

▼ Importing Packages

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator as idm
import numpy as np
import warnings
#Supressing warnings
warnings.filterwarnings('ignore')
```

▼ 2.Image Augmentation

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

Mounted at /content/drive

```
# Creating augmentation on training variable
train_flowers=idm(rescale=1./255,zoom_range=0.2,horizontal_flip=True)
```

```
# Passing training data to train variable
Xtrain = train_flowers.flow_from_directory('/content/drive/MyDrive/DATASET/',target_size=(
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-2-ef61f13bf170> in <module>
      3
      4 # Passing training data to train variable
----> 5 Xtrain =
train_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flowers-
Dataset',target_size=(76,76),class_mode='categorical',batch_size=100)
```

1 frames

```
/usr/local/lib/python3.7/dist-packages/keras/preprocessing/image.py in
__init__(self, directory, image_data_generator, target_size, color_mode, classes,
class_mode, batch_size, shuffle, seed, data_format, save_to_dir, save_prefix,
save_format, follow_links, subset, interpolation, keep_aspect_ratio, dtype)
    505     if not classes:
    506         classes = []
--> 507     for subdir in sorted(os.listdir(directory)):
    508         if os.path.isdir(os.path.join(directory, subdir)):
    509             classes.append(subdir)
```

```
FileNotFoundError: [Errno 2] No such file or directory:
'/content/drive/MyDrive/IBM/Flowers-Dataset'
```

```
# Creating augmentation on testing variable
test_flowers=idm(rescale=1./255)
```

```
# Passing testing data to test variable
Xtest = test_flowers.flow_from_directory('/content/drive/MyDrive/IBM/Flower_Training',target_size=(224,224))

Found 204 images belonging to 5 classes.
```

▼ 3. Create Model

```
Flower_model = Sequential()
Flower_model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3)))
Flower_model.add(MaxPooling2D(pool_size=(2,2)))
Flower_model.add(Flatten())
Flower_model.add(Dense(300,activation='relu'))
Flower_model.add(Dense(150,activation='relu'))
Flower_model.add(Dense(5,activation='softmax'))
```

▼ 4. Compile the Model

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

▼ 5. Fit the Model

```
Flower_model.fit_generator(Xtrain,steps_per_epoch= len (Xtrain),epochs= 10,validation_data=(Xtest,Xtest.labels))
```

```
Epoch 1/10
42/42 [=====] - 567s 14s/step - loss: 1.9592 - accuracy: 0.1250
Epoch 2/10
42/42 [=====] - 26s 618ms/step - loss: 1.1221 - accuracy: 0.2500
Epoch 3/10
42/42 [=====] - 26s 612ms/step - loss: 1.0173 - accuracy: 0.3750
Epoch 4/10
42/42 [=====] - 26s 611ms/step - loss: 0.9552 - accuracy: 0.5000
Epoch 5/10
42/42 [=====] - 26s 620ms/step - loss: 0.8832 - accuracy: 0.6250
Epoch 6/10
42/42 [=====] - 26s 621ms/step - loss: 0.8373 - accuracy: 0.7500
Epoch 7/10
42/42 [=====] - 26s 615ms/step - loss: 0.8125 - accuracy: 0.8750
Epoch 8/10
42/42 [=====] - 26s 608ms/step - loss: 0.7663 - accuracy: 0.8750
Epoch 9/10
42/42 [=====] - 26s 616ms/step - loss: 0.7333 - accuracy: 0.8750
Epoch 10/10
42/42 [=====] - 26s 613ms/step - loss: 0.7128 - accuracy: 0.8750
<keras.callbacks.History at 0x7fd5aec82f50>
```

▼ 7. Save the model

```
Flower_model.save('Flower.h5')
```

▼ 8. Test the model

```
test_img=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/sunflower/200557977_bf  
test_img
```



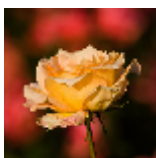
```
x=image.img_to_array(test_img)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category=['daisy','dandelion','rose','sunflower','tulip']  
Prediction_category[predicted]  
  
'sunflower'
```

```
test_img1=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/daisy/1140299375_3aa7  
test_img1
```



```
x=image.img_to_array(test_img1)  
x=np.expand_dims(x,axis=0)  
predicted=np.argmax(Flower_model.predict(x))  
Prediction_category[predicted]  
  
'daisy'
```

```
test_img2=image.load_img('/content/drive/MyDrive/IBM/Flowers-Dataset/rose/7251352826_69b62  
test_img2
```



```
x=image.img_to_array(test_img2)
x=np.expand_dims(x,axis=0)
predicted=np.argmax(Flower_model.predict(x))
Prediction_category[predicted]

'rose'
```

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

