

ASSIGNMENT-3

Assignment Date	30 September 2022
Student Name	M.Hema Preethi
Student Roll Number	962719106009
Maximum Marks	2 Marks

1. Download the Dataset

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

2. Image Augmentation

Solution:

```
pwd
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)
ls
pwd
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/flowers", target_size=(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
```

```
Image Augmentation

[9] pwd
/content/drive/MyDrive

[10] from tensorflow.keras.preprocessing.image import ImageDataGenerator

[11] train_datagen=ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True, vertical_flip=False)

[12] test_datagen=ImageDataGenerator(rescale=1./255)

[13] ls
685imguf_NAD-student-registration-Process19.pdf  Flowers-Dataset.zip
Classroom/                                       'Getting started.pdf'
'Colab Notebooks'/'                             'Student Registration'
flowers/                                         'Student Registration (1)'

[14] pwd
/content/drive/MyDrive
```

```

[14] pwd

'/content/drive/MyDrive'

[15] x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/flowers",target_size=(64,64),
class_mode='categorical',batch_size=24)

Found 4317 images belonging to 6 classes.

[16] x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/flowers",target_size=(64,64),
class_mode='categorical',batch_size=24)

Found 4317 images belonging to 6 classes.

[18] x_train.class_indices

{'_ipython_checkpoints': 0,
 'daisy': 1,
 'dandelion': 2,
 'rose': 3,
 'sunflower': 4,
 'tulip': 5}

```

3. Create Model

Solution:

```

pwd
ls
from google.colab import drive
drive.mount('/content/drive')
cd /content/drive/MyDrive
!unzip Flowers-Dataset.zip

```



The screenshot shows the Google Colab interface. On the left, the 'Files' pane displays the directory structure: drive > MyDrive > flowers > tulip. The 'tulip' folder contains several image files. On the right, the 'Code' pane shows the following commands being executed:

```

[1] pwd

'/content'

[2] ls

drive/ sample_data/

[3] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

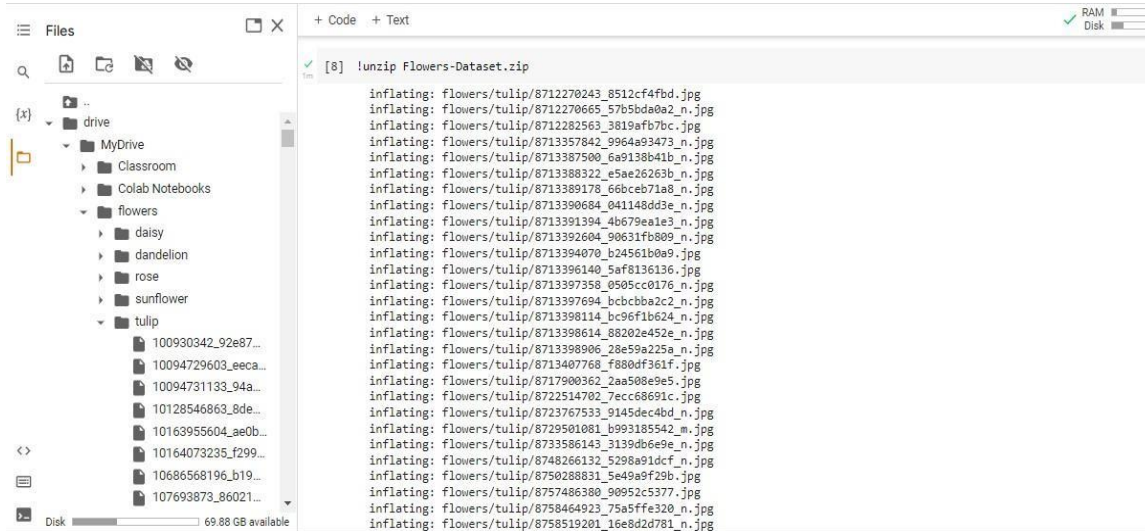
[7] cd /content/drive/MyDrive

/content/drive/MyDrive

[8] !unzip Flowers-Dataset.zip

inflating: flowers/tulip/8712270243_8512cf4fbd.jpg
inflating: flowers/tulip/8712270665_57b5bda0a2_n.jpg
inflating: flowers/tulip/8712282563_3819afb7bc.jpg

```



4. 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'))
```



The screenshot displays a Jupyter Notebook environment. On the left, a file explorer shows a directory structure with 'drive' containing 'MyDrive', 'Classroom', 'Colab Notebooks', and 'flowers'. The 'flowers' directory contains subdirectories 'daisy', 'dandelion', 'rose', 'sunflower', and 'tulip'. The 'tulip' directory contains several files with names like '100930342_92e87...', '10094729603_eeca...', etc. The right pane shows the code cells:

