

```
!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/10300722094_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_aaa9cdfe78_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_ccefec7284_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/1274102020_252220a3fa.jpg

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True,vertical_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=24)
xtest = test_datagen.flow_from_directory('/content/flowers',
                                         target_size=(64,64),
                                         class_mode='categorical',
                                         batch_size=124)
```

Found 4317 images belonging to 5 classes.
 Found 4317 images belonging to 5 classes.

```
from tensorflow.keras.models import Sequential

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

model = Sequential()

model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # Convolutional layer

model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer

model.add(Flatten()) # Flatten layer

model.add(Dense(300,activation='relu')) # Hidden layer 1

model.add(Dense(150,activation='relu')) # Hidden layer 2

model.add(Dense(4,activation='softmax')) # Output layer

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

len(xtrain)

180

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

```
model.fit(xtrain,  
          steps_per_epoch=len(xtrain),  
          epochs=5,  
          validation_data=xtest,  
          validation_steps=len(xtest))
```

Epoch 1/5

InvalidArgumentError Traceback (most recent call last)
 <ipython_input_61_9a4ef8e8eb25> in <module>

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

```
> 5 validation_steps=len(ytest))
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
def get_model(op_name, num_outputs, input_shape, weights, conv_name,
```

```
model=load_model("/content/daisy.h5")
```

```
op_name,
```

```
img = image.load_img('/content/flowers/daisy/100080576_f52e8ee070_n.jpg',target_size
```

```
0,0) if name is not None:
```

```
img
```



```
exec(code, run_globals)
```

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

```
app.launch_new_instance()
```

```
x
```

```
array([[141., 141., 139.],
       [149., 149., 149.],
       [152., 152., 154.],
       ...,
       [162., 161., 166.],
       [154., 154., 152.],
       [153., 153., 153.]],

       [[136., 135., 131.],
       [146., 145., 143.],
       [169., 168., 174.],
       ...,
       [159., 158., 163.],
       [155., 155., 153.],
       [149., 149., 149.]],

       [[125., 125., 117.],
       [138., 140., 137.],
       [152., 152., 152.],
       ...,
       [156., 156., 156.],
       [157., 157., 155.],
       [143., 142., 140.]],

       ...,
```

```

[[ 41.,  44.,  23.],
 [ 43.,  46.,  25.],
 [ 49.,  51.,  37.],
 ...,
 [128., 124., 121.],
 [125., 121., 118.],
 [125., 122., 117.]],

[[ 43.,  46.,  25.],
 [ 43.,  46.,  25.],
 [ 54.,  55.,  37.],
 ...,
 [130., 126., 125.],
 [129., 125., 124.],
 [127., 123., 122.]],

[[ 44.,  47.,  26.],
 [ 45.,  48.,  27.],
 [ 53.,  55.,  34.],
 ...,
 [137., 133., 132.],
 [133., 129., 128.],
 [130., 126., 125.]]], dtype=float32)

```

```
x.ndim
```

```
3
```

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

```
x.ndim
```

```
4
```

```
pred = model.predict(x)
```

```
pred
```

```
array([[1.0000000e+00, 3.0940360e-19, 2.0464132e-20, 1.8283872e-10]],
      dtype=float32)
```

```
labels=["daisy","dandelion","rose","sunflower","tulip"]
```

```
np.argmax(pred)
```

```
0
```

```
labels[4]
```

```
'tulip'  
labels[np.argmax(pred)]  
  
'daisy'
```

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

