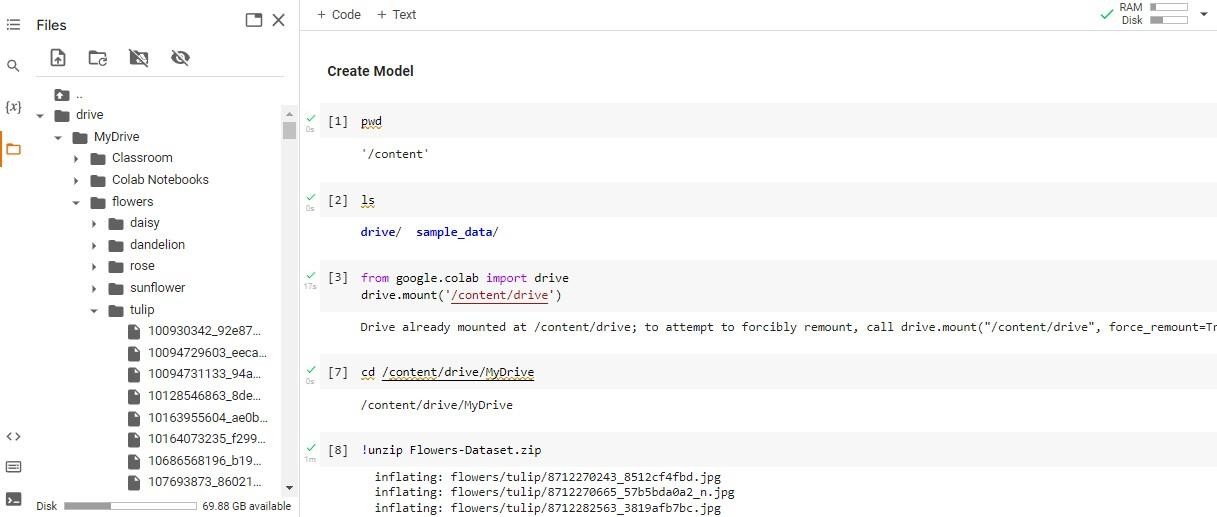


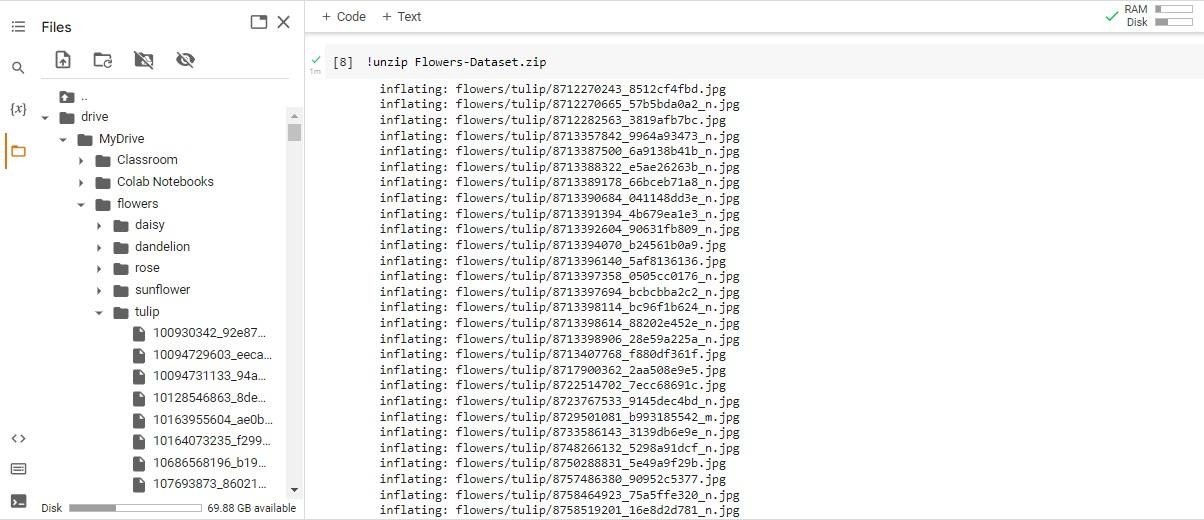
# Create Model Solution:

pwd ls

from google.colab import drive drive.mount('/content/drive') cd /content/drive/MyDrive

