```
cd /content/drive/MyDrive/CNN
```

```
/content/drive/MyDrive/CNN
```

```
!unzip flowers_data.zip
```

```
Archive:  flowers_data.zip
   creating: flowers_data/
   creating: flowers_data/Testing/
   creating: flowers_data/Testing/daisy/
  inflating: flowers_data/Testing/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers_data/Testing/daisy/102841525_bd6628ae3c.jpg
  inflating: flowers_data/Testing/daisy/105806915_a9c13e2106_n.jpg
  inflating: flowers_data/Testing/daisy/107592979_aaa9cdfe78_m.jpg
  inflating: flowers_data/Testing/daisy/134409839_71069a95d1_m.jpg
  inflating: flowers_data/Testing/daisy/144076848_57e1d662e3_m.jpg
 extracting: flowers_data/Testing/daisy/144099102_bf63a41e4f_n.jpg
  inflating: flowers_data/Testing/daisy/147068564_32bb4350cc.jpg
 extracting: flowers_data/Testing/daisy/153210866_03cc9f2f36.jpg
 extracting: flowers_data/Testing/daisy/154332674_453cea64f4.jpg
  inflating: flowers_data/Testing/daisy/158869618_f1a6704236_n.jpg
  inflating: flowers_data/Testing/daisy/162362896_99c7d851c8_n.jpg
  inflating: flowers_data/Testing/daisy/162362897_1d21b70621_m.jpg
  inflating: flowers_data/Testing/daisy/163978992_8128b49d3e_n.jpg
 extracting: flowers_data/Testing/daisy/169371301_d9b91a2a42.jpg
  inflating: flowers_data/Testing/daisy/171972704_389cf7a953.jpg
  inflating: flowers_data/Testing/daisy/172882635_4cc7b86731_m.jpg
 extracting: flowers_data/Testing/daisy/174131220_c853df1287.jpg
  inflating: flowers_data/Testing/daisy/175106495_53ebdef092_n.jpg
  inflating: flowers_data/Testing/daisy/176375506_201859bb92_m.jpg
  inflating: flowers_data/Testing/daisy/19544831049_0d738d4872_m.jpg
  inflating: flowers_data/Testing/daisy/19653086178_28156b7ce4_m.jpg
  inflating: flowers_data/Testing/daisy/19813618946_93818db7aa_m.jpg
  inflating: flowers_data/Testing/daisy/19834392829_7d697871f6.jpg
  inflating: flowers_data/Testing/daisy/19865728236_a62f8f445b_n.jpg
  inflating: flowers_data/Testing/daisy/19975899671_ebc42b7865_n.jpg
  inflating: flowers_data/Testing/daisy/20182559506_40a112f762.jpg
 extracting: flowers_data/Testing/daisy/20289938802_e16fa9f23d.jpg
  inflating: flowers_data/Testing/daisy/20329326505_a777c71cc2.jpg
  inflating: flowers_data/Testing/daisy/20580471306_ab5a011b15_n.jpg
  inflating: flowers_data/Testing/daisy/20619292635_9857a12d54.jpg
  inflating: flowers_data/Testing/daisy/20685027271_0e7306e7c1_n.jpg
  inflating: flowers_data/Testing/daisy/20703737132_179560d0fb.jpg
  inflating: flowers_data/Testing/daisy/20773528301_008fcbc5a1_n.jpg
  inflating: flowers_data/Testing/daisy/20948886919_cac7844f34_n.jpg
  inflating: flowers_data/Testing/daisy/21402054779_759366efb0_n.jpg
  inflating: flowers_data/Testing/daisy/21626652132_97e1318bb8_m.jpg
  inflating: flowers_data/Testing/daisy/21805938544_bf6bb0e4bc.jpg
  inflating: flowers_data/Testing/daisy/22244161124_53e457bb66_n.jpg
  inflating: flowers_data/Testing/daisy/22873310415_3a5674ec10_m.jpg
  inflating: flowers_data/Testing/daisy/23095658544_7226386954_n.jpg
  inflating: flowers_data/Testing/daisy/2581171297_b0a249b92b_n.jpg
  inflating: flowers_data/Testing/daisy/2590291468_2635d3e4e0_n.jpg
  inflating: flowers_data/Testing/daisy/2599662355_7782218c83.jpg
  inflating: flowers_data/Testing/daisy/2611119198_9d46b94392.jpg
  inflating: flowers_data/Testing/daisy/2612704455_efce1c2144_m.jpg
  inflating: flowers_data/Testing/daisy/2617111535_54c2ac8462.jpg
```

```
        inflating: flowers_data/Testing/daisy/2619413565_61a6cd3ac9_m.jpg
        inflating: flowers_data/Testing/daisy/2621723097_736febb4a4_n.jpg
        inflating: flowers_data/Testing/daisy/2627815904_919373e7f5.jpg
        inflating: flowers_data/Testing/daisy/2632216904_274aa17433.jpg
        inflating: flowers_data/Testing/daisy/2641979584_2b21c3fe29_m.jpg
        inflating: flowers_data/Testing/daisy/2642408410_61545fdc83_n.jpg
        inflating: flowers_data/Testing/daisy/2646438199_b309cffd65_n.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_data/Trainin
```

```
     Found 3450 images belonging to 5 classes.
