Dataset

```
!unzip '/content/drive/MyDrive/Colab Notebooks/Asssignment 3/Copy of Flowers-Dataset.zip'
```

```
  inflating: flowers/tulip/21091489585_0dcb89e65e_n.jpg
  inflating: flowers/tulip/21091503556_8b46da140e_n.jpg
  inflating: flowers/tulip/212720516_df4965ebda_n.jpg
  inflating: flowers/tulip/2220085701_896054d263_n.jpg
  inflating: flowers/tulip/2229804138_db9cba3443_n.jpg
  inflating: flowers/tulip/2232289392_9a79a0c5cb_n.jpg
  inflating: flowers/tulip/2243427551_809b603992_z.jpg
  inflating: flowers/tulip/2249756775_02e693beda_n.jpg
  inflating: flowers/tulip/2254152047_d3bf8903cd_n.jpg
  inflating: flowers/tulip/2256214682_130c01d9d9.jpg
  inflating: flowers/tulip/2256230386_08b54ca760.jpg
  inflating: flowers/tulip/2271507463_15c48d41c4_n.jpg
  inflating: flowers/tulip/2272006181_785f1be94f_n.jpg
  inflating: flowers/tulip/2280950463_86510c2789_n.jpg
  inflating: flowers/tulip/2294116183_a30d2aa2c1_m.jpg
  inflating: flowers/tulip/2322670828_34115a7050.jpg
  inflating: flowers/tulip/2333321040_3960b9d67e_n.jpg
  inflating: flowers/tulip/2336919121_851ebc4754.jpg
  inflating: flowers/tulip/2344751399_71620039f2_n.jpg
  inflating: flowers/tulip/2351637471_5dd34fd3ac_n.jpg
  inflating: flowers/tulip/2361075034_cf730b8682.jpg
  inflating: flowers/tulip/2374855021_21959b40c0_n.jpg
  inflating: flowers/tulip/2399982682_16929d1f6d_n.jpg
  inflating: flowers/tulip/2402342888_dd65677013.jpg
  inflating: flowers/tulip/2412250315_a04171da51_n.jpg
  inflating: flowers/tulip/2418823693_72eec80f42_n.jpg
  inflating: flowers/tulip/2421740440_f82ced8582.jpg
  inflating: flowers/tulip/2425067141_b27043a800_m.jpg
  inflating: flowers/tulip/2426847695_4b8409402e_n.jpg
  inflating: flowers/tulip/2426849837_baefd9a518_n.jpg
  inflating: flowers/tulip/2427626706_ffdf697f84_n.jpg
  inflating: flowers/tulip/2430566689_8543552f9b.jpg
  inflating: flowers/tulip/2431737309_1468526f8b.jpg
  inflating: flowers/tulip/2432389721_4d14971060_n.jpg
  inflating: flowers/tulip/2434178332_7fcf85aa95_n.jpg
  inflating: flowers/tulip/2436998042_4906ea07af.jpg
  inflating: flowers/tulip/2440874162_27a7030402_n.jpg
  inflating: flowers/tulip/2447151631_7551e6377b_n.jpg
  inflating: flowers/tulip/2489638840_72ff3ee527_n.jpg
  inflating: flowers/tulip/2503489175_f0848d3e8e.jpg
  inflating: flowers/tulip/251811158_75fa3034ff.jpg
  inflating: flowers/tulip/2535936698_78cc03df3f_n.jpg
  inflating: flowers/tulip/25429468133_6bfba75d94_n.jpg
  inflating: flowers/tulip/25429501953_a1f9ce09e6_n.jpg
  inflating: flowers/tulip/25759191500_1c0da35828_n.jpg
  inflating: flowers/tulip/25965526231_941b6a216d_n.jpg
  inflating: flowers/tulip/25965548411_dbbe26262b_n.jpg
  inflating: flowers/tulip/26564770956_ac4800ae85_n.jpg
  inflating: flowers/tulip/26685647236_8211cb3e95_n.jpg
  inflating: flowers/tulip/26685648806_c76dd583b9_n.jpg
  inflating: flowers/tulip/2785458179_9130812eef_m.jpg
  inflating: flowers/tulip/2813649953_2b0f20fe94_n.jpg
  inflating: flowers/tulip/2834890466_1cf220fba1.jpg
  inflating: flowers/tulip/2936181186_38ff43492e.jpg
  inflating: flowers/tulip/3002863623_cd83d6e634.jpg
  inflating: flowers/tulip/3011223301_09b4e3edb7.jpg
  inflating: flowers/tulip/303858799_942b9c09e7_m.jpg
  inflating: flowers/tulip/3143110904_66b4851a58_n.jpg
  inflating: flowers/tulip/3150964108_24dbec4b23_m.jpg
```

Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data = ImageDataGenerator(rescale=1./255, zoom_range=0.4, horizontal_flip=True, vertical_flip=True)
xtrain = train_data.flow_from_directory('/content/flowers', target_size=(64,64),class_mode='categorical',batch_size=100)
```

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

Creating CNN Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
cnn_model = Sequential()
```

