Assignment -3

Assignment Date	7 October 2022
Student Name	Javith B
Student Roll Number	621319106307
Maximum Marks	2 Marks

Question-1

1. Downloading and unzipping dataset

Solution:

```
!unzip 'drive/MyDrive/Assignment3data/Flowers-Dataset.zip'
```

```
Output exceeds the size limit. Open the full output data in a text editor
Archive:  drive/MyDrive/Assignment3data/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/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:

```
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
batch_size = 32
img_height = 180
img_width = 180
data_dir = "/content/flowers"
```

```
train_datagen = ImageDataGenerator(rescale = 1./255, horizontal_flip = True, vertical_flip = True, zoom_range = 0.2)
```

```
x_train = train_datagen.flow_from_directory('/content/flowers',
                                            target_size=(64,64),
                                            class_mode='categorical',
                                            batch_size=100)
```

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

Question-3

```
data_augmentation = Sequential(
  [
    layers.RandomFlip("vertical",input_shape=(img_height, img_width, 3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)
```

3. Creating Model

Solution:

```
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
model = Sequential()
```

```
training_ds = tf.keras.utils.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="training",
  seed=57,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

```
Found 4317 files belonging to 5 classes.
Using 3454 files for training.
```

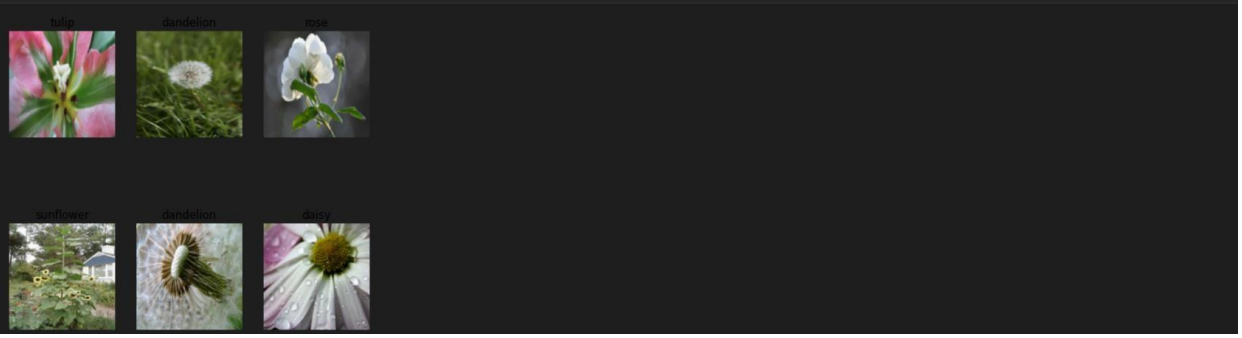
```
validation_ds = tf.keras.utils.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="validation",
seed=107,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

```
Found 4317 files belonging to 5 classes.
Using 863 files for validation.
```

```
training_ds.class_names
```

```
['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
```

```
plt.figure(figsize=(7, 7))
for data, labels in training_ds.take(1):
  for i in range(6):
    ax = plt.subplot(2, 3, i + 1)
    plt.imshow(data[i].numpy().astype("uint8"))
    plt.title(training_ds.class_names[labels[i]])
    plt.axis("off")
```



Question-3a

3a. Convolution layer

Solution:

```
model.add(Convolution2D(32, (3,3), activation = "relu", input_shape = (64,64,3) ))
```

Question-3b

3b. Maxpooling layer

Solution:

```
model.add(MaxPooling2D(pool_size = (2,2)))
```

Question-3c

3c. Flatten

Solution:

```
model.add(Flatten())
```

Question-3d

3d. Hidden/dense layers

Solution:

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

Question-3e

3e. Output layer

Solution:

```
model.add(Dense(5, activation = "softmax"))
```

Question-4

4. Compiling Model

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

```
    model.fit(x_train, epochs = 15, steps_per_epoch = len(x_train))
```

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/15
44/44 [==============================] - 29s 642ms/step - loss: 1.5728 - accuracy: 0.3806
Epoch 2/15
44/44 [==============================] - 29s 657ms/step - loss: 1.1663 - accuracy: 0.5110
Epoch 3/15
44/44 [==============================] - 30s 682ms/step - loss: 1.0635 - accuracy: 0.5793
Epoch 4/15
44/44 [==============================] - 28s 639ms/step - loss: 1.0511 - accuracy: 0.5807
Epoch 5/15
44/44 [==============================] - 28s 635ms/step - loss: 0.9920 - accuracy: 0.6018
Epoch 6/15
44/44 [==============================] - 29s 660ms/step - loss: 0.9578 - accuracy: 0.6169
Epoch 7/15
44/44 [==============================] - 30s 676ms/step - loss: 0.9241 - accuracy: 0.6382
Epoch 8/15
44/44 [==============================] - 30s 669ms/step - loss: 0.9054 - accuracy: 0.6416
Epoch 9/15
44/44 [==============================] - 32s 714ms/step - loss: 0.8622 - accuracy: 0.6674
Epoch 10/15
44/44 [==============================] - 28s 639ms/step - loss: 0.8449 - accuracy: 0.6750
Epoch 11/15
44/44 [==============================] - 34s 750ms/step - loss: 0.8143 - accuracy: 0.6838
Epoch 12/15
44/44 [==============================] - 28s 632ms/step - loss: 0.8152 - accuracy: 0.6808
Epoch 13/15
...
Epoch 14/15
44/44 [==============================] - 29s 660ms/step - loss: 0.7761 - accuracy: 0.7012
Epoch 15/15
44/44 [==============================] - 31s 695ms/step - loss: 0.7474 - accuracy: 0.7072

<keras.callbacks.History at 0x7f35de9674d0>
```

Question-5

5. Save The Model

Solution:

```
model.save("flowers.h1")
```

Question-6

6. Test The Model

Solution:

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

```
model = load_model("/content/flowers.h1")
```

```
daisy_img = image.load_img('/content/flowers/daisy/100080576_f52e8ee070_n.jpg',target_size=(64,64))
x = image.img_to_array(daisy_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)
```

```
labels = ['daisy','dandelion','roses','sunflowers','tulips']
labels[np.argmax(predicted_class)]
```

```
'daisy'
```

```
daisy_img
```