

# 1) DOWNLOAD DATA SET AND UNZIP

```
!unzip '/content/Flowers-Datasets.zip'
```

**Streaming output truncated to the last 5000 lines.**

```
inflating: flowers/Testing/daisy/5883162120_dc7274af76_n.jpg
inflating: flowers/Testing/daisy/5884807222_22f5326ba8_m.jpg
inflating: flowers/Testing/daisy/5885826924_38fdc6bcaa_n.jpg
inflating: flowers/Testing/daisy/5896103923_075a988bed_n.jpg
inflating: flowers/Testing/daisy/5896105367_fa08a65869_n.jpg
inflating: flowers/Testing/daisy/5896110423_e084b33401_n.jpg
inflating: flowers/Testing/daisy/5896674046_a4879f718e_n.jpg
inflating: flowers/Testing/daisy/5896675418_e6ff20a739_n.jpg
inflating: flowers/Testing/daisy/5896676090_68bb74b1e9_n.jpg
inflating: flowers/Testing/daisy/5896679822_5f60d35c33_n.jpg
inflating: flowers/Testing/daisy/5896680664_641de2de5a_n.jpg
inflating: flowers/Testing/daisy/5904946193_bd1eb1f39d_n.jpg
inflating: flowers/Testing/daisy/5905502226_bb23bd8fa0_n.jpg
inflating: flowers/Testing/daisy/5905504340_1d60fa9611_n.jpg
inflating: flowers/Testing/daisy/5924910021_b6debeb7b5_n.jpg
inflating: flowers/Testing/daisy/5944315415_2be8abeb2f_m.jpg
inflating: flowers/Testing/daisy/5948835387_5a98d39eff_m.jpg
inflating: flowers/Testing/daisy/5973488341_50bdf6cee3_n.jpg
inflating: flowers/Testing/daisy/5973491805_556bba93cc.jpg
inflating: flowers/Testing/daisy/5981645737_29eceac291_m.jpg
inflating: flowers/Testing/daisy/5997702776_c7bc37aa6b_n.jpg
inflating: flowers/Testing/daisy/6046940312_8faf552f3e_n.jpg
inflating: flowers/Testing/daisy/6054952060_c88612f3c5_n.jpg
inflating: flowers/Testing/daisy/6089825811_80f253fbe1.jpg
inflating: flowers/Testing/daisy/6095817094_3a5b1d793d.jpg
inflating: flowers/Testing/daisy/6136947177_47ff445eb4_n.jpg
inflating: flowers/Testing/daisy/6148728633_27afc47b0c_m.jpg
inflating: flowers/Testing/daisy/6207492986_0ff91f3296.jpg
inflating: flowers/Testing/daisy/6208851904_9d916ebb32_n.jpg
inflating: flowers/Testing/daisy/6210664514_f1d211217a.jpg
inflating: flowers/Testing/daisy/6299498346_b9774b6500.jpg
inflating: flowers/Testing/daisy/6299910262_336309ffa5_n.jpg
inflating: flowers/Testing/daisy/6323721068_3d3394af6d_n.jpg
inflating: flowers/Testing/daisy/6480809573_76a0074b69_n.jpg
inflating: flowers/Testing/daisy/6480809771_b1e14c5cc2_m.jpg
inflating: flowers/Testing/daisy/6529588249_d9cbe68aab_n.jpg
inflating: flowers/Testing/daisy/6596277835_9f86da54bb.jpg
inflating: flowers/Testing/daisy/6776075110_1ea7a09dd4_n.jpg
inflating: flowers/Testing/daisy/6864242336_0d12713fe5_n.jpg
inflating: flowers/Testing/daisy/6884975451_c74f445d69_m.jpg
inflating: flowers/Testing/daisy/6910811638_aa6f17df23.jpg
inflating: flowers/Testing/daisy/6950173662_5e9473003e_n.jpg
inflating: flowers/Testing/daisy/6978826370_7b9aa7c7d5.jpg
inflating: flowers/Testing/daisy/7066602021_2647457985_m.jpg
inflating: flowers/Testing/daisy/7133935763_82b17c8e1b_n.jpg
inflating: flowers/Testing/daisy/7188513571_c8527b123a_n.jpg
inflating: flowers/Testing/daisy/7189043225_2fe781439a_n.jpg
inflating: flowers/Testing/daisy/7191221492_610035de7c_m.jpg
inflating: flowers/Testing/daisy/7199968650_72afc16d31_m.jpg
inflating: flowers/Testing/daisy/7227973870_806d9d3e42_n.jpg
```

```

inflating: flowers/Testing/daisy/7288989324_c25d9febbf.jpg
inflating: flowers/Testing/daisy/7320089276_87b544e341.jpg
inflating: flowers/Testing/daisy/7335886184_d06a83f640.jpg
inflating: flowers/Testing/daisy/7357072446_c21c38c863_n.jpg
inflating: flowers/Testing/daisy/7358085448_b317d11cd5.jpg
inflating: flowers/Testing/daisy/7377004908_5bc0cde347_n.jpg
inflating: flowers/Testing/daisy/7410356270_9dffa1d0e2a_n.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/Training', target_size=(64,64),
```

```
xtest = test_datagen.flow_from_directory('/content/flowers/Testing', target_size=(64,64), cla
```

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

```
    Found 750 images belonging to 5 classes.
```

## # 3) CREATE MODEL

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

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

```
# Initializing the model
```

```
model = Sequential()
```

## # 4) ADD LAYERS

```
# There are 4 layers - Covolution layer, Max pooling layer, Flatten layer, Hidden layer, Outp
```

```
# Covolution layer
```

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

```
# Max pooling layer
```

