

▼ Download the Dataset

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

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



Unzip the Dataset

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



Archive: /content/Flowers-Dataset.zip

```

  inflating: flowers/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
  inflating: flowers/daisy/10172379554_b296050f82_n.jpg
  inflating: flowers/daisy/10172567486_2748826a8b.jpg
  inflating: flowers/daisy/10172636503_21bededa75_n.jpg
  inflating: flowers/daisy/102841525_bd6628ae3c.jpg
  inflating: flowers/daisy/103007222094_28fa978807_n.jpg
  inflating: flowers/daisy/1031799732_e7f4008c03.jpg
  inflating: flowers/daisy/10391248763_1d16681106_n.jpg
  inflating: flowers/daisy/10437754174_22ec990b77_m.jpg
  inflating: flowers/daisy/10437770546_8bb6f7bdd3_m.jpg
  inflating: flowers/daisy/10437929963_bc13eebe0c.jpg
  inflating: flowers/daisy/10466290366_cc72e33532.jpg
  inflating: flowers/daisy/10466558316_a7198b87e2.jpg
  inflating: flowers/daisy/10555749515_13a12a026e.jpg
  inflating: flowers/daisy/10555815624_dc211569b0.jpg
  inflating: flowers/daisy/10555826524_423eb8bf71_n.jpg
  inflating: flowers/daisy/10559679065_50d2b16f6d.jpg
  inflating: flowers/daisy/105806915_a9c13e2106_n.jpg
  inflating: flowers/daisy/10712722853_5632165b04.jpg
  inflating: flowers/daisy/107592979_aaa9cdf7e78_m.jpg
  inflating: flowers/daisy/10770585085_4742b9dac3_n.jpg
  inflating: flowers/daisy/10841136265_af473efc60.jpg
  inflating: flowers/daisy/10993710036_2033222c91.jpg
  inflating: flowers/daisy/10993818044_4c19b86c82.jpg
  inflating: flowers/daisy/10994032453_ac7f8d9e2e.jpg
  inflating: flowers/daisy/11023214096_b5b39fab08.jpg
  inflating: flowers/daisy/11023272144_fce94401f2_m.jpg
  inflating: flowers/daisy/11023277956_8980d53169_m.jpg
  inflating: flowers/daisy/11124324295_503f3a0804.jpg
  inflating: flowers/daisy/1140299375_3aa7024466.jpg
  inflating: flowers/daisy/11439894966_dca877f0cd.jpg
  inflating: flowers/daisy/1150395827_6f94a5c6e4_n.jpg
  inflating: flowers/daisy/11642632_1e7627a2cc.jpg
  inflating: flowers/daisy/11834945233_a53b7a92ac_m.jpg
  inflating: flowers/daisy/11870378973_2ec1919f12.jpg
  inflating: flowers/daisy/11891885265_ccfefec7284_n.jpg
  inflating: flowers/daisy/12193032636_b50ae7db35_n.jpg
  inflating: flowers/daisy/12348343085_d4c396e5b5_m.jpg
  inflating: flowers/daisy/12585131704_0f64b17059_m.jpg
```

```

inflating: flowers/daisy/12601254324_3cb62c254a_m.jpg
inflating: flowers/daisy/1265350143_6e2b276ec9.jpg
inflating: flowers/daisy/12701063955_4840594ea6_n.jpg
inflating: flowers/daisy/1285423653_18926dc2c8_n.jpg
inflating: flowers/daisy/1286274236_1d7ac84efb_n.jpg
inflating: flowers/daisy/12891819633_e4c82b51e8.jpg
inflating: flowers/daisy/1299501272_59d9da5510_n.jpg
inflating: flowers/daisy/1306119996_ab8ae14d72_n.jpg
inflating: flowers/daisy/1314069875_da8dc023c6_m.jpg
inflating: flowers/daisy/1342002397_9503c97b49.jpg
inflating: flowers/daisy/134409839_71069a95d1_m.jpg
inflating: flowers/daisy/1344985627_c3115e2d71_n.jpg
inflating: flowers/daisy/13491959645_2cd9df44d6_n.jpg
inflating: flowers/daisy/1354396826_2868631432_m.jpg
inflating: flowers/daisy/1355787476_32e9f2a30b.jpg
inflating: flowers/daisy/13583238844_573df2de8e_m.jpg
inflating: flowers/daisy/1374193928_a52320eafa.jpg

```

▼ Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen=ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True, vertic
```

```
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train=train_datagen.flow_from_directory(r"/content/flowers", target_size=(64,64), class_mo
```

```
    Found 8634 images belonging to 6 classes.
```

```
x_test=test_datagen.flow_from_directory(r"/content/flowers", target_size=(64,64), class_mode
```

```
    Found 8634 images belonging to 6 classes.
```

```
x_train.class_indices
```

```

{'daisy': 0,
 'dandelion': 1,
 'flowers': 2,
 'rose': 3,
 'sunflower': 4,
 'tulip': 5}

```

▼ Create Model

```
from tensorflow.keras.models import Sequential
```

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

```
model=Sequential()
```

▼ Add Layers

Convolution

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

MaxPooling

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

Flatten

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

```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
flatten (Flatten)	(None, 28800)	0
Total params: 10,144		
Trainable params: 10,144		
Non-trainable params: 0		

```
32*(3*3*3+1)
```

```
896
```

Hidden Layers

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

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

Output

```
model.add(Dense(4,activation='softmax'))
```

▼ Compile The Model

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

```
len(x_train)
```

```
360
```

```
8634/24
```

```
359.75
```

▼ Fit The Model

```
model.fit
```

```
<bound method Model.fit of <keras.engine.sequential.Sequential object at 0x7f1116f22250>>
```

