```
import tensorflow as tf
import numpy as np
import cv2
import PIL.Image as Image
import os
import matplotlib.pyplot as plt
import tensorflow_hub as hub
import pathlib
```

```
Image_Shape = (224,224)
```

```
URL_dataset = "https://storage.googleapis.com/download.tensorflow.org/example_images/flower_p
```

```
data_dir = tf.keras.utils.get_file(origin=URL_dataset,
fname='flower_photos',untar=True)
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/example_im
228813984/228813984 [==============================] - 1s 0us/step
```

```
data_dir = pathlib.Path(data_dir)
```

```
image_count = len(list(data_dir.glob('*/*.jpg')))
print(image_count)
```

```
3670
```

```
flowers_images_dict = {
"daisy" : list(data_dir.glob('daisy/*')),
"dandelion" : list(data_dir.glob('dandelion/*')),
"roses" : list(data_dir.glob('roses/*')),
"sunflowers" : list(data_dir.glob('sunflowers/*')),
"tulips" : list(data_dir.glob('tulips/*'))
}
```

```
flowers_labels_dict= {
"daisy" : 0,
"dandelion" : 1,
"roses" : 2,
"sunflowers" : 3,
"tulips" : 4
```

```
}

X, Y = [],[]


for flower_name, images in flowers_images_dict.items():
  for image in images:
    img = cv2.imread(str(image))
    resized_img = cv2.resize(img, Image_Shape)
    X.append(resized_img)
    Y.append(flowers_labels_dict[flower_name])
X = np.array(X)
y = np.array(Y)


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
X_train_scaled = X_train / 255
X_test_scaled = X_test / 255


tf_model="https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4"



classifier = tf.keras.Sequential([
hub.KerasLayer(tf_model,input_shape=(224,224,3), trainable=False),
tf.keras.layers.Dense(len(flowers_labels_dict), activation="softmax")
])
classifier.summary()
classifier.compile(
optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=["accuracy"]
)
classifier.fit(X_train_scaled, y_train,epochs=5)
classifier.evaluate(X_test_scaled, y_test)
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 1280)              2257984

 dense (Dense)               (None, 5)                 6405

=================================================================
Total params: 2,264,389
Trainable params: 6,405
Non-trainable params: 2,257,984
_________________________________________________________________
Epoch 1/5
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1082: UserWar
```

```
  return dispatch_target(*args, **kwargs)
86/86 [==============================] - 82s 901ms/step - loss: 0.8665 - accuracy: 0.66
Epoch 2/5
86/86 [==============================] - 78s 908ms/step - loss: 0.4247 - accuracy: 0.84
Epoch 3/5
86/86 [==============================] - 80s 926ms/step - loss: 0.3304 - accuracy: 0.88
Epoch 4/5
86/86 [==============================] - 78s 904ms/step - loss: 0.2794 - accuracy: 0.91
Epoch 5/5
86/86 [==============================] - 77s 901ms/step - loss: 0.2366 - accuracy: 0.92
29/29 [==============================] - 27s 901ms/step - loss: 0.3671 - accuracy: 0.87
[0.36709731817245483, 0.8779956698417664]
```

```
from IPython import display
display.Image('/content/drive/MyDrive/rose.jpg',width=200,height=200)
```

```
from PIL import Image

img = Image.open("/content/drive/MyDrive/rose.jpg")
img = tf.keras.preprocessing.image.img_to_array(img.resize(Image_Shape))
img = np.array([img])
res = classifier.predict(img)
print("The prediction is : {}".format(list(flowers_labels_dict.keys())[np.argmax(res)]))
```

```
1/1 [==============================] - 0s 56ms/step
The prediction is : roses
```

Colab paid products - Cancel contracts here

✓ 0s completed at 12:09 AM