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

```
# Flatten layer
```

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

```
# Hidden layer
```

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

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

```
# Output layer
```

```
model.add(Dense(5,activation='softmax'))
```

```
# 5) COMPILE THE MODEL
```

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

```
# 6) 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:7: UserWarning: `Model.fit_
import sys
Epoch 1/10
44/44 [=====] - 29s 634ms/step - loss: 1.7077 - accuracy: 0.392
Epoch 2/10
44/44 [=====] - 28s 636ms/step - loss: 1.0900 - accuracy: 0.564
Epoch 3/10
44/44 [=====] - 28s 623ms/step - loss: 1.0195 - accuracy: 0.593
Epoch 4/10
44/44 [=====] - 28s 626ms/step - loss: 0.9325 - accuracy: 0.646
Epoch 5/10
44/44 [=====] - 28s 632ms/step - loss: 0.8705 - accuracy: 0.658
Epoch 6/10
44/44 [=====] - 28s 640ms/step - loss: 0.8384 - accuracy: 0.675
Epoch 7/10
44/44 [=====] - 28s 628ms/step - loss: 0.7914 - accuracy: 0.705
Epoch 8/10
44/44 [=====] - 28s 627ms/step - loss: 0.7457 - accuracy: 0.715
Epoch 9/10
44/44 [=====] - 28s 632ms/step - loss: 0.7190 - accuracy: 0.725
Epoch 10/10
44/44 [=====] - 29s 657ms/step - loss: 0.6893 - accuracy: 0.736
<keras.callbacks.History at 0x7fd6651654d0>
```

```
# 7) SAVE THE MODEL
```

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

```
# 8) TEST THE MODEL
```

```
import numpy as np
from tensorflow.keras.preprocessing import image
op = ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
```

```
img = image.load_img('/content/flowers/Testing/daisy/11642632_1e7627a2cc.jpg',target_size=(64
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
```

```
print(pred, model.predict(x))
print(op[pred])
```

```
# wrong prediction
```

```
1 [[0. 1. 0. 0. 0.]]
dandelion
```

```
img = image.load_img('/content/flowers/Testing/rose/15498482197_8878cdfb07_n.jpg',target_size
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
print(op[pred])
```

```
2 [[0. 0. 1. 0. 0.]]
rose
```

```
img = image.load_img('/content/flowers/Testing/tulip/14084211971_0f921f11fe_n.jpg',target_siz
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
print(op[pred])
```

```
# wrong prediction
```

```
2 [[0.0000000e+00 0.0000000e+00 1.0000000e+00 0.0000000e+00 1.6980128e-12]]
rose
```

```
img = image.load_img('/content/flowers/Testing/sunflower/5180859236_60aa57ff9b_n.jpg',target_
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
print(op[pred])
```

```
3 [[0. 0. 0. 1. 0.]]
sunflower
```

```
# TUNING THE MODEL FOR MORE ACCURATE PREDICTION
```

```
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

```
early_stop = EarlyStopping(monitor='val_accuracy',
                           patience=5)
```

```
lr = ReduceLROnPlateau(monitor='val_accuracy',
                       factor=0.5,
```

```

        patience=5,
        min_lr=0.00001)

callbacks = [early_stop,lr]

model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=100,
                    callbacks=callbacks,
                    validation_data=xtest,
                    validation_steps=len(xtest),)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:20: UserWarning: `Model.fit
Epoch 1/100
44/44 [=====] - 28s 625ms/step - loss: 0.6726 - accuracy: 0.743
Epoch 2/100
44/44 [=====] - 27s 616ms/step - loss: 0.6292 - accuracy: 0.758
Epoch 3/100
44/44 [=====] - 28s 623ms/step - loss: 0.5962 - accuracy: 0.774
Epoch 4/100
44/44 [=====] - 28s 623ms/step - loss: 0.5785 - accuracy: 0.779
Epoch 5/100
44/44 [=====] - 28s 640ms/step - loss: 0.5489 - accuracy: 0.796
Epoch 6/100
44/44 [=====] - 29s 655ms/step - loss: 0.5140 - accuracy: 0.816
Epoch 7/100
44/44 [=====] - 28s 643ms/step - loss: 0.4971 - accuracy: 0.817
Epoch 8/100
44/44 [=====] - 29s 651ms/step - loss: 0.5052 - accuracy: 0.816
Epoch 9/100
44/44 [=====] - 28s 643ms/step - loss: 0.4848 - accuracy: 0.826
Epoch 10/100
44/44 [=====] - 28s 634ms/step - loss: 0.4612 - accuracy: 0.831
Epoch 11/100
44/44 [=====] - 28s 637ms/step - loss: 0.4254 - accuracy: 0.843
Epoch 12/100
44/44 [=====] - 28s 637ms/step - loss: 0.4008 - accuracy: 0.858
Epoch 13/100
44/44 [=====] - 28s 635ms/step - loss: 0.3699 - accuracy: 0.863
Epoch 14/100
44/44 [=====] - 28s 638ms/step - loss: 0.3492 - accuracy: 0.873
<keras.callbacks.History at 0x7fd665137cd0>

```

```

img = image.load_img('/content/flowers/Testing/tulip/14084211971_0f921f11fe_n.jpg',target_size=(256,256))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
print(op[pred])

```

```
# correct prediction after tuning the model
```

```
4 [[0. 0. 0. 0. 1.]]  
tulip
```

```
img = image.load_img('/content/flowers/Testing/daisy/11642632_1e7627a2cc.jpg',target_size=(64  
x = image.img_to_array(img)  
x = np.expand_dims(x,axis=0)  
pred = np.argmax(model.predict(x))  
print(pred, model.predict(x))  
print(op[pred])
```

```
# after tuning the model - correct prediction
```

```
0 [[1. 0. 0. 0. 0.]]  
daisy
```

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