```
[24] model.summary()

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
-----
Total params: 896
Trainable params: 896
Non-trainable params: 0

[25] 32*(3*3*3+1)

896

Hidden Layers

[26] model.add(Dense(300,activation='relu'))
```

The second screenshot shows the continuation of the code cells:

```
[25] 32*(3*3*3+1)

896

Hidden Layers

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

Output Layers

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

Compile the model

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

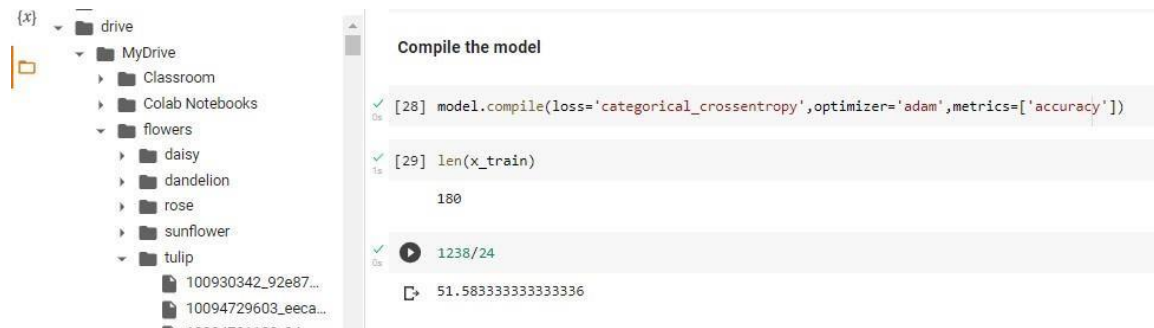
[29] len(x_train)

180
```

5. Compile The Model

Solution:

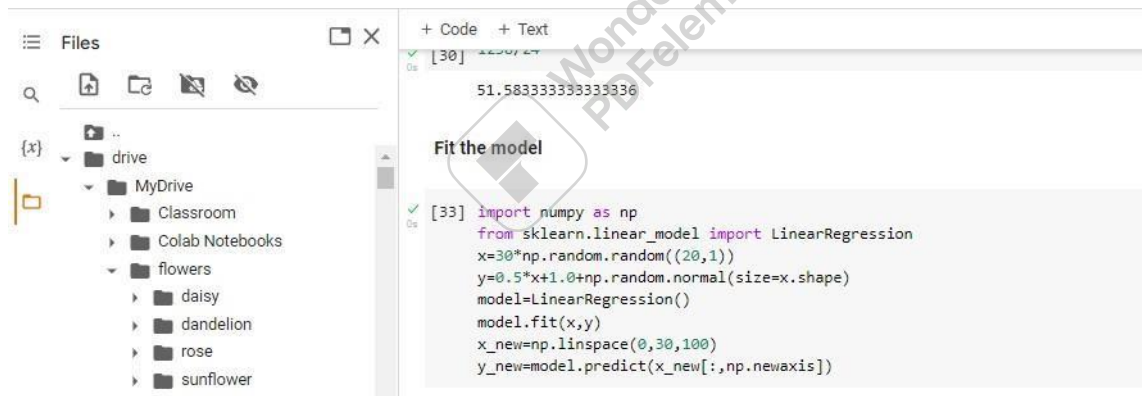
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
1238/24
```



6. Fit The Model

Solution:

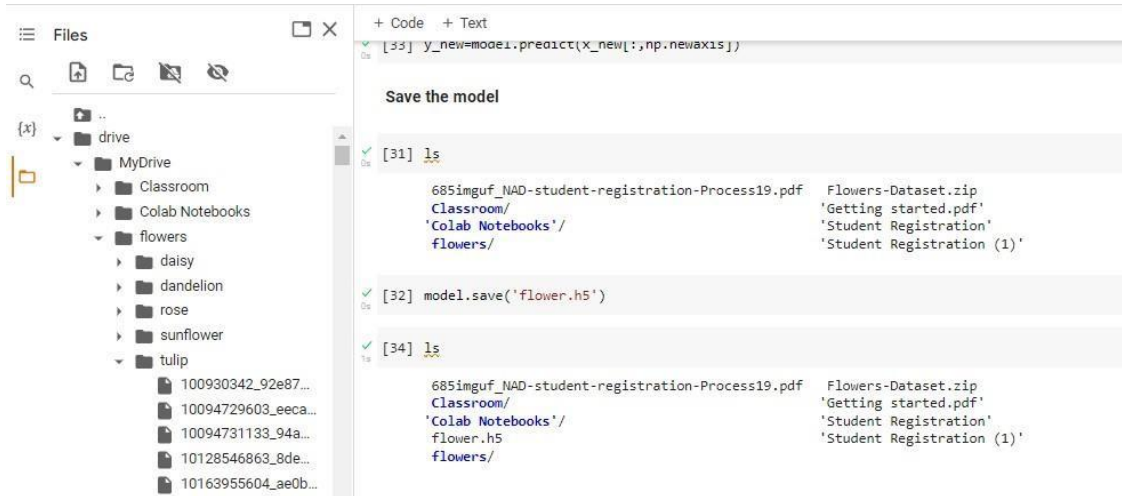
```
import numpy as np
from sklearn.linear_model import LinearRegression
x=30*np.random.random((20,1))
y=0.5*x+1.0+np.random.normal(size=x.shape)
model=LinearRegression()
model.fit(x,y)
x_new=np.linspace(0,30,100)
y_new=model.predict(x_new[:,np.newaxis])
```



