

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

➞ Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force\_remount=True).

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

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

```
inflating: flowers/tulip/14090534565_5857ce4b7c_n.jpg
inflating: flowers/tulip/14090546015_504c8becd1.jpg
inflating: flowers/tulip/14093565032_a8f1e349d1.jpg
inflating: flowers/tulip/14093884601_c87b5cd663_n.jpg
inflating: flowers/tulip/14093907931_dd8f642574.jpg
inflating: flowers/tulip/14094114202_4c1d7f1116.jpg
inflating: flowers/tulip/14094146241_0b9fa7b3e0.jpg
inflating: flowers/tulip/14097111174_87a2e7e0c7_n.jpg
inflating: flowers/tulip/14097328354_4f1469a170.jpg
inflating: flowers/tulip/14097366955_84ef6369f2.jpg
inflating: flowers/tulip/14097676864_4ca8e8b20d_n.jpg
inflating: flowers/tulip/14097745904_436c4ba1b4_n.jpg
inflating: flowers/tulip/14099204939_60e6ffa4c3_n.jpg
inflating: flowers/tulip/14103897845_7986002615.jpg
inflating: flowers/tulip/14110615113_bd7b3fcb84.jpg
inflating: flowers/tulip/14110616533_e04775e7b1.jpg
inflating: flowers/tulip/14116780333_7836f4448c.jpg
inflating: flowers/tulip/14116826873_d4bab623bf_n.jpg
inflating: flowers/tulip/14122029097_3e3285ca5c_n.jpg
inflating: flowers/tulip/14124669683_7fb74f20c3.jpg
inflating: flowers/tulip/14127532150_112823a8f6.jpg
inflating: flowers/tulip/14149603605_eedfe9678c_n.jpg
inflating: flowers/tulip/14171673854_1208c19be3_m.jpg
inflating: flowers/tulip/142218310_d06005030a_n.jpg
inflating: flowers/tulip/142235017_07816937c6.jpg
inflating: flowers/tulip/142235237_da662d925c.jpg
inflating: flowers/tulip/142235914_5419ff8a4a.jpg
inflating: flowers/tulip/14233348496_134e274d0b_n.jpg
inflating: flowers/tulip/14233348826_89949b4784_n.jpg
inflating: flowers/tulip/14235021006_dd001ea8ed_n.jpg
inflating: flowers/tulip/14238880561_7181ccb9fb_n.jpg
inflating: flowers/tulip/14238880681_2b86165b41_n.jpg
inflating: flowers/tulip/14238880951_2f62f042be_n.jpg
inflating: flowers/tulip/14239946452_b0c8f018fe_n.jpg
inflating: flowers/tulip/14241883744_ff69de8b7f_n.jpg
inflating: flowers/tulip/14253178581_4dfcedb4d3_n.jpg
inflating: flowers/tulip/14253178841_8a637b64f0_n.jpg
inflating: flowers/tulip/14254329062_140c0f4f4f_n.jpg
inflating: flowers/tulip/14254329512_8b4229b898_n.jpg
inflating: flowers/tulip/14254393301_ffb19c6445_n.jpg
inflating: flowers/tulip/14255917256_84c23c572b.jpg
inflating: flowers/tulip/14256111494_f9985585f5_n.jpg
inflating: flowers/tulip/14256112284_5f9d6d1aae_n.jpg
inflating: flowers/tulip/14256494455_0718057a48_n.jpg
inflating: flowers/tulip/14256494695_741a989ebb_n.jpg
inflating: flowers/tulip/14256494935_0ee69cfa03_n.jpg
inflating: flowers/tulip/14262354955_cc2ab3b112_m.jpg
inflating: flowers/tulip/14266093711_66d18a1e44_n.jpg
inflating: flowers/tulip/14270573963_f122c40438.jpg
inflating: flowers/tulip/14275234071_6e6f473356.jpg
inflating: flowers/tulip/14276668513_f205b5270d_n.jpg
inflating: flowers/tulip/14278331403_4c475f9a9b.jpg
inflating: flowers/tulip/14487705209_ea723109e1_m.jpg
inflating: flowers/tulip/14487712670_aebe715525_m.jpg
inflating: flowers/tulip/14487762578_baba13d16a_m.jpg
inflating: flowers/tulip/14487943607_651e8062a1_m.jpg
inflating: flowers/tulip/14491997336_36ba524713.jpg
inflating: flowers/tulip/14651383746_419dc73634_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 = 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.
```

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

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

```
# Add Layers
```

```
model=Sequential()
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(4,activation='softmax'))
```

```
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 [=====] - 28s 605ms/step - loss: 1.6109 - accuracy: 0.3838
Epoch 2/15
44/44 [=====] - 27s 618ms/step - loss: 1.1662 - accuracy: 0.5031
Epoch 3/15
44/44 [=====] - 28s 636ms/step - loss: 1.1181 - accuracy: 0.5335
Epoch 4/15
44/44 [=====] - 27s 600ms/step - loss: 1.0624 - accuracy: 0.5729
Epoch 5/15
44/44 [=====] - 27s 600ms/step - loss: 1.0085 - accuracy: 0.6011
Epoch 6/15
44/44 [=====] - 26s 597ms/step - loss: 0.9588 - accuracy: 0.6308
Epoch 7/15
44/44 [=====] - 27s 600ms/step - loss: 0.9475 - accuracy: 0.6240
Epoch 8/15
44/44 [=====] - 27s 597ms/step - loss: 0.9255 - accuracy: 0.6442
Epoch 9/15
44/44 [=====] - 27s 599ms/step - loss: 0.8724 - accuracy: 0.6620
Epoch 10/15
44/44 [=====] - 27s 599ms/step - loss: 0.8676 - accuracy: 0.6586
Epoch 11/15
44/44 [=====] - 27s 602ms/step - loss: 0.8468 - accuracy: 0.6745
Epoch 12/15
44/44 [=====] - 27s 600ms/step - loss: 0.8417 - accuracy: 0.6708
Epoch 13/15
44/44 [=====] - 27s 604ms/step - loss: 0.7904 - accuracy: 0.6982
Epoch 14/15
44/44 [=====] - 27s 596ms/step - loss: 0.8153 - accuracy: 0.6908
Epoch 15/15
44/44 [=====] - 26s 610ms/step - loss: 0.7668 - accuracy: 0.7063
<keras.callbacks.History at 0x7f329e5b9d10>
```

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

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

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

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

rose_img = image.load_img('/content/flowers/rose/10090824183_d02c613f10_m.jpg',target_size=(64,64))
x = image.img_to_array(rose_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'
```

rose\_img



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 12:01 PM

