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

```
Mounted at /content/gdrive
```

```
# Unzip the data
# Extract data
```

```
!unzip '/content/gdrive/MyDrive/Flowers-Dataset.zip'
```

```
  inflating: flowers/sunflower/20112366233_d6cb3b6e15_n.jpg
  inflating: flowers/sunflower/20148493928_9f75a99783.jpg
  inflating: flowers/sunflower/20156280765_a6baea3176.jpg
  inflating: flowers/sunflower/20171662239_f69b6c12bd_n.jpg
  inflating: flowers/sunflower/201809908_0ef84bb351.jpg
  inflating: flowers/sunflower/20183028616_beb937e75c_m.jpg
  inflating: flowers/sunflower/20183071136_c297e74fcc_m.jpg
  inflating: flowers/sunflower/20258015499_93b9951800_m.jpg
  inflating: flowers/sunflower/20342824594_9740b7b160.jpg
  inflating: flowers/sunflower/20344282483_05abb0b837.jpg
  inflating: flowers/sunflower/20344366953_44fb51051b.jpg
  inflating: flowers/sunflower/20406385204_469f6749e2_n.jpg
  inflating: flowers/sunflower/20407896403_a50fef58ac_n.jpg
  inflating: flowers/sunflower/20410533613_56da1cce7c.jpg
  inflating: flowers/sunflower/20410697750_c43973d1eb.jpg
  inflating: flowers/sunflower/20481273479_d459834a3e_n.jpg
  inflating: flowers/sunflower/20621698991_dcb323911d.jpg
  inflating: flowers/sunflower/20658775992_1619cd0a9b_n.jpg
  inflating: flowers/sunflower/20667988875_6e73ac2879_n.jpg
  inflating: flowers/sunflower/2067882323_8de6623ffd.jpg
  inflating: flowers/sunflower/20704967595_a9c9b8d431.jpg
  inflating: flowers/sunflower/20753711039_0b11d24b50_n.jpg
  inflating: flowers/sunflower/20777358950_c63ea569a1.jpg
  inflating: flowers/sunflower/20777375650_ef854bf645.jpg
  inflating: flowers/sunflower/20812318934_82f10c45a1_n.jpg
  inflating: flowers/sunflower/20871601265_daa4be4291_n.jpg
  inflating: flowers/sunflower/20905163782_312e2c3bda_n.jpg
  inflating: flowers/sunflower/20938724084_7fe6bf87ae_n.jpg
  inflating: flowers/sunflower/20965412955_2c640b13bd.jpg
  inflating: flowers/sunflower/20972862281_5367f4af88.jpg
  inflating: flowers/sunflower/20972866151_e6a928b00a.jpg
  inflating: flowers/sunflower/210076535_80951bc5d5.jpg
  inflating: flowers/sunflower/21134000558_d7d6c9b1fe_n.jpg
  inflating: flowers/sunflower/21349789961_18ba1af5b7_n.jpg
  inflating: flowers/sunflower/21374127408_5ffbe87bb2.jpg
  inflating: flowers/sunflower/21518663809_3d69f5b995_n.jpg
  inflating: flowers/sunflower/215798352_184d8040d1.jpg
  inflating: flowers/sunflower/215798354_429de28c2d.jpg
  inflating: flowers/sunflower/215798357_3f4bfa27b7.jpg
  inflating: flowers/sunflower/21728822928_9f6817325a_n.jpg
  inflating: flowers/sunflower/21796333524_38fc8e0ab5_n.jpg
  inflating: flowers/sunflower/21821266773_7113d34c35_m.jpg
  inflating: flowers/sunflower/21899501660_7065d1c1fa_n.jpg
  inflating: flowers/sunflower/21984860006_20dfacea1c_m.jpg
  inflating: flowers/sunflower/21995435890_e5672244a4_m.jpg
  inflating: flowers/sunflower/22183521655_56221bf2a4_n.jpg
```

```
        inflating: flowers/sunflower/22183529245_ce13557515_m.jpg
        inflating: flowers/sunflower/22203670478_9ec5c2700b_n.jpg
        inflating: flowers/sunflower/22255608949_172d7c8d22_m.jpg
        inflating: flowers/sunflower/22405882322_d4561f8469_n.jpg
        inflating: flowers/sunflower/22405887122_75eda1872f_m.jpg
        inflating: flowers/sunflower/22416421196_caf131c9fa_m.jpg
        inflating: flowers/sunflower/22419079265_8902cddb7d_n.jpg
        inflating: flowers/sunflower/22429146402_332fa2fc72_m.jpg
        inflating: flowers/sunflower/22429946721_e17a12cb39_n.jpg
        inflating: flowers/sunflower/22478719251_276cb094f9_n.jpg
        inflating: flowers/sunflower/22686342422_c0b9e2f38e.jpg
        inflating: flowers/sunflower/22992257000_76dbc599e7_m.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, vertical_flip
```

```
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),
  ]
)
```

