

Assignment - 3
Python Programming

Assignment Date	07 October 2022
Student Name	Yashika.U
Student Roll Number	511919104025
Maximum Marks	2 Marks

Dataset

Question-1:

- Importing the dataset

Solution-1:

```
from google.colab import  
drivedrive.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. Open the full output data in a text editor
```

```
Archive: Flowers-Dataset.zip inflating:  
flowers/daisy/100080576_f52e8ee070_n.jpginflating:  
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/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
```

- Image Augmentation

Solution-1:

```
from tensorflow.keras.preprocessing.image import

ImageDataGeneratortrain_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/M  
yDrive/flowers',target_size=(64,64),  
class_mode='categorical'  
,batch_size=100)
```

Output:

```
Found 4317 images belonging to 5 classes.
```

- 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:

- Compile the model

Solution:

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.summary()
```

Output:

```
Model: "sequential"
-----
Layer (type)                 Output Shape              Param #
-----
conv2d (Conv2D)              (None, 62, 62, 32)       896
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)       0
flatten (Flatten)            (None, 30752)             0
dense (Dense)                 (None, 300)              9225900
dense_1 (Dense)               (None, 150)              45150
dense_2 (Dense)               (None, 4)                604
-----
Total params: 9,272,550
Trainable params: 9,272,550
Non-trainable params: 0
```

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:

```
Epoch 1/10  
Output exceeds the size limit. Open the full output data in a text editor  
-----  
InvalidArgumentError                                Traceback (most recent call last)  
<ipython-input-54-93f25d2cf72d> in <module>  
      3         epochs=10,  
      4         callbacks=callback,  
----> 5         validation_data=x_train)  
      6         # validation_steps=len(x_train))  
  
/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py in error_handler(*args, **kwargs)  
    65     except Exception as e: # pylint: disable=broad-except  
    66         filtered_tb = _process_traceback_frames(e.__traceback__)  
--> 67         raise e.with_traceback(filtered_tb) from None  
    68     finally:  
    69         del filtered_tb  
  
/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)  
    53     ctx.ensure_initialized()  
    54     tensors = pywrap_tfe.TFE_Py_Execute(ctx.handle, device_name, op_name,  
--> 55                                         inputs, attrs, num_outputs)  
    56 except core._NotOkStatusException as e:  
    57     if name is not None:  
  
InvalidArgumentError: Graph execution error:  
  
Detected at node 'categorical_crossentropy/softmax_cross_entropy_with_logits' defined at (most recent call last):  
...  
File "/usr/local/lib/python3.7/dist-packages/keras/backend.py", line 5099, in categorical_crossentropy  
    labels=target, logits=output, axis=axis)  
Node: 'categorical_crossentropy/softmax_cross_entropy_with_logits'  
logits and labels must be broadcastable: logits_size=[24,4] labels_size=[24,5]  
[[{{node categorical_crossentropy/softmax_cross_entropy_with_logits}}]] [Op:__inference_train_function_1259]
```

```
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_e6  
dd915111_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_692
7679c8a.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'
```