```

```
x_test = test_data.flow_from_directory(r"/content/drive/MyDrive/CNN/flowers_data/Testing",
```

```
     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()
```

```
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 [==============================] - 39s 290ms/step - loss: 0.7330 - accuracy:
Epoch 2/10
133/133 [==============================] - 39s 292ms/step - loss: 0.7146 - accuracy:
Epoch 3/10
133/133 [==============================] - 40s 297ms/step - loss: 0.6777 - accuracy:
Epoch 4/10
133/133 [==============================] - 38s 284ms/step - loss: 0.6675 - accuracy:
Epoch 5/10
133/133 [==============================] - 39s 291ms/step - loss: 0.6282 - accuracy:
Epoch 6/10
133/133 [==============================] - 38s 288ms/step - loss: 0.6241 - accuracy:
Epoch 7/10
133/133 [==============================] - 39s 290ms/step - loss: 0.6017 - accuracy:
Epoch 8/10
133/133 [==============================] - 40s 298ms/step - loss: 0.5786 - accuracy:
Epoch 9/10
133/133 [==============================] - 38s 283ms/step - loss: 0.5632 - accuracy:
Epoch 10/10
133/133 [==============================] - 38s 285ms/step - loss: 0.5399 - accuracy:
<keras.callbacks.History at 0x7f371d2aded0>
```

```
my_model.fit(x_train, epochs = 10, validation_data=x_test, steps_per_epoch=len(x_train), v
```

```
Epoch 1/10
133/133 [==============================] - 38s 287ms/step - loss: 0.5066 - accuracy:
Epoch 2/10
133/133 [==============================] - 40s 302ms/step - loss: 0.5025 - accuracy:
Epoch 3/10
133/133 [==============================] - 38s 286ms/step - loss: 0.4714 - accuracy:
Epoch 4/10
133/133 [==============================] - 39s 296ms/step - loss: 0.4690 - accuracy:
Epoch 5/10
133/133 [==============================] - 38s 287ms/step - loss: 0.4525 - accuracy:
Epoch 6/10
133/133 [==============================] - 38s 286ms/step - loss: 0.4219 - accuracy:
Epoch 7/10
133/133 [==============================] - 40s 301ms/step - loss: 0.4177 - accuracy:
Epoch 8/10
133/133 [==============================] - 38s 287ms/step - loss: 0.4177 - accuracy:
Epoch 9/10
133/133 [==============================] - 38s 289ms/step - loss: 0.4051 - accuracy:
Epoch 10/10
133/133 [==============================] - 38s 288ms/step - loss: 0.3537 - accuracy:
<keras.callbacks.History at 0x7f371bbf1110>
```

```
my_model.fit(x_train, epochs = 10, validation_data=x_test, steps_per_epoch=len(x_train), v
```

```
Epoch 1/10
133/133 [==============================] - 39s 291ms/step - loss: 0.3609 - accuracy:
Epoch 2/10
133/133 [==============================] - 39s 291ms/step - loss: 0.3526 - accuracy:
Epoch 3/10
133/133 [==============================] - 40s 302ms/step - loss: 0.3445 - accuracy:
Epoch 4/10
133/133 [==============================] - 38s 285ms/step - loss: 0.3170 - accuracy:
```

```
Epoch 5/10
133/133 [==============================] - 38s 289ms/step - loss: 0.3364 - accuracy:
Epoch 6/10
133/133 [==============================] - 40s 301ms/step - loss: 0.3045 - accuracy:
Epoch 7/10
133/133 [==============================] - 38s 288ms/step - loss: 0.2776 - accuracy:
Epoch 8/10
133/133 [==============================] - 40s 299ms/step - loss: 0.2938 - accuracy:
Epoch 9/10
133/133 [==============================] - 38s 286ms/step - loss: 0.2657 - accuracy:
Epoch 10/10
133/133 [==============================] - 38s 285ms/step - loss: 0.2533 - accuracy:
<keras.callbacks.History at 0x7f371ab8d990>
```

```
my_model.fit(x_train, epochs = 6, validation_data=x_test, steps_per_epoch=len(x_train), va
```

```
Epoch 1/6
133/133 [==============================] - 38s 284ms/step - loss: 0.2418 - accuracy:
Epoch 2/6
133/133 [==============================] - 40s 299ms/step - loss: 0.2343 - accuracy:
Epoch 3/6
133/133 [==============================] - 38s 285ms/step - loss: 0.2405 - accuracy:
Epoch 4/6
133/133 [==============================] - 38s 284ms/step - loss: 0.2287 - accuracy:
Epoch 5/6
133/133 [==============================] - 40s 297ms/step - loss: 0.2357 - accuracy:
Epoch 6/6
133/133 [==============================] - 38s 286ms/step - loss: 0.2211 - accuracy:
<keras.callbacks.History at 0x7f371c9c6450>
```

```
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/downloffad (3).jfif",target_size=(64,64))
```

```
photo
```

```
img= image.img_to_array(photo)
```

```
img = np.expand_dims(img,axis = 0)
```

```
pred = model.predict(img)
```

```
labels =['daisy','dandelion','rose','sunflower','tulip']
```

```
np.argmax(pred)
labels[np.argmax(pred)]
```

```
'daisy'
```