

### Assignment - 3

Assignment Date	03.10.2022
Student Name	Abirami Rathina
Student Roll Number	2019115006
Maximum Marks	2 Marks

#### Question-1

Download the Dataset

```
#Importing Packages

In [50]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
         from tensorflow.keras.preprocessing.image import ImageDataGenerator as idm
         import numpy as np
         import warnings
         #Supressing warnings|
         warnings.filterwarnings('ignore')
```

#### Question-2

Image Augmentation

**Solution:**

Creating augmentation on training variable

```
train_flowers=idm(rescale=1./255,zoom_range=0.2,hor
izontal_flip=True)
```

# Passing training data to train variable

Xtrain =

```
train_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flowers-
Dataset',target_size=(76,76),class_mode='categorical',batch_
size=100)
```

# Creating augmentation on testing variable

```
test_flowers=idm(rescale=1./255)
```

# Passing testing data to test variable

Xtest =

```
test_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flower_Training',target_size=(76,76),class_mode='categorical',batch_size=100)
```

**Screenshot:**

```
2.Image Augmentation

In [51]: # Creating augmentation on training variable
         train_flowers=idm(rescale=1./255,zoom_range=0.2,horizontal_flip=True)

         # Passing training data to train variable
         Xtrain = train_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flowers-Dataset',target_size=(76,76),class_mode='categorical',batch_size=100)

         Found 4141 images belonging to 5 classes.

In [52]: # Creating augmentation on testing variable
         test_flowers=idm(rescale=1./255)

         # Passing testing data to test variable
         Xtest = test_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flower_Training',target_size=(76,76),class_mode='categorical',batch_size=100)

         Found 204 images belonging to 5 classes.
```

#### Question-3

Create Model

**Solution:**

```
# Creating augmentation on training variable
train_flow=ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
```

```
# Passing training data to train variable
Xtrain =
train_flow.flow_from_directory('/content/drive/MyDrive/IBM/Flowers-Dataset', target_size=(76,76), class_mode='categorical', batch_size=100)
```

```
# Creating augmentation on testing variable
test_flow=ImageDataGenerator(rescale=1./255)
```

```
# Passing testing data to test variable
Xtest =
test_flow.flow_from_directory('/content/drive/MyDrive/IBM/Flower_Training', target_size=(76,76), class_mode='categorical', batch_size=100)
```

**Screenshot:**



#### Question-4

Compile The Model

**Solution:**

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

**Screenshot:**



#### Question-5

Fit The Model

**Solution:**

```
Flower_model.fit_generator(Xtrain,steps_per_epoch=
len (Xtrain),epochs=
10,validation_data=Xtest,validation_steps= len (Xtest))
```

**Screenshot:**

```
#5: Fit the Model

In [55]: Flower_model.fit_generator(Xtrain,steps_per_epoch= len (Xtrain),epochs= 10,validation_data=Xtest,validation_steps= len (Xtest))

Epoch 1/10
42/42 [=====] - 567s 14s/step - loss: 1.9592 - accuracy: 0.3700 - val_loss: 1.1356 - val_accuracy: 0.5496
Epoch 2/10
42/42 [=====] - 26s 618ms/step - loss: 1.1221 - accuracy: 0.5412 - val_loss: 1.1446 - val_accuracy: 0.6422
Epoch 3/10
42/42 [=====] - 26s 612ms/step - loss: 1.0173 - accuracy: 0.6042 - val_loss: 1.1835 - val_accuracy: 0.6225
Epoch 4/10
42/42 [=====] - 26s 611ms/step - loss: 0.9552 - accuracy: 0.6264 - val_loss: 1.0033 - val_accuracy: 0.6765
Epoch 5/10
42/42 [=====] - 26s 620ms/step - loss: 0.8832 - accuracy: 0.6619 - val_loss: 0.9993 - val_accuracy: 0.7059
Epoch 6/10
42/42 [=====] - 26s 621ms/step - loss: 0.8373 - accuracy: 0.6783 - val_loss: 0.9690 - val_accuracy: 0.7286
Epoch 7/10
42/42 [=====] - 26s 615ms/step - loss: 0.8125 - accuracy: 0.6923 - val_loss: 0.8731 - val_accuracy: 0.7659
Epoch 8/10
42/42 [=====] - 26s 608ms/step - loss: 0.7663 - accuracy: 0.7073 - val_loss: 1.0149 - val_accuracy: 0.6667
Epoch 9/10
42/42 [=====] - 26s 616ms/step - loss: 0.7333 - accuracy: 0.7242 - val_loss: 0.9583 - val_accuracy: 0.6863
Epoch 10/10
42/42 [=====] - 26s 613ms/step - loss: 0.7128 - accuracy: 0.7262 - val_loss: 0.9158 - val_accuracy: 0.7206

Out[55]: <keras.callbacks.History at 0x7fd5aec82f50>
```

### Question-6

Save The Model

**Solution:**

Flower\_model.save('Flower.h5')

**Screenshot:**

```
#7: Save the model

In [56]: Flower_model.save('Flower.h5')

#8: Test the model
```

### Question-7

Test The Model

**Solution:**

```
test_img=image.load_img('/content/drive/MyDrive/IBM/Flowers-
Dataset/sunflower/200557977_bf24d9550b.jpg',target_size=(
76,76))
test_img
```

```
x=image.img_to_array(test_img)
x=np.expand_dims(x,axis=0)
predicted=np.argmax(Flower_model.predict(x))
Prediction_category=['daisy','dandelion','rose','sunflower','tulip']
Prediction_category[predicted]
```

```
test_img1=image.load_img('/content/drive/MyDrive/IBM/Flowers-
Dataset/daisy/1140299375_3aa7024466.jpg',target_size=(76,
76))
test_img1
```





```
x=image.img_to_array(test_img1)
x=np.expand_dims(x,axis=0)
predicted=np.argmax(Flower_model.predict(x))
```

Prediction\_category[predicted]

```
test_img2=image.load_img('/content/drive/MyDrive/IBM/Flowers-  
Dataset/rose/7251352826_69b62cba2c_m.jpg',target_size=(7  
6,76))  
test_img2
```

```
x=image.img_to_array(test_img2)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category[predicted]
```

**Screenshot:**

```
#8. Test the model  
  
In [60]: test_img=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/sunflower/200557977_bf24d9550b.jpg',target_size=(76,76))  
test_img  
Out[60]:   
  
In [61]: x=image.img_to_array(test_img)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category=[ 'daisy', 'dandelion', 'rose', 'sunflower', 'tulip' ]  
Prediction_category[predicted]  
Out[61]: 'sunflower'  
  
In [59]: test_img1=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/daisy/1140299375_3aa7024466.jpg',target_size=(76,76))  
test_img1  
Out[59]:   
  
In [46]: x=image.img_to_array(test_img1)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category=[predicted]  
In [59]: test_img1=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/daisy/1140299375_3aa7024466.jpg',target_size=(76,76))  
test_img1  
Out[59]:   
  
In [46]: x=image.img_to_array(test_img1)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category=[predicted]  
Out[46]: 'daisy'  
  
In [65]: test_img2=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/rose/7251352826_69b62cba2c_m.jpg',target_size=(76,76))  
test_img2  
Out[65]:   
  
In [66]: x=image.img_to_array(test_img2)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category=[predicted]  
Out[66]: 'rose'
```