▼ Save The Model

```
ls
```

```
daisy/ dandelion/ flowers/ rose/ sunflower/ tulip/
```

```
model.save('flowers.h5')
```

```
ls
```

```
daisy/ dandelion/ flowers/ flowers.h5 rose/ sunflower/ tulip/
```

▼ Test The Model

```
import numpy as np
```

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

```
model=load_model('flowers.h5')
```

```
img=image.load_img(r"/content/flowers/flowers/rose/10090824183_d02c613f10_m.jpg")
```

```
img
```



```
img=image.load_img(r"/content/flowers/flowers/rose/10090824183_d02c613f10_m.jpg",target_si
img
```



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

```
x
```

```
array([[[120., 120., 108.],
        [ 97.,  87.,  88.],
        [109.,  99.,  98.],
        ...,
        [124., 136., 114.],
        [163., 172., 153.],
        [138., 143., 121.]],

       [[ 95.,  88.,  96.],
        [ 68.,  48.,  73.],
        [ 89.,  75.,  90.],
        ...,
        [132., 146., 133.],
        [112., 121., 104.],
        [113., 125., 105.]],

       [[ 89.,  76.,  94.],
        [ 83.,  70.,  87.],
        [ 83.,  77.,  87.],
        ...,
```

```

[105., 104., 99.],
[112., 121., 104.],
[105., 111., 97.]],

...,

[[ 22.,  7., 38.],
 [ 23.,  8., 39.],
 [ 22.,  3., 31.],
 ...,
 [ 39., 16., 70.],
 [ 46., 21., 79.],
 [ 47., 21., 82.]],

[[ 26., 11., 40.],
 [ 23.,  8., 37.],
 [ 24., 15., 44.],
 ...,
 [ 38., 17., 73.],
 [ 42., 18., 76.],
 [ 45., 19., 80.]],

[[ 24.,  8., 37.],
 [ 26., 10., 39.],
 [ 26.,  2., 34.],
 ...,
 [ 48., 19., 77.],
 [ 52., 22., 82.],
 [ 52., 24., 85.]]], dtype=float32)

```

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

```
x
```

```

array([[[[120., 120., 108.],
 [ 97.,  87.,  88.],
 [109.,  99.,  98.],
 ...,
 [124., 136., 114.],
 [163., 172., 153.],
 [138., 143., 121.]]],

[[ 95.,  88.,  96.],
 [ 68.,  48.,  73.],
 [ 89.,  75.,  90.],
 ...,
 [132., 146., 133.],
 [112., 121., 104.],
 [113., 125., 105.]],

[[ 89.,  76.,  94.],
 [ 83.,  70.,  87.],
 [ 83.,  77.,  87.],
 ...,
 [105., 104.,  99.],
 [112., 121., 104.],
 [105., 111.,  97.]],

...,

```

```

[[ 22.,  7., 38.],
 [ 23.,  8., 39.],
 [ 22.,  3., 31.],
 ...,
 [ 39., 16., 70.],
 [ 46., 21., 79.],
 [ 47., 21., 82.]],

[[ 26., 11., 40.],
 [ 23.,  8., 37.],
 [ 24., 15., 44.],
 ...,
 [ 38., 17., 73.],
 [ 42., 18., 76.],
 [ 45., 19., 80.]],

[[ 24.,  8., 37.],
 [ 26., 10., 39.],
 [ 26.,  2., 34.],
 ...,
 [ 48., 19., 77.],
 [ 52., 22., 82.],
 [ 52., 24., 85.]]], dtype=float32)

```

```
y=np.argmax(model.predict(x),axis=1)
```

y

```

array([[41, 41, 41],
       [43, 43, 41],
       [44, 44, 44],
       [39, 39, 48],
       [34, 49, 49],
       [42, 54, 54],
       [41, 60, 55],
       [54, 54, 53],
       [39, 39, 39],
       [60, 60, 60],
       [60, 61, 61],
       [41, 41, 41],
       [30, 30, 30],
       [30, 30, 33],
       [31, 32, 32],
       [55, 32, 55],
       [33, 32, 33],
       [32,  5, 31],
       [32, 31, 31],
       [31, 13, 15],
       [20, 20, 20],
       [30, 30, 18],
       [28, 10, 12],
       [27, 10, 10],
       [40, 40, 40],
       [45, 45, 16],
       [28, 28, 10],
       [28, 28, 26],
       [28, 28, 27],
       [28, 28, 28],

```

```
[63, 61, 61],  
[56, 56, 60],  
[61, 61, 61],  
[61, 59, 61],  
[57, 57, 57],  
[57, 57, 57],  
[55, 55, 52],  
[54, 54, 59],  
[54, 54, 54],  
[54, 54, 54],  
[54, 54, 54],  
[19, 20, 20],  
[31, 31, 31],  
[16, 16, 16],  
[32, 17, 17],  
[57, 15, 15],  
[56, 10, 10],  
[54, 54, 54],  
[53, 56, 58],  
[50, 55, 59],  
[43, 43, 51],  
[53, 57, 38],  
[51, 44, 44],  
[38, 48, 56],  
[37, 57, 57],  
[60, 56, 43],  
[44, 57, 60],  
[41, 41, 57],
```

```
x_train.class_indices
```

```
{'daisy': 0,  
 'dandelion': 1,  
 'flowers': 2,  
 'rose': 3,  
 'sunflower': 4,  
 'tulip': 5}
```

```
index=['daisy','dandelion','flowers','rose','sunflower','tulip']
```

```
img=image.load_img(r"/content/flowers/flowers/daisy/10172379554_b296050f82_n.jpg",target_s  
img
```



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