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

```
#Image Augumentation accuracy
data_augmentation = Sequential(
  [
    layers.RandomFlip("horizontal",input_shape=(img_height, img_width, 3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
```

```
  ]
)
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.
```

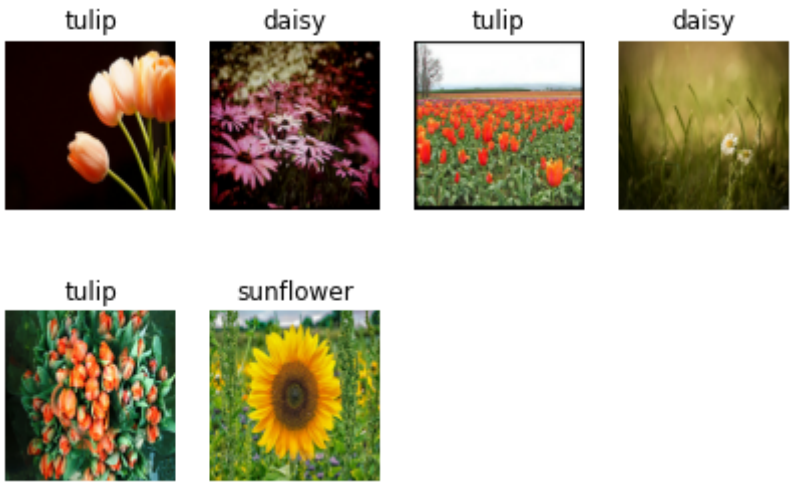
```
validation_ds = tf.keras.utils.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="validation",
seed=107,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

```
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(3, 4, i + 1)
    plt.imshow(data[i].numpy().astype("uint8"))
    plt.title(training_ds.class_names[labels[i]])
    plt.axis("off")
```




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

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


model.add(Flatten())


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


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


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


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

```
Epoch 1/15
44/44 [==============================] - 27s 599ms/step - loss: 1.4458 - accuracy: 0
Epoch 2/15
44/44 [==============================] - 27s 604ms/step - loss: 1.1130 - accuracy: 0
Epoch 3/15
44/44 [==============================] - 26s 580ms/step - loss: 1.0361 - accuracy: 0
Epoch 4/15
44/44 [==============================] - 26s 581ms/step - loss: 0.9897 - accuracy: 0
Epoch 5/15
44/44 [==============================] - 26s 581ms/step - loss: 0.9449 - accuracy: 0
Epoch 6/15
44/44 [==============================] - 26s 578ms/step - loss: 0.9060 - accuracy: 0
Epoch 7/15
44/44 [==============================] - 26s 581ms/step - loss: 0.8853 - accuracy: 0
Epoch 8/15
44/44 [==============================] - 26s 581ms/step - loss: 0.8512 - accuracy: 0
Epoch 9/15
44/44 [==============================] - 27s 590ms/step - loss: 0.8720 - accuracy: 0
Epoch 10/15
44/44 [==============================] - 26s 605ms/step - loss: 0.8267 - accuracy: 0
Epoch 11/15
44/44 [==============================] - 26s 581ms/step - loss: 0.7844 - accuracy: 0
Epoch 12/15
44/44 [==============================] - 26s 580ms/step - loss: 0.8154 - accuracy: 0
Epoch 13/15
44/44 [==============================] - 26s 577ms/step - loss: 0.7587 - accuracy: 0
Epoch 14/15
44/44 [==============================] - 26s 579ms/step - loss: 0.7552 - accuracy: 0
Epoch 15/15
44/44 [==============================] - 26s 575ms/step - loss: 0.7396 - accuracy: 0
<keras.callbacks.History at 0x7efc8d39f250>
```

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


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


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

```
tulip_img = image.load_img('/content/flowers/tulip/112428919_f0c5ad7d9d_n.jpg',target_size
x = image.img_to_array(tulip_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)
```

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

```
'tulips'
```

```
tulip_img
```