```
import zipfile
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).
```

```
!unzip '/content/drive/MyDrive/Flowers-Dataset/flowers'
```

```
unzip:  cannot find or open
/content/drive/MyDrive/Flowers-Dataset/flowers,
/content/drive/MyDrive/Flowers-Dataset/flowers.zip or
/content/drive/MyDrive/Flowers-Dataset/flowers.ZIP.
```

```
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/drive/MyDrive/Flowers-Dataset/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/drive/MyDrive/Flowers-
Dataset/flowers',
                                                target_size=(64,64),

class_mode='categorical',
                                                batch_size=100)
```

```
Found 4327 images belonging to 5 classes.
```

```
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 4327 files belonging to 5 classes.
Using 3462 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 4327 files belonging to 5 classes.
Using 865 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")
```

```
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 = 25, steps_per_epoch = len(x_train))
```

```
Epoch 1/25
44/44 [==============================] - 525s 12s/step - loss: 1.4070
- accuracy: 0.3973
Epoch 2/25
44/44 [==============================] - 23s 521ms/step - loss: 1.0944
- accuracy: 0.5567
```

```
Epoch 3/25
44/44 [==============================] - 24s 532ms/step - loss: 1.0323
- accuracy: 0.5838
Epoch 4/25
44/44 [==============================] - 23s 526ms/step - loss: 0.9654
- accuracy: 0.6270
Epoch 5/25
44/44 [==============================] - 23s 528ms/step - loss: 0.9591
- accuracy: 0.6254
Epoch 6/25
44/44 [==============================] - 24s 536ms/step - loss: 0.8865
- accuracy: 0.6610
Epoch 7/25
44/44 [==============================] - 23s 528ms/step - loss: 0.8594
- accuracy: 0.6630
Epoch 8/25
44/44 [==============================] - 23s 528ms/step - loss: 0.8351
- accuracy: 0.6767
Epoch 9/25
44/44 [==============================] - 23s 522ms/step - loss: 0.8187
- accuracy: 0.6832
Epoch 10/25
44/44 [==============================] - 24s 534ms/step - loss: 0.7780
- accuracy: 0.6954
Epoch 11/25
44/44 [==============================] - 23s 521ms/step - loss: 0.7471
- accuracy: 0.7162
Epoch 12/25
44/44 [==============================] - 23s 525ms/step - loss: 0.7351
- accuracy: 0.7169
Epoch 13/25
44/44 [==============================] - 23s 528ms/step - loss: 0.7056
- accuracy: 0.7331
Epoch 14/25
44/44 [==============================] - 23s 524ms/step - loss: 0.6842
- accuracy: 0.7386
Epoch 15/25
44/44 [==============================] - 23s 525ms/step - loss: 0.6750
- accuracy: 0.7458
Epoch 16/25
44/44 [==============================] - 24s 533ms/step - loss: 0.6516
- accuracy: 0.7492
Epoch 17/25
44/44 [==============================] - 23s 526ms/step - loss: 0.6041
- accuracy: 0.7696
Epoch 18/25
44/44 [==============================] - 24s 531ms/step - loss: 0.6121
- accuracy: 0.7694
Epoch 19/25
44/44 [==============================] - 23s 528ms/step - loss: 0.5921
```

```
- accuracy: 0.7781
Epoch 20/25
44/44 [==============================] - 24s 534ms/step - loss: 0.5720
- accuracy: 0.7839
Epoch 21/25
44/44 [==============================] - 24s 533ms/step - loss: 0.5280
- accuracy: 0.8019
Epoch 22/25
44/44 [==============================] - 24s 534ms/step - loss: 0.5380
- accuracy: 0.7925
Epoch 23/25
44/44 [==============================] - 24s 537ms/step - loss: 0.5194
- accuracy: 0.8070
Epoch 24/25
44/44 [==============================] - 23s 530ms/step - loss: 0.5066
- accuracy: 0.8093
Epoch 25/25
44/44 [==============================] - 23s 524ms/step - loss: 0.5005
- accuracy: 0.8110

<keras.callbacks.History at 0x7fb3c98786d0>
```

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

```
WARNING:absl:Found untraced functions such as
_jit_compiled_convolution_op while saving (showing 1 of 1). These
functions will not be directly callable after loading.
```

```
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/drive/MyDrive/Flowers-Dataset/flowers/rose/
11233672494_d8bf0a3dbf_n.jpg',target_size=(64,64))
x = image.img_to_array(rose_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)
```

```
1/1 [==============================] - 0s 16ms/step
```

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

{"type":"string"}

```
rose_img
```