

TensorFlow hub is a library for using reusable parts of the machine learning models, which is a library which you can reuse the TensorFlow graphs, along with it's weights, and assets.

This repository contains two source files `retrain.py`, and `test.py`

- **retrain.py** Retrain the TensorFlow hub's module
- **test.py** Test an image on retrained model

1. Create a virtual environment

```
H:\tf-hub>virtualenv hub
Using base prefix 'c:\program files (x86)\microsoft visual studio\shared\python36_64'
New python executable in H:\tf-hub\hub\Scripts\python.exe
Installing setuptools, pip, wheel...
done.

H:\tf-hub>cd hub

H:\tf-hub\hub>dir
Volume in drive H is SYSTEM
Volume Serial Number is F6A4-DD4E

Directory of H:\tf-hub\hub

01/06/2019  08:14 PM  <DIR>          .
01/06/2019  08:14 PM  <DIR>          ..
10/18/2018  05:52 PM  <DIR>          Include
01/06/2019  08:14 PM  <DIR>          Lib
01/06/2019  08:14 PM  <DIR>          Scripts
01/06/2019  08:14 PM  <DIR>          tcl
                0 File(s)          0 bytes
                6 Dir(s)  65,377,394,688 bytes free

H:\tf-hub\hub>cd scripts
H:\tf-hub\hub\Scripts>activate
```

2. Install required dependencies

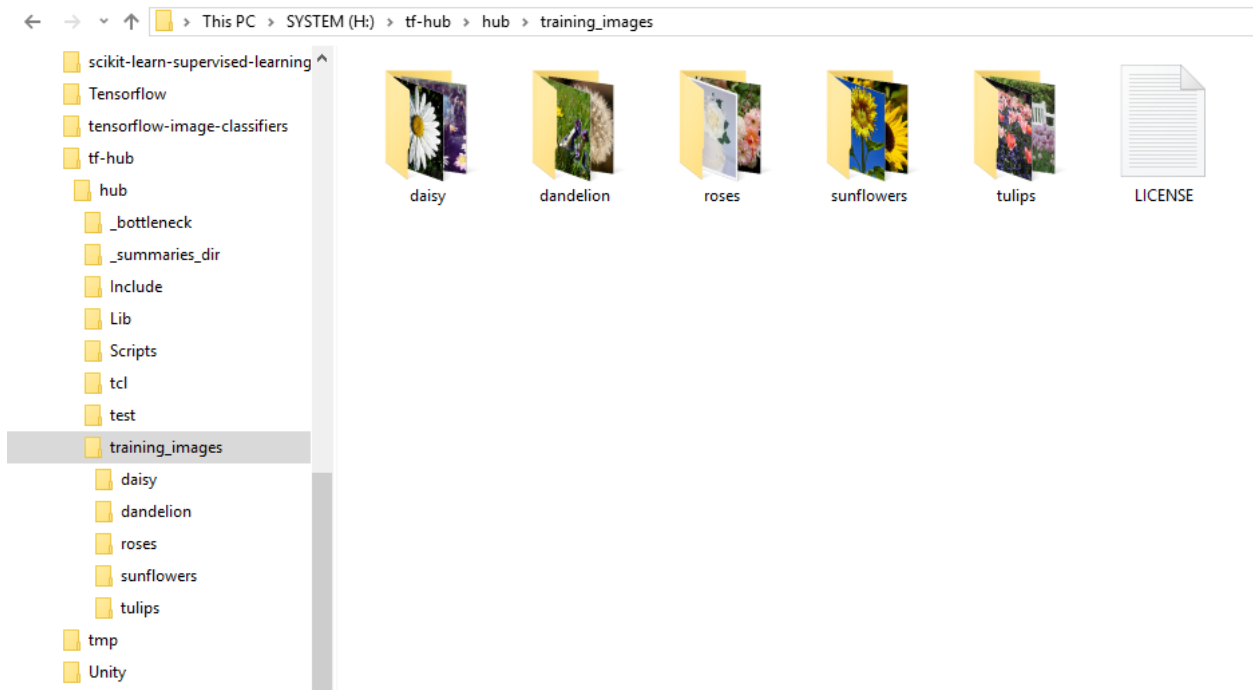
- NumPy
- TensorFlow
- TensorFlow hub

```
pip install numpy tensorflow tensorflow-hub
```

```
(hub) H:\tf-hub\hub\Scripts>activate

(hub) H:\tf-hub\hub\Scripts>pip install numpy tensorflow tensorflow-hub
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/51/70/7096a735b27359dbc0c380b23b9c9bd05fea62...
Collecting tensorflow
  Using cached https://files.pythonhosted.org/packages/05/cd/c171d2e33c0192b04560ce864c26eba83fed88...
Collecting tensorflow-hub
  Using cached https://files.pythonhosted.org/packages/5a/76/18c9aaf24f1b888bafa40351493815050e1000...
Collecting keras-preprocessing>=1.0.5 (from tensorflow)
Installing collected packages: tensorflow-hub, tensorflow, numpy
Successfully installed numpy-1.14.3 tensorflow-1.12.0 tensorflow-hub-0.1.0
```

