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

Mounted at /content/drive

## cd /content/drive/MyDrive/CNN

/content/drive/MyDrive/CNN

!unzip Flowers-Dataset.zip

```
☐→ Archive:

              Flowers-Dataset.zip
      inflating: flowers/daisy/100080576 f52e8ee070 n.jpg
      inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
      inflating: flowers/daisy/10172379554 b296050f82 n.jpg
      inflating: flowers/daisy/10172567486_2748826a8b.jpg
      inflating: flowers/daisy/10172636503 21bededa75 n.jpg
      inflating: flowers/daisy/102841525_bd6628ae3c.jpg
      inflating: flowers/daisy/10300722094_28fa978807_n.jpg
      inflating: flowers/daisy/1031799732 e7f4008c03.jpg
      inflating: flowers/daisy/10391248763_1d16681106_n.jpg
      inflating: flowers/daisy/10437754174 22ec990b77 m.jpg
      inflating: flowers/daisy/10437770546 8bb6f7bdd3 m.jpg
      inflating: flowers/daisy/10437929963 bc13eebe0c.jpg
      inflating: flowers/daisy/10466290366_cc72e33532.jpg
      inflating: flowers/daisy/10466558316_a7198b87e2.jpg
      inflating: flowers/daisy/10555749515 13a12a026e.jpg
      inflating: flowers/daisy/10555815624 dc211569b0.jpg
      inflating: flowers/daisy/10555826524_423eb8bf71_n.jpg
      inflating: flowers/daisy/10559679065_50d2b16f6d.jpg
      inflating: flowers/daisy/105806915 a9c13e2106 n.jpg
      inflating: flowers/daisy/10712722853_5632165b04.jpg
      inflating: flowers/daisy/107592979_aaa9cdfe78_m.jpg
      inflating: flowers/daisy/10770585085 4742b9dac3 n.jpg
      inflating: flowers/daisy/10841136265_af473efc60.jpg
Saving innflating: flowers/daisy/10993710036_2033222c91.jpg
      inflating: flowers/daisy/10993818044_4c19b86c82.jpg
      inflating: flowers/daisy/10994032453_ac7f8d9e2e.jpg
      inflating: flowers/daisy/11023214096_b5b39fab08.jpg
      inflating: flowers/daisy/11023272144_fce94401f2_m.jpg
      inflating: flowers/daisy/11023277956_8980d53169_m.jpg
      inflating: flowers/daisy/11124324295_503f3a0804.jpg
      inflating: flowers/daisy/1140299375 3aa7024466.jpg
      inflating: flowers/daisy/11439894966 dca877f0cd.jpg
      inflating: flowers/daisy/1150395827 6f94a5c6e4 n.jpg
      inflating: flowers/daisy/11642632_1e7627a2cc.jpg
      inflating: flowers/daisy/11834945233_a53b7a92ac_m.jpg
      inflating: flowers/daisy/11870378973 2ec1919f12.jpg
      inflating: flowers/daisy/11891885265 ccefec7284 n.jpg
      inflating: flowers/daisy/12193032636_b50ae7db35_n.jpg
      inflating: flowers/daisy/12348343085_d4c396e5b5_m.jpg
      inflating: flowers/daisy/12585131704_0f64b17059_m.jpg
      inflating: flowers/daisy/12601254324 3cb62c254a m.jpg
      inflating: flowers/daisy/1265350143 6e2b276ec9.jpg
      inflating: flowers/daisy/12701063955_4840594ea6_n.jpg
      inflating: flowers/daisy/1285423653_18926dc2c8_n.jpg
```

```
inflating: flowers/daisy/1286274236_1d7ac84efb_n.jpg
       inflating: flowers/daisy/12891819633 e4c82b51e8.jpg
       inflating: flowers/daisy/1299501272 59d9da5510 n.jpg
       inflating: flowers/daisy/1306119996_ab8ae14d72_n.jpg
       inflating: flowers/daisy/1314069875 da8dc023c6 m.jpg
       inflating: flowers/daisy/1342002397_9503c97b49.jpg
       inflating: flowers/daisy/134409839_71069a95d1_m.jpg
       inflating: flowers/daisy/1344985627 c3115e2d71 n.jpg
       inflating: flowers/daisy/13491959645 2cd9df44d6 n.jpg
       inflating: flowers/daisy/1354396826_2868631432_m.jpg
       inflating: flowers/daisy/1355787476_32e9f2a30b.jpg
