**Assignment -1**

Python Programming

|  |  |
| --- | --- |
| Assignment Date | 07 October 2022 |
| Student Name | Yashika U |
| Student Roll Number | 511919104026 |
| Maximum Marks | 2 Marks |

**Dataset**

**Question-1:**

1. Importing the dataset

**Solution-1:**

from google.colab import drive

drive.mount ('/content/drive')

**Output:**

Mounted at /content/drive

**Solution-2:**

cd/content/drive/MyDrive/

**Output:**

/content/drive/MyDrive

**Solution-3:**

! unzip Flowers-Dataset.zip

**Output:**

Output exceeds the [size limit](command:workbench.action.openSettings?%5B%22notebook.output.textLineLimit%22%5D). Open the full output data [in a text editor](command:workbench.action.openLargeOutput?2e49110e-47af-49cd-adbc-06d6ef0f1ad3)

Archive: Flowers-Dataset.zip inflating: flowers/daisy/100080576\_f52e8ee070\_n.jpg inflating: flowers/daisy/10140303196\_b88d3d6cec.jpg inflating: flowers/daisy/10172379554\_b296050f82\_n.jpg inflating: flowers/daisy/10172567486\_2748826a8b.jpg inflating: flowers/daisy/10172636503\_21bededa75\_n.jpg inflating: flowers/daisy/102841525\_bd6628ae3c.jpg inflating: flowers/daisy/10300722094\_28fa978807\_n.jpg inflating: flowers/daisy/1031799732\_e7f4008c03.jpg inflating: flowers/daisy/10391248763\_1d16681106\_n.jpg inflating: flowers/daisy/10437754174\_22ec990b77\_m.jpg inflating: flowers/daisy/10437770546\_8bb6f7bdd3\_m.jpg inflating: flowers/daisy/10437929963\_bc13eebe0c.jpg inflating: flowers/daisy/10466290366\_cc72e33532.jpg inflating: flowers/daisy/10466558316\_a7198b87e2.jpg inflating: flowers/daisy/10555749515\_13a12a026e.jpg inflating: flowers/daisy/10555815624\_dc211569b0.jpg inflating: flowers/daisy/10555826524\_423eb8bf71\_n.jpg inflating: flowers/daisy/10559679065\_50d2b16f6d.jpg inflating: flowers/daisy/105806915\_a9c13e2106\_n.jpg

inflating: flowers/daisy/10712722853\_5632165b04.jpg inflating: flowers/daisy/107592979\_aaa9cdfe78\_m.jpg inflating: flowers/daisy/10770585085\_4742b9dac3\_n.jpg inflating: flowers/daisy/10841136265\_af473efc60.jpg inflating: flowers/daisy/10993710036\_2033222c91.jpg

...

inflating: flowers/tulip/9870557734\_88eb3b9e3b\_n.jpg inflating: flowers/tulip/9947374414\_fdf1d0861c\_n.jpg inflating: flowers/tulip/9947385346\_3a8cacea02\_n.jpg inflating: flowers/tulip/9976515506\_d496c5e72c.jpg

**Question-2:**

2. Image Augmentation

**Solution-1:**

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(r"/content/drive/MyDrive/flowers",target\_size =(64,64),class\_mode ='categorical',batch\_size=24)

**Output:**

Found 4317 images belonging to 5 classes.

**Solution-2:**

x\_train.class\_indices

**Output:**

{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

**Solution-3:**

xtest = test\_datagen.flow\_from\_directory('/content/drive/MyDrive/flowers',

target\_size=(64,64),

class\_mode='categorical',

batch\_size=100)

**Output:**

Found 4317 images belonging to 5 classes.

**Question-3:**

3. Create model and adding layers

**Solution:**

from tensorflow.keras.models import Sequential

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

model = Sequential() #Initializing sequential model

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

model.add(MaxPooling2D(pool\_size=(2,2))) #MaxPooling layer

model.add(Flatten()) #Flatten layer

model.add(Dense(300,activation='relu')) #Hidden layer 1

model.add(Dense(150,activation='relu')) #Hidden layer 2

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

**Output:**

Model created successfully

**Question-4:**

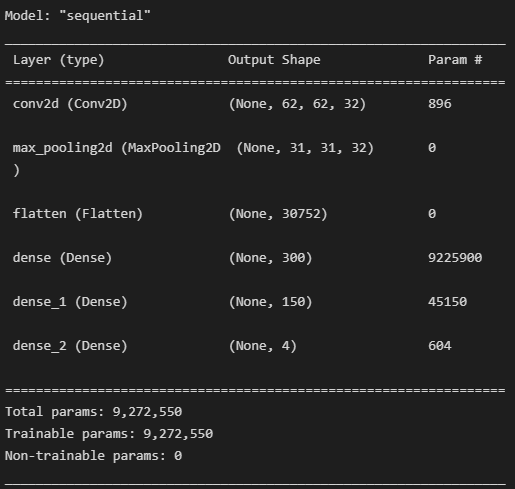
4. Compile the model

**Solution:**

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

model.summary()

**Output:**



**Question-5:**

5. Fit the model

**Solution:**

from keras.callbacks import EarlyStopping, ReduceLROnPlateau

early\_stopping = EarlyStopping(monitor='val\_accuracy', patience=5)

reduce\_lr = ReduceLROnPlateau(monitor='val\_accuracy', patience=5, factor=0.5,min\_lr=0.00001)

callback = [reduce\_lr,early\_stopping]

model.fit(x\_train,

steps\_per\_epoch=len(x\_train),

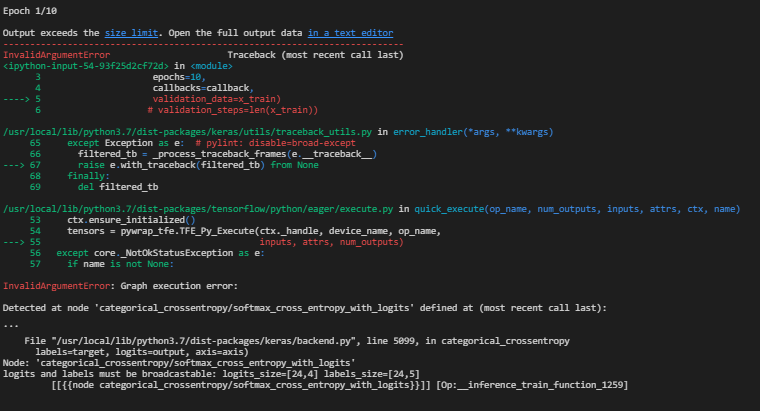
epochs=50,

callbacks=callback,

validation\_data=x\_train,

validation\_steps=len(x\_train))

**Output:**



**Question-6:**

6. Save the model

**Solution:**

model.save('flowers.h5')

**Output:**

'Model saved

**Question-7:**

7. Test the model

**Solution-1:**

img=image.load\_img('/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)

pred=np.argmax(model.predict(x))

op=['daisy','dandelion','rose','sunflower','tulip']

op[pred]

**Output:**

‘sunflower'

**Solution-2:**

img=image.load\_img('/content/drive/MyDrive/flowers/sunflower/1008566138\_6927679c8a.jpg',target\_size=(64,64))

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

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

pred=np.argmax(model.predict(x))

op=['daisy','dandelion','rose','sunflower','tulip']

op[pred]

**Output:**

‘sunflower'