!unzip Flowers-Dataset.zip





# Add Layers(Convolution,Maxpooling,Flatten,Dense-(Hidden Layers),Output) Solution:

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=(64,64,3),activation='relu')) model.add(MaxPooling2D(pool\_size=(2,2)))

model.add(Flatten()) model.summary() 32\*(3\*3\*3+1)

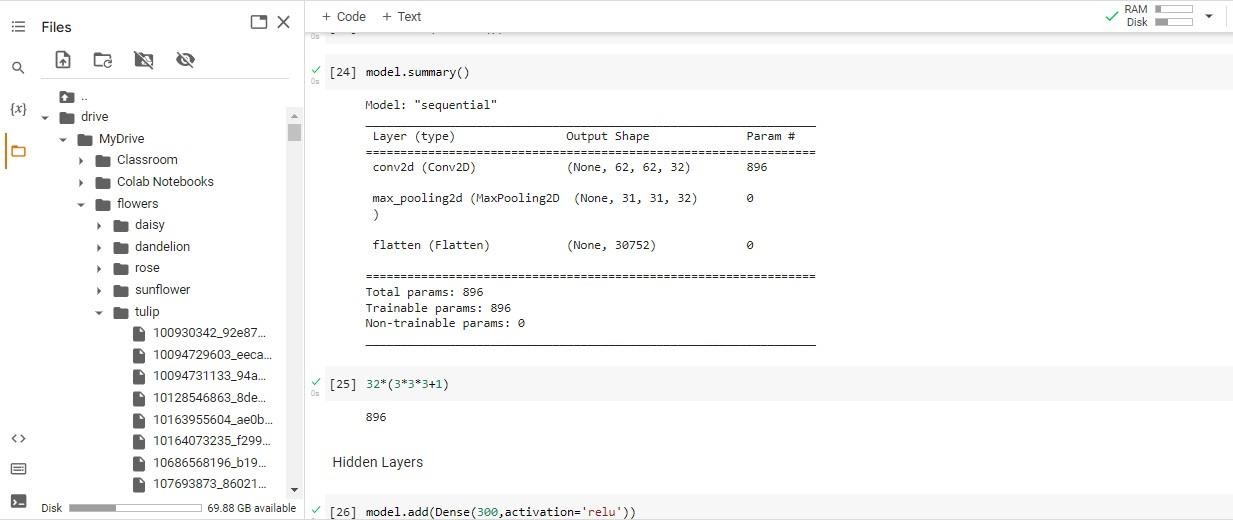
# Hidden layer

model.add(Dense(300,activation='relu')) model.add(Dense(150,activation='relu'))

# Output layer

model.add(Dense(4,activation='softmax'))