       inflating: flowers/daisy/13583238844 573df2de8e m.jpg
       inflating: flowers/daisy/1374193928 a52320eafa.jpg
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data = ImageDataGenerator(rescale= 1./255,horizontal_flip = True,vertical_flip = Tru
test_data = ImageDataGenerator(rescale= 1./255)
x_train = train_data.flow_from_directory(r'/content/drive/MyDrive/CNN/flowers_dataset/Trai
     Found 3450 images belonging to 5 classes.
x_test = test_data.flow_from_directory(r"/content/drive/MyDrive/CNN/flowers_dataset/Testin
     Found 867 images belonging to 5 classes.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
my model = Sequential()
 Saving...
my_model.add(Convolution2D(32,(3,3),activation = "relu",input_shape = (64,64,3)))
my_model.add(MaxPooling2D(pool_size = (2,2)))
my_model.add(Flatten())
my_model.add(Dense(300, activation='relu'))
my_model.add(Dense(300, activation='relu'))
my_model.add(Dense(300, activation='relu'))
my_model.add(Dense(5, activation="softmax"))
```

```
my_model.compile(loss="categorical_crossentropy", metrics=["accuracy"], optimizer='adam')
my_model.fit(x_train, epochs = 10, validation_data=x_test, steps_per_epoch=len(x_train), v
  Epoch 1/10
  133/133 [================== ] - 42s 316ms/step - loss: 0.7164 - accuracy:
  Epoch 2/10
  133/133 [=============== ] - 40s 298ms/step - loss: 0.6944 - accuracy:
  Epoch 3/10
  133/133 [==================== ] - 40s 297ms/step - loss: 0.6963 - accuracy:
  Epoch 4/10
  Epoch 5/10
  133/133 [=============== ] - 41s 309ms/step - loss: 0.6434 - accuracy:
  Epoch 6/10
  Epoch 7/10
  Epoch 8/10
  133/133 [================== ] - 41s 308ms/step - loss: 0.5418 - accuracy:
  Epoch 9/10
  Epoch 10/10
  133/133 [============== ] - 40s 297ms/step - loss: 0.4957 - accuracy:
  <keras.callbacks.History at 0x7fed539646d0>
my_model.fit(x_train, epochs = 10, validation_data=x_test, steps_per_epoch=len(x_train), v
  Epoch 1/10
  Epoch 2/10
  133/133 [================= ] - 41s 306ms/step - loss: 0.4708 - accuracy:
  Epoch 3/10
  Epoch 4/10
Epoch 5/10
  Epoch 6/10
  Epoch 7/10
  Epoch 8/10
  Epoch 9/10
  Epoch 10/10
  133/133 [================= ] - 42s 314ms/step - loss: 0.3427 - accuracy:
  <keras.callbacks.History at 0x7fed5392a7d0>
my model.fit(x train, epochs = 10, validation data=x test, steps per epoch=len(x train), v
  Epoch 1/10
```

```
Epoch 2/10
  133/133 [================== ] - 41s 308ms/step - loss: 0.3309 - accuracy:
  Epoch 3/10
  133/133 [================== ] - 40s 300ms/step - loss: 0.3483 - accuracy:
  Epoch 4/10
  133/133 [=============== ] - 40s 301ms/step - loss: 0.2861 - accuracy:
  Epoch 5/10
  133/133 [============ ] - 42s 315ms/step - loss: 0.2925 - accuracy:
  Epoch 6/10
  133/133 [============== ] - 40s 301ms/step - loss: 0.2820 - accuracy:
  Epoch 7/10
  Epoch 8/10
  Epoch 9/10
  Epoch 10/10
  <keras.callbacks.History at 0x7fed538e53d0>
my_model.fit(x_train, epochs = 6, validation_data=x_test, steps_per_epoch=len(x_train), va
  Epoch 1/6
  Epoch 2/6
  Epoch 3/6
  Epoch 4/6
  Epoch 5/6
  133/133 [=============== ] - 41s 308ms/step - loss: 0.2163 - accuracy:
  Epoch 6/6
  <keras.callbacks.History at 0x7fed538a4350>
Saving...
my model.save('flowers prediction.h5')
Testing
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model = load_model('flowers_prediction.h5')
photo = image.load img("/content/drive/MyDrive/CNN/flowers dataset/Training/tulip/13530786
photo
```



Saving... ×

Colab paid products - Cancel contracts here