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

```
Mounted at /content/drive
```

```
!unzip '/content/drive/MyDrive/Colab Notebooks/Flowers-Dataset.zip'
```

```
  inflating: flowers/tulip/8712270243_8512cf4fbd.jpg
  inflating: flowers/tulip/8712270665_57b5bda0a2_n.jpg
  inflating: flowers/tulip/8712282563_3819afb7bc.jpg
  inflating: flowers/tulip/8713357842_9964a93473_n.jpg
  inflating: flowers/tulip/8713387500_6a9138b41b_n.jpg
  inflating: flowers/tulip/8713388322_e5ae26263b_n.jpg
  inflating: flowers/tulip/8713389178_66bceb71a8_n.jpg
  inflating: flowers/tulip/8713390684_041148dd3e_n.jpg
  inflating: flowers/tulip/8713391394_4b679ea1e3_n.jpg
  inflating: flowers/tulip/8713392604_90631fb809_n.jpg
  inflating: flowers/tulip/8713394070_b24561b0a9.jpg
  inflating: flowers/tulip/8713396140_5af8136136.jpg
  inflating: flowers/tulip/8713397358_0505cc0176_n.jpg
  inflating: flowers/tulip/8713397694_bcbcbba2c2_n.jpg
  inflating: flowers/tulip/8713398114_bc96f1b624_n.jpg
  inflating: flowers/tulip/8713398614_88202e452e_n.jpg
  inflating: flowers/tulip/8713398906_28e59a225a_n.jpg
  inflating: flowers/tulip/8713407768_f880df361f.jpg
  inflating: flowers/tulip/8717900362_2aa508e9e5.jpg
  inflating: flowers/tulip/8722514702_7ecc68691c.jpg
  inflating: flowers/tulip/8723767533_9145dec4bd_n.jpg
  inflating: flowers/tulip/8729501081_b993185542_m.jpg
  inflating: flowers/tulip/8733586143_3139db6e9e_n.jpg
  inflating: flowers/tulip/8748266132_5298a91dcf_n.jpg
  inflating: flowers/tulip/8750288831_5e49a9f29b.jpg
  inflating: flowers/tulip/8757486380_90952c5377.jpg
  inflating: flowers/tulip/8758464923_75a5ffe320_n.jpg
  inflating: flowers/tulip/8758519201_16e8d2d781_n.jpg
  inflating: flowers/tulip/8759594528_2534c0ec65_n.jpg
  inflating: flowers/tulip/8759597778_7fca5d434b_n.jpg
  inflating: flowers/tulip/8759601388_36e2a50d98_n.jpg
  inflating: flowers/tulip/8759606166_8e475013fa_n.jpg
  inflating: flowers/tulip/8759618746_f5e39fdbf8_n.jpg
  inflating: flowers/tulip/8762189906_8223cef62f.jpg
  inflating: flowers/tulip/8762193202_0fbf2f6a81.jpg
  inflating: flowers/tulip/8768645961_8f1e097170_n.jpg
  inflating: flowers/tulip/8817622133_a42bb90e38_n.jpg
  inflating: flowers/tulip/8838347159_746d14e6c1_m.jpg
  inflating: flowers/tulip/8838354855_c474fc66a3_m.jpg
  inflating: flowers/tulip/8838914676_8ef4db7f50_n.jpg
  inflating: flowers/tulip/8838975946_f54194894e_m.jpg
  inflating: flowers/tulip/8838983024_5c1a767878_n.jpg
  inflating: flowers/tulip/8892851067_79242a7362_n.jpg
  inflating: flowers/tulip/8904780994_8867d64155_n.jpg
  inflating: flowers/tulip/8908062479_449200a1b4.jpg
  inflating: flowers/tulip/8908097235_c3e746d36e_n.jpg
  inflating: flowers/tulip/9019694597_2d3bbedb17.jpg
  inflating: flowers/tulip/9030467406_05e93ff171_n.jpg
  inflating: flowers/tulip/9048307967_40a164a459_m.jpg
  inflating: flowers/tulip/924782410_94ed7913ca_m.jpg
```

```
  inflating: flowers/tulip/9378657435_89fabf13c9_n.jpg
  inflating: flowers/tulip/9444202147_405290415b_n.jpg
  inflating: flowers/tulip/9446982168_06c4d71da3_n.jpg
  inflating: flowers/tulip/9831362123_5aac525a99_n.jpg
  inflating: flowers/tulip/9870557734_88eb3b9e3b_n.jpg
  inflating: flowers/tulip/9947374414_fdf1d0861c_n.jpg
  inflating: flowers/tulip/9947385346_3a8cacea02_n.jpg
  inflating: flowers/tulip/9976515506_d496c5e72c.jpg
```

```
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_mo
```

```
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}
```

