2. How to use TF-Slim

This is based on the tutorials written on the GitHub page https://github.com/tensorflow/models/tree/master/slim. What I did is modify the codes in the 'slim' folder so that I can use TF-Slim with my own images.

I assume that people who is reading this page have already installed Python, Tensorflow and downloaded code at https://github.com/tensorflow/models.

I can use TF-Slim in 3 major steps. First, transforming images into TFRecord File. Second, training and evaluating the model. Third, using the trained model(labelling images).

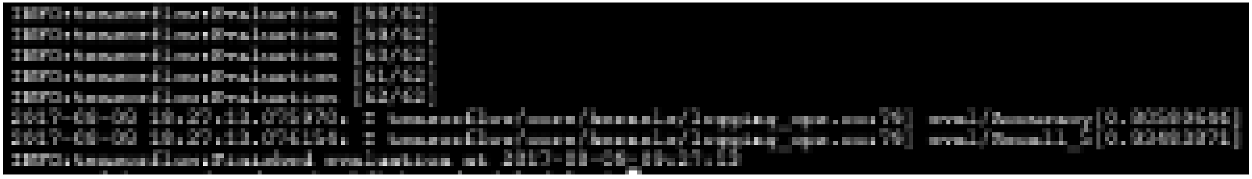
Before I start with the Real Images(which is collected from the camera in the wild), I practiced with the data that is not included in the TF-Slim as a default. So I worked with Caltech-256 Dataset. And below is the steps that I worked with TF-Slim and Caltech-256 dataset before I work with the real images.

1. Download Caltech-256 images.
   I can get at : http://www.vision.caltech.edu/Image_Datasets/Caltech256/256_ObjectCategories.tar
   1-1. Change names of folders : 256_Objectcategories\256_Objectcategories caltech256\caltech256_photos
   1-2. Move the location of folder caltech256 to tmp\

2. Transform dataset into TFRecord file
   For each dataset, I need to convert the raw data to TFRecord format which is available on the Tensorflow. So I need to modify some code in 'slim'
   2-1. Modify 'download_and_convert_data.py' : I modified it to file 'convert_data_for_caltech256'
   2-2. Make some files on slim\datasets : I added 'caltech256.py' and 'download_and_convert_caltech256.py'. I modified these files referring to the 'flowers.py' and 'download_and_convert_flowers.py'
   2-3. Modify 'dataset_factory.py' : I added categories for caltech256
   2-4. Execute **python convert_data_for_caltech256.py --dataset_name=caltech256 --dataset_dir=/tmp/caltech256** : During the execution you may get some errors. These are about the format of images(some of them are directories not images). It was okay after I erase the junk files.

3. Fine-tuning a model from an existing checkpoint(Training a model with a pre-trained model)
   3-1. Download checkpoint at : http://download.tensorflow.org/models/inception_v3_2016_08_28.tar.gz // and upzip it in the location tmp/my_checkpoints
   3-2. Execute **python train_image_classifier.py --train_dir=/tmp/train_inception_v3_caltech256_FineTune_logs --dataset_name=caltech256 --dataset_split_name=train --dataset_dir=/tmp/caltech256 --model_name=inception_v3 --checkpoint_path=/tmp/my_checkpoints/inception_v3.ckpt --checkpoint_exclude_scopes=InceptionV3/Logits,InceptionV3/AuxLogits --trainable_scopes=InceptionV3/Logits,InceptionV3/AuxLogits --max_number_of_steps=1000 --batch_size=32 --learning_rate=0.01 --learning_rate_decay_type=fixed --save_interval_secs=60 --save_summaries_secs=60 --log_every_n_steps=100 --optimizer=rmsprop --weight_decay=0.00004**
   This execution is about training the last layer so that it can result in the 257 class(which is the number of classes in Caltech256). It takes about 5~10mins to finish the training.
   3-3. Execute **python train_image_classifier.py --train_dir=/tmp/train_inception_v3_caltech256_FineTune_logs/all --dataset_name=caltech256 --dataset_split_name=train --dataset_dir=/tmp/caltech256 --model_name=inception_v3 --checkpoint_path=/tmp/train_inception_v3_caltech256_FineTune_logs --max_number_of_steps=500 --batch_size=32 --learning_rate=0.0001 --learning_rate_decay_type=fixed --save_interval_secs=60 --save_summaries_secs=60 --log_every_n_steps=10 --optimizer=rmsprop --weight_decay=0.00004**
   This execution is about training the whole layer. This takes about 5~10 mins to finish the training as well.

4. Evaluate the model of 3-3
   4-1. Execute **python eval_image_classifier.py –alsologtostderr --checkpoint_path=/tmp/train_inception_v3_caltech256_FineTune_logs/all/ --dataset_dir=/tmp/caltech256 --dataset_name=caltech256 --dataset_split_name=validation --model_name=inception_v3**



```
INFO:tensorflow:Evaluation [58/62]
INFO:tensorflow:Evaluation [59/62]
INFO:tensorflow:Evaluation [60/62]
INFO:tensorflow:Evaluation [61/62]
INFO:tensorflow:Evaluation [62/62]
2017-08-02 18:27:13.075870: I tensorflow/core/kernels/logging_ops.cc:79] eval/Accuracy[[illegible]]
2017-08-02 18:27:13.076154: I tensorflow/core/kernels/logging_ops.cc:79] eval/Recall_5[[illegible]]
INFO:tensorflow:Finished evaluation at 2017-08-02-18:27:13
```

   This is the result of the evaluation. Top1 Accuracy is about 81% and Top5 Accuracy is about 93%. Even though it took less than 30 minutes to train and evaluate, it gives me relatively high accuracy.

5. Label images
   5-1. Execute **python label_image.py** you can label the images with this python file

6. Label images per folder

6-1. You can classify images in a folder. Execute **python classify_images.py** you can classify all of the images in /slim/images/ folder. The result will be shown in CLASSIFIED folder

IMPORTANT

- You can get every single python code above at https://github.com/sai223/2017CSIRO
- You can contact me whenever you want at sai223@ajou.ac.kr OR rlarlghd1004@gmail.com