

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

Mounted at /content/drive

```
!unzip "/content/drive/MyDrive/Flowers-Dataset.zip"
```

```
inflating: flowers/dandelion/25218127947_9d237779bb_n.jpg
inflating: flowers/dandelion/2521827947_9d237779bb_n.jpg
inflating: flowers/dandelion/2522454811_f87af57d8b.jpg
inflating: flowers/dandelion/2535727910_769c020c0d_n.jpg
inflating: flowers/dandelion/2535769822_513be6bbe9.jpg
inflating: flowers/dandelion/253622055_d72964a7fd_n.jpg
inflating: flowers/dandelion/2538797744_deb53ac253.jpg
inflating: flowers/dandelion/2540640433_dedd577263.jpg
inflating: flowers/dandelion/2542908888_25a1c78ff0.jpg
inflating: flowers/dandelion/2553703483_558d12668c_n.jpg
inflating: flowers/dandelion/2569516382_9fd7097b9b.jpg
inflating: flowers/dandelion/2596413098_7ef69b7e1d_m.jpg
inflating: flowers/dandelion/2597655841_07fb2955a4.jpg
inflating: flowers/dandelion/2598486434_bf349854f2_m.jpg
inflating: flowers/dandelion/2600382379_5791b0b35a_m.jpg
inflating: flowers/dandelion/26004221274_74900d17e1_n.jpg
inflating: flowers/dandelion/2608937632_cfd93bc7cd.jpg
inflating: flowers/dandelion/2620243133_e801981efe_n.jpg
inflating: flowers/dandelion/2622697182_ea4aff29dd_n.jpg
inflating: flowers/dandelion/2625836599_03e192266f.jpg
inflating: flowers/dandelion/2628514700_b6d5325797_n.jpg
inflating: flowers/dandelion/2634665077_597910235f_m.jpg
inflating: flowers/dandelion/2634666217_d5ef87c9f7_m.jpg
inflating: flowers/dandelion/2635422362_a1bf641547_m.jpg
inflating: flowers/dandelion/2637883118_cf6ce37be4_n.jpg
inflating: flowers/dandelion/2661585172_94707236be_m.jpg
inflating: flowers/dandelion/2670304799_a3f2eef516_m.jpg
inflating: flowers/dandelion/26741270544_f44f3a1b19_n.jpg
inflating: flowers/dandelion/2674176237_e265ea64cc_n.jpg
inflating: flowers/dandelion/2683330456_0f7bbce110_m.jpg
inflating: flowers/dandelion/2693136371_dde2570813.jpg
inflating: flowers/dandelion/2697283969_c1f9cbb936.jpg
inflating: flowers/dandelion/2698102820_f15445a3f7.jpg
inflating: flowers/dandelion/27166475803_f5503f51f8_n.jpg
inflating: flowers/dandelion/27186992702_449dfa54ef_n.jpg
inflating: flowers/dandelion/27299697786_75340698c5_n.jpg
inflating: flowers/dandelion/27446317092_ff9bb852d5_n.jpg
inflating: flowers/dandelion/2753166154_0cb51a127b.jpg
inflating: flowers/dandelion/2780702427_312333ef33.jpg
inflating: flowers/dandelion/2831102668_eb65cd40b9_n.jpg
inflating: flowers/dandelion/284497199_93a01f48f6.jpg
inflating: flowers/dandelion/284497233_c19801752c.jpg
inflating: flowers/dandelion/29138994986_267e0e36c9_n.jpg
inflating: flowers/dandelion/29157239893_f43793c697_n.jpg
inflating: flowers/dandelion/2938040169_eb38581359.jpg
inflating: flowers/dandelion/29535628436_2e79a9628d_n.jpg
inflating: flowers/dandelion/29556932571_f124d8ac5d_n.jpg
inflating: flowers/dandelion/2963905796_227d37ff12.jpg
inflating: flowers/dandelion/29687446176_096b86f44c_n.jpg
```

```

inflating: flowers/dandelion/2995221296_a6ddaccc39.jpg
inflating: flowers/dandelion/3005677730_2662753d3f_m.jpg
inflating: flowers/dandelion/3021333497_b927cd8596.jpg
inflating: flowers/dandelion/3149809654_6a4b31314d_n.jpg
inflating: flowers/dandelion/31530587330_ba31bd196e_n.jpg
inflating: flowers/dandelion/315645471_dda66c6338_m.jpg
inflating: flowers/dandelion/3198028825_fdfaa1d020.jpg
inflating: flowers/dandelion/32120685303_90b5f21ab2_n.jpg
inflating: flowers/dandelion/32558425090_d6b6e86d85_n.jpg
inflating: flowers/dandelion/32701230112_a33f8003a5_n.jpg