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

```
model=Sequential()
```

```
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"
_________________________________________________________________
 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
_________________________________________________________________
```

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



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


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


len(x_train)
```

```
180
```

```
model.fit_generator(x_train,steps_per_epoch=len(x_train), validation_data=x_test, validati
```

```
Epoch 3/30
180/180 [==============================] - 32s 178ms/step - loss: 0.9286 - accura
Epoch 4/30
180/180 [==============================] - 32s 179ms/step - loss: 0.8492 - accura
Epoch 5/30
180/180 [==============================] - 34s 187ms/step - loss: 0.8032 - accura
Epoch 6/30
180/180 [==============================] - 33s 183ms/step - loss: 0.7605 - accura
Epoch 7/30
180/180 [==============================] - 32s 180ms/step - loss: 0.7071 - accura
Epoch 8/30
180/180 [==============================] - 33s 181ms/step - loss: 0.6585 - accura
Epoch 9/30
180/180 [==============================] - 33s 181ms/step - loss: 0.6306 - accura
Epoch 10/30
180/180 [==============================] - 32s 180ms/step - loss: 0.5846 - accura
Epoch 11/30
180/180 [==============================] - 33s 181ms/step - loss: 0.5644 - accura
Epoch 12/30
180/180 [==============================] - 33s 186ms/step - loss: 0.5113 - accura
Epoch 13/30
180/180 [==============================] - 32s 178ms/step - loss: 0.4701 - accura
Epoch 14/30
180/180 [==============================] - 32s 178ms/step - loss: 0.4517 - accura
Epoch 15/30
180/180 [==============================] - 34s 188ms/step - loss: 0.4134 - accura
Epoch 16/30
180/180 [==============================] - 32s 179ms/step - loss: 0.3778 - accura
Epoch 17/30
180/180 [==============================] - 32s 178ms/step - loss: 0.3695 - accura
Epoch 18/30
180/180 [==============================] - 32s 177ms/step - loss: 0.3322 - accura
```

```
Epoch 19/30
180/180 [==============================] - 33s 184ms/step - loss: 0.2891 - accura
Epoch 20/30
180/180 [==============================] - 32s 178ms/step - loss: 0.3045 - accura
Epoch 21/30
180/180 [==============================] - 32s 176ms/step - loss: 0.2753 - accura
Epoch 22/30
180/180 [==============================] - 33s 185ms/step - loss: 0.2448 - accura
Epoch 23/30
180/180 [==============================] - 32s 179ms/step - loss: 0.2455 - accura
Epoch 24/30
180/180 [==============================] - 32s 178ms/step - loss: 0.2023 - accura
Epoch 25/30
180/180 [==============================] - 32s 179ms/step - loss: 0.2331 - accura
Epoch 26/30
180/180 [==============================] - 33s 186ms/step - loss: 0.2265 - accura
Epoch 27/30
180/180 [==============================] - 32s 177ms/step - loss: 0.1868 - accura
Epoch 28/30
180/180 [==============================] - 32s 178ms/step - loss: 0.1673 - accura
Epoch 29/30
180/180 [==============================] - 34s 187ms/step - loss: 0.1642 - accura
Epoch 30/30
180/180 [==============================] - 32s 180ms/step - loss: 0.1391 - accura
<keras.callbacks.History at 0x7f02e3a8dc50>
```

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

```
ls
```

```
drive/  flowers/  Flowers_classification_model1.h5  sample_data/
```

```
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/flowers/sunflower/10386522775_4f8c616999_m.jpg",target_size=
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 91ms/step
'sunflower'
```

Colab paid products - Cancel contracts here

✓ 18m 39s completed at 12:15 PM