Documentation to fine-tune neural networks with TF-slim

0) Install Tensorflow and TF-slim :
    tensorflow : https://www.tensorflow.org/install/
    TF-slim : https://github.com/tensorflow/models/tree/master/slim
readme.md chapters « Installing latest version of TF-Slim » and « Installing the TF-Slim image models library »

1) group the samples under a same directory. Do a sub directory for each class

example : /home/workspace/dataset/pedestrian_photos/class1
  /home/workspace/dataset/pedestrian_photos/class2

2) Convert the dataset to tensorflow records. Launch from the directory where TF-slim is installed (e.g : /home/workspace/models/slim) :

```
python ./datasets/convert_to_records.py /home/workspace/dataset/
```

multiple .tfrecord files will be created under the given « dataset » directory

3) Do a copy of each of these files and rename it from « pedestrian_train_0000*-of-00005.tfrecord » to « train-0000*-of-00005.tfrecord ». Do the same for validation records.

4) Replace the « train_image_classifier.py » from slim directory by the one pulled from the git repository

Launch (from slim directory) fine-tuning of the last classification layer :

```
python train_image_classifier.py
--train_dir=/home/workspace/output/directory/to_store/graph
--dataset_dir=/home/workspace/dataset/
--model_name=inception_v4
--checkpoint_path=/home/workspace/checkpoints/inception_v4.ckpt
--checkpoint_exclude_scopes=InceptionV4/Logits,InceptionV4/AuxLogits
--trainable_scopes=InceptionV4/Logits,InceptionV4/AuxLogits
```

The --model_name can be whatever you want, all models available for fine-tuning with tf-slim are listed in the readme.md here : https://github.com/tensorflow/models/tree/master/slim chapter « Pre-trained models »

The --checkpoint_path has to match the given model name, tensorflow checkpoints can be downloaded at the same address as above

--checkpoint_exclude_scopes indicates which layers we do not want to load when reading the checkpoint (here it corresponds the last classification layer)

--trainable_scopes indicates the layers we want to train, weights for all other layers will remain the same as in the checkpoint.