7. Save The Model

Solution:

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



8. Test The Model

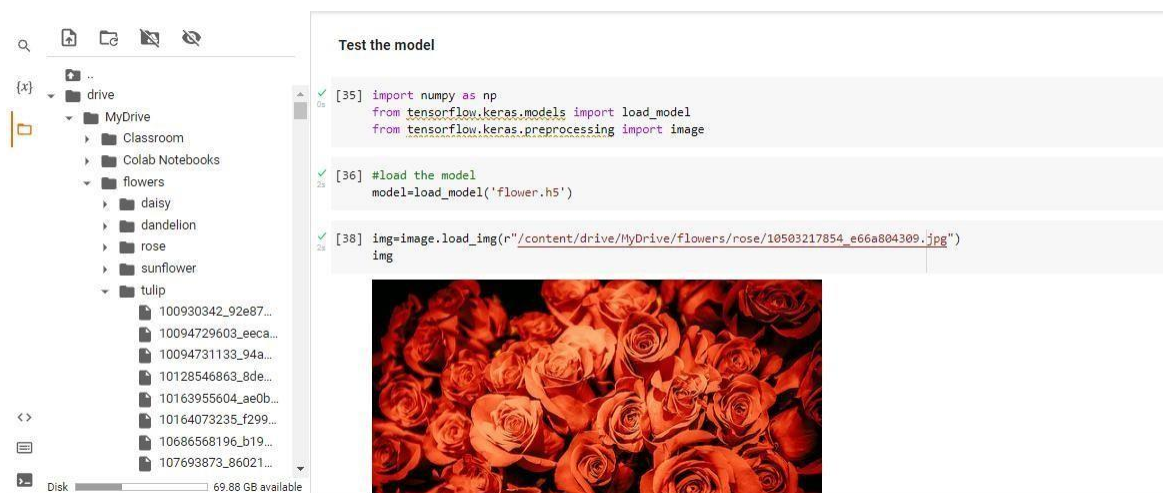
Solution:

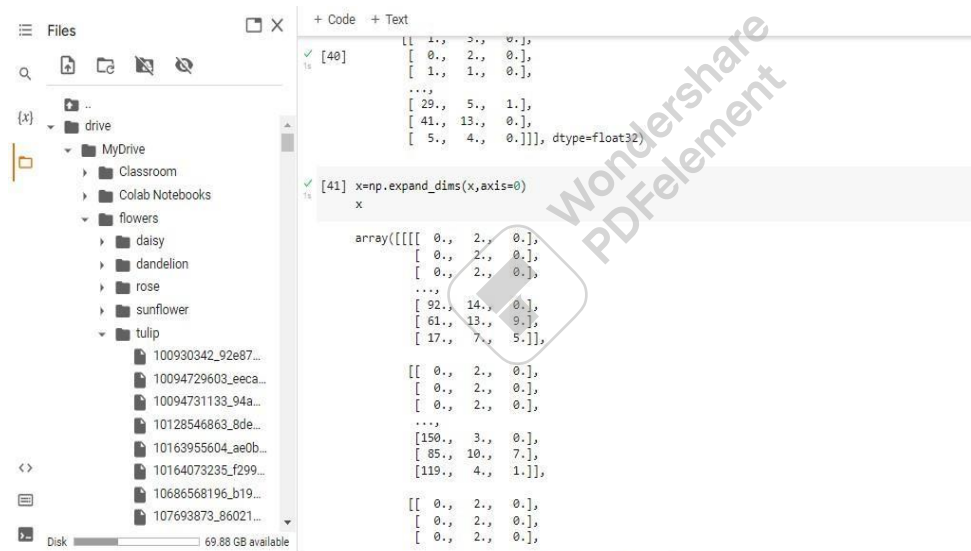
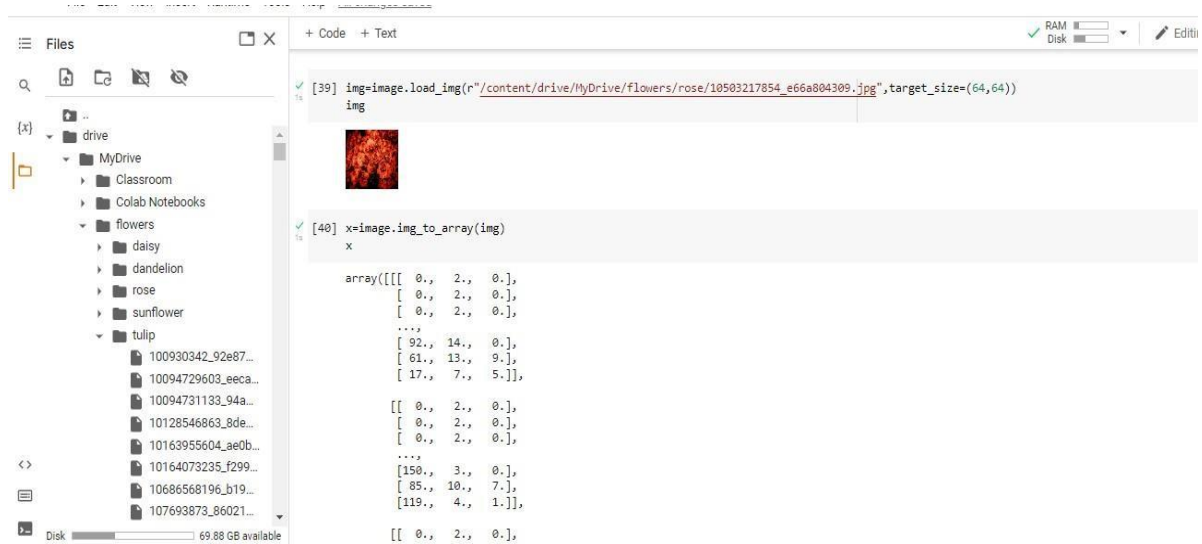
```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
#load the model
model=load_model('flower.h5')
img=image.load_img(r"/content/drive/MyDrive/flowers/rose/10503217854_e66a804309.jpg")
img
img=image.load_img(r"/content/drive/MyDrive/flowers/rose/10503217854_e66a804309.jpg",t
target_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))
```

```
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]]
```





The screenshot shows a Jupyter Notebook interface. On the left is a file explorer sidebar with a tree view showing a directory structure: a root folder containing 'drive', 'MyDrive', 'Classroom', 'Colab Notebooks', and 'flowers'. The 'flowers' folder is expanded, showing subfolders 'daisy', 'dandelion', 'rose', 'sunflower', and 'tulip'. The 'tulip' folder is selected, displaying a list of image files with names like '100930342_92e87...', '10094729603_eeca...', etc. The main area on the right contains a code editor with five cells. The first cell (index 42) runs `y=np.argmax(model.predict(x),axis=1)` and outputs `array([3])`. The second cell (index 43) runs `x_train.class_indices` and outputs a dictionary mapping class names to indices: `{'.ipynb_checkpoints': 0, 'daisy': 1, 'dandelion': 2, 'rose': 3, 'sunflower': 4, 'tulip': 5}`. The third cell (index 44) runs `index=['daisy','dandelion','rose','sunflower','tulip']` and outputs the list. The fourth cell (index 46) runs `index[y[0]]` and outputs `'sunflower'`. The fifth cell (index 50) runs `img=image.load_img(r"/content/drive/MyDrive/flowers/daisy/100080576_f52e8ee070_n.jpg",target_size=(64,64))` and outputs `x=image.img to array(img)`. Each cell has a green checkmark icon indicating successful execution.

The screenshot shows a Jupyter Notebook environment. On the left, a file explorer displays the directory structure: 'drive' > 'MyDrive' > 'Classroom' > 'Colab Notebooks' > 'flowers'. The 'flowers' directory contains subfolders for 'daisy', 'dandelion', 'rose', and 'sunflower', each with a corresponding 'tulip' subfolder. The main area shows two code cells. The first cell, labeled '[3]', contains code to load an image from 'content/drive/mydrive/flowers/sunflower/100930342_92e870430_03x7070108.jpg', convert it to an array, and use a model to predict the class. The output shows the predicted class is 'rose'. The second cell, labeled '[54]', contains similar code for an image from 'content/drive/mydrive/flowers/tulip/100930342_92e8746431_n.jpg'. The output shows the predicted class is 'dandelion'. The bottom status bar indicates '69.88 GB available'.