```

Image Augmentation

```

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True, verti
test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory(r"/content/flowers", target_size=(64,64), class_mc

```

Found 4317 images belonging to 5 classes.

```

x_test=test_datagen.flow_from_directory(r"/content/flowers", target_size=(64,64), class_mode

```

Found 4317 images belonging to 5 classes.

```

x_train.class_indices

```

```

{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

```

Initializing CNN And Create Model

```

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

```

Add layers

```

model=Sequential()

```

Input Layers (Convolution ,MaxPooling,Flatten)

```

model.add(Convolution2D(32, (3,3), input_shape=(64,64,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
Model: "sequential"

```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
flatten (Flatten)	(None, 30752)	0
Total params: 896		
Trainable params: 896		
Non-trainable params: 0		

Hidden Layers

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
```

Output Layers

```
model.add(Dense(5,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
```

180

Train the Model

```
model.fit_generator(x_train,steps_per_epoch=len(x_train), validation_data=x_test, validation_steps=10)
Epoch 3/30
180/180 [=====] - 23s 131ms/step - loss: 0.4911 - accuracy: 0.3481
Epoch 4/30
180/180 [=====] - 24s 132ms/step - loss: 0.4545 - accuracy: 0.3783
Epoch 5/30
180/180 [=====] - 24s 132ms/step - loss: 0.4424 - accuracy: 0.3913
Epoch 6/30
180/180 [=====] - 25s 138ms/step - loss: 0.4112 - accuracy: 0.4112
Epoch 7/30
180/180 [=====] - 24s 131ms/step - loss: 0.3783 - accuracy: 0.4424
Epoch 8/30
180/180 [=====] - 24s 131ms/step - loss: 0.3458 - accuracy: 0.4545
Epoch 9/30
180/180 [=====] - 23s 131ms/step - loss: 0.3481 - accuracy: 0.4675
Epoch 10/30
180/180 [=====] - 23s 131ms/step - loss: 0.3054 - accuracy: 0.4911
Epoch 11/30
```

```

180/180 [=====] - 24s 132ms/step - loss: 0.2810 - accura
Epoch 12/30
180/180 [=====] - 23s 130ms/step - loss: 0.2428 - accura
Epoch 13/30
180/180 [=====] - 23s 131ms/step - loss: 0.2493 - accura
Epoch 14/30
180/180 [=====] - 24s 131ms/step - loss: 0.2523 - accura
Epoch 15/30
180/180 [=====] - 23s 131ms/step - loss: 0.1955 - accura
Epoch 16/30
180/180 [=====] - 25s 138ms/step - loss: 0.2079 - accura
Epoch 17/30
180/180 [=====] - 24s 131ms/step - loss: 0.2032 - accura
Epoch 18/30
180/180 [=====] - 24s 131ms/step - loss: 0.1884 - accura
Epoch 19/30
180/180 [=====] - 24s 131ms/step - loss: 0.1739 - accura
Epoch 20/30
180/180 [=====] - 24s 131ms/step - loss: 0.1633 - accura
Epoch 21/30
180/180 [=====] - 24s 131ms/step - loss: 0.1720 - accura
Epoch 22/30
180/180 [=====] - 23s 130ms/step - loss: 0.1480 - accura
Epoch 23/30
180/180 [=====] - 23s 130ms/step - loss: 0.1433 - accura
Epoch 24/30
180/180 [=====] - 24s 131ms/step - loss: 0.1459 - accura
Epoch 25/30
180/180 [=====] - 23s 130ms/step - loss: 0.1271 - accura
Epoch 26/30
180/180 [=====] - 25s 137ms/step - loss: 0.1349 - accura
Epoch 27/30
180/180 [=====] - 23s 129ms/step - loss: 0.1226 - accura
Epoch 28/30
180/180 [=====] - 23s 129ms/step - loss: 0.1102 - accura
Epoch 29/30
180/180 [=====] - 23s 130ms/step - loss: 0.1168 - accura
Epoch 30/30
180/180 [=====] - 23s 130ms/step - loss: 0.1125 - accura
<keras.callbacks.History at 0x7f1c702ebe90>

```

Save The model

```
model.save('Flowers_classification_model1.h5')
```

Test The model

```

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

```

```

# Load the model
model=load_model('Flowers_classification_model1.h5')

```

```
img=image.load_img(r"/content/Red-Rose.jpg",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
# x_train.class_indices
index=['daisy','dandelion','rose','sunflower','tulip']
index[y[0]]
```

```
1/1 [=====] - 0s 55ms/step
'rose'
```

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