Adding Layers

```
cnn_model.add(Convolution2D(32,(4,4), activation='relu', input_shape=(64,64,3)))
cnn_model.add(MaxPooling2D(pool_size=(3,3)))
cnn_model.add(Flatten())
cnn_model.add(Dense(100,activation='relu'))
cnn_model.add(Dense(400,activation='relu'))
cnn_model.add(Dense(800,activation='relu'))
cnn_model.add(Dense(1000,activation='relu'))
cnn_model.add(Dense(500,activation='relu'))
cnn_model.add(Dense(5,activation='softmax'))
```

Compile The Model

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

Fit The Model

```
cnn_model.fit(xtrain, steps_per_epoch=len(xtrain), epochs=80)
```

```
    Epoch 1/80
```

```
44/44 [==============================] - 23s 326ms/step - loss: 1.4632 - accuracy: 0.3465
Epoch 2/80
44/44 [==============================] - 16s 360ms/step - loss: 1.2408 - accuracy: 0.4628
Epoch 3/80
44/44 [==============================] - 14s 328ms/step - loss: 1.1547 - accuracy: 0.5112
Epoch 4/80
44/44 [==============================] - 14s 324ms/step - loss: 1.0998 - accuracy: 0.5457
Epoch 5/80
44/44 [==============================] - 14s 325ms/step - loss: 1.0559 - accuracy: 0.5675
Epoch 6/80
44/44 [==============================] - 14s 326ms/step - loss: 1.0344 - accuracy: 0.5791
Epoch 7/80
44/44 [==============================] - 14s 325ms/step - loss: 1.0161 - accuracy: 0.5946
Epoch 8/80
44/44 [==============================] - 14s 326ms/step - loss: 0.9884 - accuracy: 0.6078
Epoch 9/80
44/44 [==============================] - 14s 327ms/step - loss: 0.9597 - accuracy: 0.6132
Epoch 10/80
44/44 [==============================] - 14s 324ms/step - loss: 0.9369 - accuracy: 0.6308
Epoch 11/80
44/44 [==============================] - 14s 325ms/step - loss: 0.9509 - accuracy: 0.6280
Epoch 12/80
44/44 [==============================] - 14s 325ms/step - loss: 0.9322 - accuracy: 0.6352
Epoch 13/80
44/44 [==============================] - 14s 328ms/step - loss: 0.9070 - accuracy: 0.6599
Epoch 14/80
44/44 [==============================] - 14s 325ms/step - loss: 0.9010 - accuracy: 0.6542
Epoch 15/80
44/44 [==============================] - 16s 360ms/step - loss: 0.8668 - accuracy: 0.6627
Epoch 16/80
44/44 [==============================] - 14s 327ms/step - loss: 0.8729 - accuracy: 0.6660
Epoch 17/80
44/44 [==============================] - 14s 327ms/step - loss: 0.8381 - accuracy: 0.6732
Epoch 18/80
44/44 [==============================] - 15s 330ms/step - loss: 0.8493 - accuracy: 0.6771
Epoch 19/80
44/44 [==============================] - 14s 329ms/step - loss: 0.8333 - accuracy: 0.6720
Epoch 20/80
44/44 [==============================] - 14s 326ms/step - loss: 0.8160 - accuracy: 0.6780
Epoch 21/80
44/44 [==============================] - 14s 326ms/step - loss: 0.8197 - accuracy: 0.6757
```

```
Epoch 22/80
44/44 [==============================] - 14s 325ms/step - loss: 0.8107 - accuracy: 0.6836
Epoch 23/80
44/44 [==============================] - 15s 331ms/step - loss: 0.7977 - accuracy: 0.6919
Epoch 24/80
44/44 [==============================] - 14s 327ms/step - loss: 0.7844 - accuracy: 0.6926
Epoch 25/80
44/44 [==============================] - 14s 327ms/step - loss: 0.8034 - accuracy: 0.6926
Epoch 26/80
44/44 [==============================] - 14s 326ms/step - loss: 0.7880 - accuracy: 0.6924
Epoch 27/80
44/44 [==============================] - 16s 356ms/step - loss: 0.7771 - accuracy: 0.6982
Epoch 28/80
44/44 [==============================] - 14s 327ms/step - loss: 0.7662 - accuracy: 0.7037
Epoch 29/80
44/44 [==============================] - 14s 326ms/step - loss: 0.7625 - accuracy: 0.7054
```

Save The Model

```
cnn_model.save('finalmodel.h5')
```

Test The Model

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

```
for i in range(1,2):
  imagepath = f'/content/download ({i}).jpg'
  img = image.load_img(imagepath,target_size=(64,64))
  a = image.img_to_array(img)
  a = np.expand_dims(a,axis=0)
  pred = np.argmax(cnn_model.predict(a))
  print(pred, cnn_model.predict(a))
```

```
output = ['daisy','dandelion','rose','sunflower','tulip']
print(f'{i}\t{output[pred]}')
```

```
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 14ms/step
4 [[0. 0. 0. 0. 1.]]
1       tulip
```