

▼ 1.DOWNLOAD THE DATA SET AND UNZIP IT

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

Mounted at /content/drive

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



Archive: /content/drive/MyDrive/IBM/Assignment/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_aaa9cdfef78_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/1374102028_552220a2fa.jpg

```

▼ 2.IMAGE AUGMENTATION

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

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_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=100)
```

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

▼ 3.CREAT MODEL

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

▼ 4.ADD LAYERS

```

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

```

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

▼ 5.COMPILE THE MODEL

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

▼ 6.FIT THE MODEL

```
model.fit(xtrain,  
          steps_per_epoch=len(xtrain),  
          epochs=10,  
          )
```

```
Epoch 1/10  
44/44 [=====] - 23s 279ms/step - loss: 1.5390 - accuracy: 0  
Epoch 2/10  
44/44 [=====] - 13s 292ms/step - loss: 1.1185 - accuracy: 0  
Epoch 3/10  
44/44 [=====] - 12s 282ms/step - loss: 1.0539 - accuracy: 0  
Epoch 4/10  
44/44 [=====] - 13s 290ms/step - loss: 0.9824 - accuracy: 0  
Epoch 5/10  
44/44 [=====] - 12s 281ms/step - loss: 0.9226 - accuracy: 0  
Epoch 6/10  
44/44 [=====] - 12s 281ms/step - loss: 0.9125 - accuracy: 0  
Epoch 7/10  
44/44 [=====] - 13s 286ms/step - loss: 0.8573 - accuracy: 0  
Epoch 8/10  
44/44 [=====] - 14s 311ms/step - loss: 0.8110 - accuracy: 0  
Epoch 9/10  
44/44 [=====] - 13s 289ms/step - loss: 0.7749 - accuracy: 0  
Epoch 10/10  
44/44 [=====] - 12s 280ms/step - loss: 0.7356 - accuracy: 0  
<keras.callbacks.History at 0x7f7d0849dad0>
```



▼ 7.SAVE THE MODEL

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

▼ 8.TESTING THE MODEL

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

```
img = image.load_img('/content/flowers/sunflower/1008566138_6927679c8a.jpg',target_size=(64,64))
img
```



```
x = image.img_to_array(img)
x = x.reshape((1,)+x.shape[1:])
x = np.expand_dims(x,axis=0)
model.predict(x)

array([[0., 0., 0., 1., 0.]], 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]
```

```
'sunflower'
```

▼ Testing

Tulip Testing

```
from numpy.lib.type_check import imag
img = image.load_img('/content/flowers/tulip/10128546863_8de70c610d.jpg',target_size=(64,64))
img
```



```
x = image.img_to_array(img)
x = x.reshape((1,)+x.shape[1:])
pred = np.argmax(model.predict(x))
op[pred]
```

```
'tulip'
```

Rose Testing

```
img = image.load_img('/content/flowers/rose/102501987_3cdb8e5394_n.jpg',target_size=(64,64)
img
```



```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op[pred]
```

'rose'

Daisy Testing

```
img = image.load_img('/content/flowers/daisy/10172567486_2748826a8b.jpg',target_size=(64,64)
img
```



```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op[pred]
```

'tulip'

Dandelion Testing

```
img = image.load_img('/content/flowers/dandelion/11405573_24a8a838cc_n.jpg',target_size=(64,64)
img
```



```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op[pred]
```

'dandelion'