```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Task

1. Downloading and unzipping dataset

```
!unzip '/content/drive/MyDrive/Assignment 3/Flowers-Dataset.zip'
```

```
inflating: flowers/tulip/8712270243_8512cf4fbd.jpg
inflating: flowers/tulip/8712270665_57b5bda0a2_n.jpg
inflating: flowers/tulip/8712282563_3819afb7bc.jpg
inflating: flowers/tulip/8713357842_9964a93473_n.jpg
inflating: flowers/tulip/8713387500_6a9138b41b_n.jpg
inflating: flowers/tulip/8713388322_e5ae26263b_n.jpg
inflating: flowers/tulip/8713389178_66bceb71a8_n.jpg
inflating: flowers/tulip/8713390684_041148dd3e_n.jpg
inflating: flowers/tulip/8713391394_4b679ea1e3_n.jpg
inflating: flowers/tulip/8713392604_90631fb809_n.jpg
inflating: flowers/tulip/8713394070_b24561b0a9.jpg
inflating: flowers/tulip/8713396140_5af8136136.jpg
inflating: flowers/tulip/8713397358_0505cc0176_n.jpg
inflating: flowers/tulip/8713397694_bcbcbba2c2_n.jpg
inflating: flowers/tulip/8713398114_bc96f1b624_n.jpg
inflating: flowers/tulip/8713398614_88202e452e_n.jpg
inflating: flowers/tulip/8713398906_28e59a225a_n.jpg
inflating: flowers/tulip/8713407768_f880df361f.jpg
inflating: flowers/tulip/8717900362_2aa508e9e5.jpg
inflating: flowers/tulip/8722514702_7ecc68691c.jpg
inflating: flowers/tulip/8723767533_9145dec4bd_n.jpg
inflating: flowers/tulip/8729501081_b993185542_m.jpg
inflating: flowers/tulip/8733586143_3139db6e9e_n.jpg
inflating: flowers/tulip/8748266132_5298a91dcf_n.jpg
inflating: flowers/tulip/8750288831_5e49a9f29b.jpg
inflating: flowers/tulip/8757486380_90952c5377.jpg
inflating: flowers/tulip/8758464923_75a5ffe320_n.jpg
inflating: flowers/tulip/8758519201_16e8d2d781_n.jpg
inflating: flowers/tulip/8759594528_2534c0ec65_n.jpg
inflating: flowers/tulip/8759597778_7fca5d434b_n.jpg
inflating: flowers/tulip/8759601388_36e2a50d98_n.jpg
inflating: flowers/tulip/8759606166_8e475013fa_n.jpg
inflating: flowers/tulip/8759618746_f5e39fdbf8_n.jpg
inflating: flowers/tulip/8762189906_8223cef62f.jpg
inflating: flowers/tulip/8762193202_0fbf2f6a81.jpg
inflating: flowers/tulip/8768645961_8f1e097170_n.jpg
inflating: flowers/tulip/8817622133_a42bb90e38_n.jpg
inflating: flowers/tulip/8838347159_746d14e6c1_m.jpg
inflating: flowers/tulip/8838354855_c474fc66a3_m.jpg
inflating: flowers/tulip/8838914676_8ef4db7f50_n.jpg
inflating: flowers/tulip/8838975946_f54194894e_m.jpg
inflating: flowers/tulip/8838983024_5c1a767878_n.jpg
```

```
  inflating: flowers/tulip/8892851067_79242a7362_n.jpg
  inflating: flowers/tulip/8904780994_8867d64155_n.jpg
  inflating: flowers/tulip/8908062479_449200a1b4.jpg
  inflating: flowers/tulip/8908097235_c3e746d36e_n.jpg
  inflating: flowers/tulip/9019694597_2d3bbedb17.jpg
  inflating: flowers/tulip/9030467406_05e93ff171_n.jpg
  inflating: flowers/tulip/9048307967_40a164a459_m.jpg
  inflating: flowers/tulip/924782410_94ed7913ca_m.jpg
  inflating: flowers/tulip/9378657435_89fabf13c9_n.jpg
  inflating: flowers/tulip/9444202147_405290415b_n.jpg
  inflating: flowers/tulip/9446982168_06c4d71da3_n.jpg
  inflating: flowers/tulip/9831362123_5aac525a99_n.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
```

```
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, verti
```

```
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.
```

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

3. Creating Model

```
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")
```

3a. Convolution layer

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

3b. Maxpooling layer

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

3c. Flatten

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

3d. Hidden/dense layers

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

3e. Output layer

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

4. Compiling Model

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

4. Fit The Model

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

```
Epoch 1/15
44/44 [==============================] - 33s 722ms/step - loss: 1.9328 - accu
Epoch 2/15
44/44 [==============================] - 30s 676ms/step - loss: 1.1435 - accu
Epoch 3/15
44/44 [==============================] - 32s 720ms/step - loss: 1.0711 - accu
Epoch 4/15
44/44 [==============================] - 30s 680ms/step - loss: 1.0149 - accu
```

```
Epoch 5/15
44/44 [==============================] - 30s 671ms/step - loss: 0.9642 - accu
Epoch 6/15
44/44 [==============================] - 30s 669ms/step - loss: 0.9278 - accu
Epoch 7/15
44/44 [==============================] - 32s 733ms/step - loss: 0.9032 - accu
Epoch 8/15
44/44 [==============================] - 30s 674ms/step - loss: 0.8751 - accu
Epoch 9/15
44/44 [==============================] - 30s 681ms/step - loss: 0.8442 - accu
Epoch 10/15
44/44 [==============================] - 31s 693ms/step - loss: 0.8211 - accu
Epoch 11/15
44/44 [==============================] - 32s 730ms/step - loss: 0.8088 - accu
Epoch 12/15
44/44 [==============================] - 31s 687ms/step - loss: 0.7778 - accu
Epoch 13/15
44/44 [==============================] - 30s 682ms/step - loss: 0.7652 - accu
Epoch 14/15
44/44 [==============================] - 30s 682ms/step - loss: 0.7466 - accu
Epoch 15/15
44/44 [==============================] - 31s 701ms/step - loss: 0.7403 - accu
<keras.callbacks.History at 0x7fe2ea323a10>
```

5. Save The Model

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

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op wh
```

6. Test The Model

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


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


sunflower_img = image.load_img('/content/flowers/sunflower/1008566138_6927679c8a.j
x = image.img_to_array(sunflower_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)
```

```
1/1 [==============================] - 0s 43ms/step
```

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

```
    'sunflowers'
sunflower_img
```

✓ 0s completed at 09:23