3. Create a training directory



4. Train your model

Change the name of the **image_dir** which contains training set

```

41  def main():
42      image_dir = os.getcwd() + '\\training_images\\' # str    path to folders of labeled images
43      output_graph = os.getcwd() + '\\_graph.pb' # str    where to save the trained graph
44      intermediate_output_graphs_dir = os.getcwd() + '\\_intermediate_graph\\' # str    where to save the intermediate graphs
45      intermediate_store_frequency = 0 # int    How many steps to store intermediate graph. If "0" then will not store
46      output_labels = os.getcwd() + '\\output_labels.txt' # str    where to save the trained graph's labels
47      summaries_dir = os.getcwd() + '\\summaries_dir' # str    where to save summary logs for TensorBoard
48      how_many_training_steps = 100 # int    default 4000 how many training steps to run before ending
49      learning_rate = 0.01 # float    how large a learning rate to use when training

```

```
python retrain.py
```

```
(hub) H:\tf-hub\hub\Scripts>cd ..  
(hub) H:\tf-hub\hub>python retrain.py  
main()  
prepare_file_system() H:\tf-hub\hub\_summaries_dir  
create_image_lists(H:\tf-hub\hub\training_images\  
INFO:tensorflow:Looking for images in 'daisy'  
INFO:tensorflow:Looking for images in 'dandelion'  
INFO:tensorflow:Looking for images in 'roses'  
INFO:tensorflow:Looking for images in 'sunflowers'  
INFO:tensorflow:Looking for images in 'tulips'  
image_lists OrderedDict([('daisy', {'dir': 'daisy', 'training': ['100080576_f52e8ee070_n.jpg', '10
```

5. Test the images

Change the name of the **file_name** which is used to test the predictions

```
67 def main():
68     file_name = os.getcwd() + "\\test\\sunflower.jpg" # image to be processed
69     model_file = os.getcwd() + "\\_graph.pb" # graph/model to be executed
70     label_file = os.getcwd() + "\\_labels.txt" # name of file containing labels
71     input_height = 299 # input height
72     input_width = 299 # input width
73     input_mean = 1 # input mean
74     input_std = 255 # input std
75     input_layer = "Placeholder" # name of input layer
76     output_layer = "final_result" # name of output layer
77     graph = load_graph(model_file)
```

python test.py

```
add_evaluation_step(tensor([final_result:0, shape=(?, 3), dtype=float32], tensor([input/groundtruth_input:0, shape=(?, 3),
INFO:tensorflow:Froze 378 variables.
65     return label
INFO:tensorflow:Converted 378 variables to const ops.
67 def main():
68     file_name = os.getcwd() + "\\test\\sunflower.jpg" # image to be processed
69     model_file = os.getcwd() + "\\_graph.pb" # graph/model to be executed
70     label_file = os.getcwd() + "\\_labels.txt" # name of file containing labels
71     input_height = 299 # input height
72     input_width = 299 # input width
73     input_mean = 1 # input mean
74     input_std = 255 # input std
75     input_layer = "Placeholder" # name of input layer
76     output_layer = "final_result" # name of output layer

(hub) H:\tf-hub\hub>python test.py
2019-01-06 20:59:48.664466: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this
2
sunflowers 0.48977023
roses 0.19306159
tulips 0.13504855
dandelion 0.12648144
daisy 0.055638164
(hub) H:\tf-hub\hub>
```

Repository <https://github.com/gayankuruppu/tensorflowhub-retrain.py-test.py>