

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.

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

```
inflating: flowers/tulip/8695372372_302135aeb2.jpg
inflating: flowers/tulip/8697784345_e75913d220.jpg
inflating: flowers/tulip/8702982836_75222725d7.jpg
inflating: flowers/tulip/8706523526_a0f161b72b.jpg
inflating: flowers/tulip/8708209606_d3aede4801.jpg
inflating: flowers/tulip/8708856019_f3be2353a4_n.jpg
inflating: flowers/tulip/8710148289_6fc196a0f8_n.jpg
inflating: flowers/tulip/8711277462_b43df5454b_m.jpg
inflating: flowers/tulip/8712230357_1298b8513b.jpg
inflating: flowers/tulip/8712243901_54d686319e_m.jpg
inflating: flowers/tulip/8712244311_da8e90bf8e_n.jpg
inflating: flowers/tulip/8712260079_c0ff42e0e2_n.jpg
inflating: flowers/tulip/8712263493_3db76c5f82.jpg
inflating: flowers/tulip/8712266605_3787e346cd_n.jpg
inflating: flowers/tulip/8712267391_c756f18ee7_n.jpg
inflating: flowers/tulip/8712267813_f7a9be2ec5.jpg
inflating: flowers/tulip/8712268519_f4c2c39a06_n.jpg
inflating: flowers/tulip/8712269349_2b933da2b8_n.jpg
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

```

Image augmentation

```

import tensorflow as tensorflow
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data_aug = ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
train_data_aug = ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
test_data_aug = ImageDataGenerator(rescale=1./255)
xtrain = train_data_aug.flow_from_directory('/content/flowers', target_size=(64,64), class_mode='categorical')

```

Found 4317 images belonging to 5 classes.

```

xtest = test_data_aug.flow_from_directory('/content/flowers', target_size=(64,64), class_mode='categorical')

```

Found 4317 images belonging to 5 classes.

Create Model

```

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

```

Add Layers (Convolution, Max-Pooling, Flatten, Dense-(Hidden Layers), Output)

```

model = Sequential()
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(64,64,3))) # Convolution 1
model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
# Fully connected layers (ANN)
model.add(Dense(300, activation='relu')) # Hidden layer 1
model.add(Dense(150, activation='relu')) # Hidden layer 2
model.add(Dense(5, activation='softmax')) # Output layer

```

Compile the Model

```

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

```

Fit the Model

```
model.fit_generator(xtrain,
steps_per_epoch=len(xtrain),
epochs=10,
validation_data=xtest,
validation_steps=len(xtest))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: `Model.
"""
Epoch 1/10
44/44 [=====] - 46s 1s/step - loss: 1.4892 - accuracy: 0.38
Epoch 2/10
44/44 [=====] - 44s 1s/step - loss: 1.1191 - accuracy: 0.55
Epoch 3/10
44/44 [=====] - 48s 1s/step - loss: 1.0183 - accuracy: 0.60
Epoch 4/10
44/44 [=====] - 44s 999ms/step - loss: 0.9719 - accuracy: 0
Epoch 5/10
44/44 [=====] - 43s 972ms/step - loss: 0.8939 - accuracy: 0
Epoch 6/10
44/44 [=====] - 44s 1s/step - loss: 0.8514 - accuracy: 0.67
Epoch 7/10
44/44 [=====] - 45s 1s/step - loss: 0.8057 - accuracy: 0.69
Epoch 8/10
44/44 [=====] - 44s 1s/step - loss: 0.7917 - accuracy: 0.69
Epoch 9/10
44/44 [=====] - 44s 1s/step - loss: 0.7449 - accuracy: 0.72
Epoch 10/10
44/44 [=====] - 44s 988ms/step - loss: 0.7185 - accuracy: 0
<keras.callbacks.History at 0x7f72ec45b6d0>
```

Save the Model

```
model.save('/content/flowers')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op while sav
```

Test the Model

```
import numpy as np
from tensorflow.keras.preprocessing import image
img = image.load_img('/content/flowers/rose/10503217854_e66a804309.jpg',target_size=(64,64))
```

```
img
```



```
x = image.img_to_array(img)
```

```
x
```

```
array([[ [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        ...,
        [ 92., 14.,  0.],
        [ 61., 13.,  9.],
        [ 17.,  7.,  5.]],

       [[ [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        ...,
        [150.,  3.,  0.],
        [ 85., 10.,  7.],
        [119.,  4.,  1.]],

       [[ [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        [ 0.,  2.,  0.],
        ...,
        [ 88.,  9.,  0.],
        [207.,  7., 10.],
        [152.,  0.,  0.]],

       ...,

       [[ [ 0.,  4.,  0.],
        [ 1.,  3.,  0.],
        [ 0.,  2.,  0.],
        ...,
        [ 2.,  2.,  4.],
        [ 0.,  2.,  5.],
        [ 51., 10.,  6.]],

       [[ [ 0.,  2.,  0.],
        [ 1.,  3.,  0.],
        [ 1.,  3.,  0.],
        ...,
        [ 0.,  3.,  1.],
        [ 0.,  3.,  4.],
        [ 0.,  5.,  3.]],

       [[ [ 1.,  3.,  0.],
        [ 0.,  2.,  0.],
        [ 1.,  1.,  0.],
        ...,
        [ 29.,  5.,  1.],
        [ 41., 13.,  0.],
        [ 5.,  4.,  0.] ]], dtype=float32)
```

```
x = np.expand_dims(x,axis=0)
```

```
x
```

```
array([[[[ 0.,  2.,  0.],
         [ 0.,  2.,  0.],
         [ 0.,  2.,  0.],
```

```

...,
[ 92., 14., 0.],
[ 61., 13., 9.],
[ 17., 7., 5.]],

[[ 0., 2., 0.],
[ 0., 2., 0.],
[ 0., 2., 0.],
...,
[150., 3., 0.],
[ 85., 10., 7.],
[119., 4., 1.]],

[[ 0., 2., 0.],
[ 0., 2., 0.],
[ 0., 2., 0.],
...,
[ 88., 9., 0.],
[207., 7., 10.],
[152., 0., 0.]],

...,

[[ 0., 4., 0.],
[ 1., 3., 0.],
[ 0., 2., 0.],
...,
[ 2., 2., 4.],
[ 0., 2., 5.],
[ 51., 10., 6.]],

[[ 0., 2., 0.],
[ 1., 3., 0.],
[ 1., 3., 0.],
...,
[ 0., 3., 1.],
[ 0., 3., 4.],
[ 0., 5., 3.]],

[[ 1., 3., 0.],
[ 0., 2., 0.],
[ 1., 1., 0.],
...,
[ 29., 5., 1.],
[ 41., 13., 0.],
[ 5., 4., 0.]]], dtype=float32)

```

```
model.predict(x)
```

```

1/1 [=====] - 0s 130ms/step
array([[0., 0., 0., 0., 1.]], dtype=float32)

```

```
xtrain.class_indices
```

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

```
op = ['daisy','dandelion','rose','sunflower','tulip']  
pred = np.argmax(model.predict(x))  
op[pred]
```

```
1/1 [=====] - 0s 26ms/step  
'tulip'
```

```
img = image.load_img('/content/flowers/dandelion/10043234166_e6dd915111_n.jpg',target_size  
x = image.img_to_array(img)  
x = np.expand_dims(x,axis=0)  
pred = np.argmax(model.predict(x))  
op[pred]
```

```
1/1 [=====] - 0s 26ms/step  
'sunflower'
```

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

✓ 0s completed at 3:02 PM

