


```

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 = True)
```

```
test_data = ImageDataGenerator(rescale= 1./255)
```

```
x_train = train_data.flow_from_directory(r'/content/drive/MyDrive/CNN/flowers_dataset/Train')
```

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

```
x_test = test_data.flow_from_directory(r'/content/drive/MyDrive/CNN/flowers_dataset/Test')
```

```
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(64, 64, 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
 133/133 [=====] - 39s 294ms/step - loss: 0.6560 - accuracy:
 Epoch 5/10
 133/133 [=====] - 41s 309ms/step - loss: 0.6434 - accuracy:
 Epoch 6/10
 133/133 [=====] - 39s 294ms/step - loss: 0.6037 - accuracy:
 Epoch 7/10
 133/133 [=====] - 39s 295ms/step - loss: 0.5737 - accuracy:
 Epoch 8/10
 133/133 [=====] - 41s 308ms/step - loss: 0.5418 - accuracy:
 Epoch 9/10
 133/133 [=====] - 39s 295ms/step - loss: 0.5328 - accuracy:
 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
 133/133 [=====] - 42s 316ms/step - loss: 0.4991 - accuracy:
 Epoch 2/10
 133/133 [=====] - 41s 306ms/step - loss: 0.4708 - accuracy:
 Epoch 3/10
 133/133 [=====] - 39s 293ms/step - loss: 0.4629 - accuracy:
 Epoch 4/10
 133/133 [=====] - 39s 294ms/step - loss: 0.4459 - accuracy:
 Epoch 5/10
 133/133 [=====] - 39s 294ms/step - loss: 0.4510 - accuracy:
 Epoch 6/10
 133/133 [=====] - 41s 306ms/step - loss: 0.4171 - accuracy:
 Epoch 7/10
 133/133 [=====] - 39s 294ms/step - loss: 0.3759 - accuracy:
 Epoch 8/10
 133/133 [=====] - 39s 296ms/step - loss: 0.3821 - accuracy:
 Epoch 9/10
 133/133 [=====] - 40s 297ms/step - loss: 0.3501 - accuracy:
 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
 133/133 [=====] - 40s 297ms/step - loss: 0.3205 - accuracy:

```

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
133/133 [=====] - 40s 296ms/step - loss: 0.2935 - accuracy:
Epoch 8/10
133/133 [=====] - 40s 297ms/step - loss: 0.2522 - accuracy:
Epoch 9/10
133/133 [=====] - 41s 309ms/step - loss: 0.2682 - accuracy:
Epoch 10/10
133/133 [=====] - 40s 300ms/step - loss: 0.2424 - accuracy:
<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
133/133 [=====] - 42s 319ms/step - loss: 0.2394 - accuracy:
Epoch 2/6
133/133 [=====] - 40s 298ms/step - loss: 0.2613 - accuracy:
Epoch 3/6
133/133 [=====] - 39s 295ms/step - loss: 0.2261 - accuracy:
Epoch 4/6
133/133 [=====] - 40s 297ms/step - loss: 0.1953 - accuracy:
Epoch 5/6
133/133 [=====] - 41s 308ms/step - loss: 0.2163 - accuracy:
Epoch 6/6
133/133 [=====] - 39s 296ms/step - loss: 0.2151 - accuracy:
<keras.callbacks.History at 0x7fed538a4350>

```

Saving...

```
my_model.save('flowers_prediction.h5')
```

Testing

```

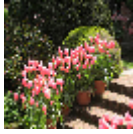
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

```

```
model = load_model('flowers_prediction.h5')
```

```
photo = image.load_img("/content/drive/MyDrive/CNN/flowers_dataset/Training/tulip/13530786
```

```
photo
```



```
img= image.img_to_array(photo)

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

pred = model.predict(img)

1/1 [=====] - 0s 137ms/step

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

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

'tulip'
```

Saving...



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