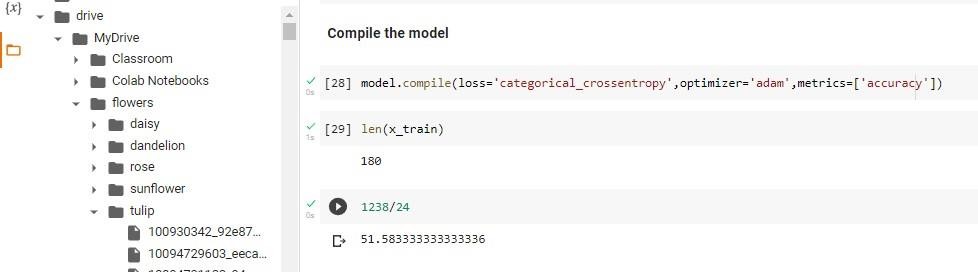




# Compile The Model Solution:

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

1238/24

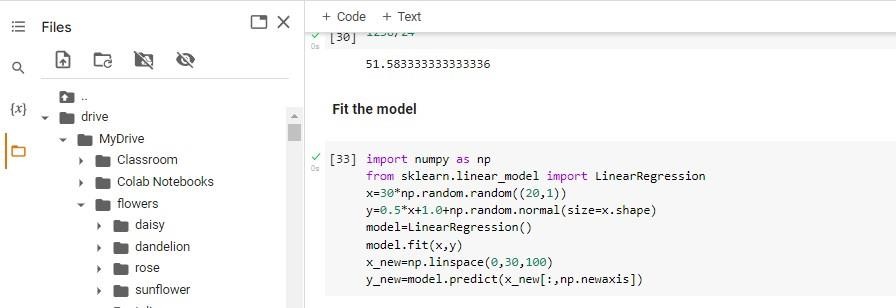


# Fit The Model Solution:

import numpy as np

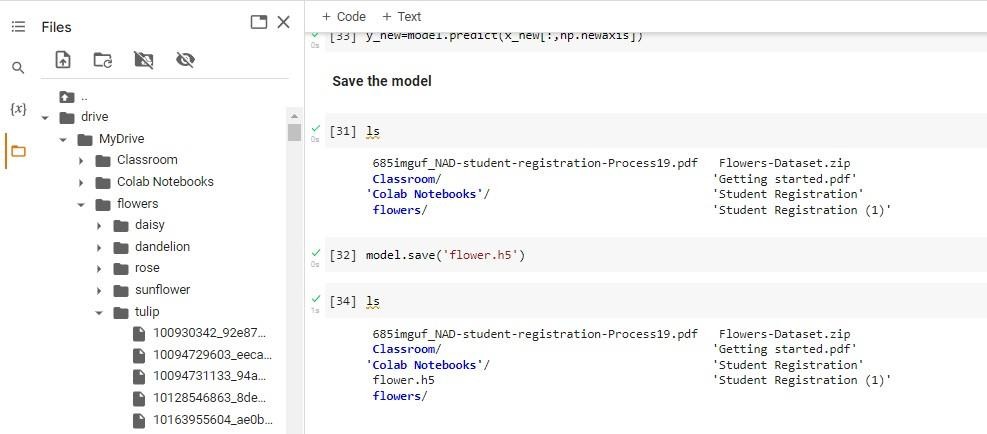
from sklearn.linear\_model import LinearRegression x=30\*np.random.random((20,1)) y=0.5\*x+1.0+np.random.normal(size=x.shape) model=LinearRegression()

model.fit(x,y) x\_new=np.linspace(0,30,100) y\_new=model.predict(x\_new[:,np.newaxis])



# Save The Model Solution:

Ls model.save('flower.h5') ls



# Test The Model Solution:

import numpy as np

from tensorflow.keras.models import load\_model from tensorflow.keras.preprocessing import image #load the model

model=load\_model('flower.h5') img=image.load\_img(r"/content/drive/MyDrive/flowers/rose/10503217854\_e66a804309.jpg") img img=image.load\_img(r"/content/drive/MyDrive/flowers/rose/10503217854\_e66a804309.jpg",t arget\_size=(64,64))

img x=image.img\_to\_array(img) x x=np.expand\_dims(x,axis=0) x

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

x\_train.class\_indices index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]

img=image.load\_img(r"/content/drive/MyDrive/flowers/daisy/100080576\_f52e8ee070\_n.jpg", target\_size=(64,64))

x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]

img=image.load\_img(r"/content/drive/MyDrive/flowers/dandelion/10043234166\_e6dd915111\_n

.jpg",target\_size=(64,64))

x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]

img=image.load\_img(r"/content/drive/MyDrive/flowers/rose/10090824183\_d02c613f10\_m.jpg"

,target\_size=(64,64)) x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]

img=image.load\_img(r"/content/drive/MyDrive/flowers/sunflower/1008566138\_6927679c8a.jp g",target\_size=(64,64))

x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]

img=image.load\_img(r"/content/drive/MyDrive/flowers/tulip/100930342\_92e8746431\_n.jpg", target\_size=(64,64))

x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['daisy','dandelion','rose','sunflower','tulip'] index[y[0]]



