Step by Step guide to run the object detection

If you want to run the object detection model using my prepared configurations, please do the following steps. However, if you want to train object detection models on your own datasets you should take a look at my Lab report called "NLP_report_Samin_Payro.pdf". The python scripts mentioned in section 2.1 of this report are saved in directory "scripts".

If you are new to object detection, I really recommend that you read my lab report or any other tutorial on web (https://medium.com/google-cloud/object-detection-tensorflow-and-google-cloud-platform-72e0a3f3bdd6)to perceive the task. If you only want to run my work please do the following steps. As you might face errors or something, do not hesitate to send me an email at saminpayro@gmail.com if the error was unclear for you, so that I could help you.

First lets take a look at the files I uploaded:

What is inside requirements directory:

1. Data: training data
2. Fine_tune_model folders: The trained model. You use the frozen_inference_graph.pb files to test your images.
3. *.config files: configuration files of models which you should change the number of classes, the number of steps, the path to initial checkpoints, paths to test and train data and label maps in them.
4. export_inference_graph.py is used after training to turn your desired checkpoint into a frozen interface graph that you could use it for testing.
5. Inside models directory, you see some folders starting with "train" which contain all the check points and analysis graphs after training. You also seen 3 other folders like "fastrcnn_model" which contain the initial checkpoints of models.

Now let's go to step by step object detection:

1. Pre-requirments: Python 3.6 and python 2.7
2. Install Tensorflow library using pip3
3. Find the directory of your python 3.6 and go to this path "Python36\Lib\site-packages\tensorflow"
4. Retrieve the object detection API by running:
5. "git clone https://github.com/tensorflow/models"
6. Install required libraries mentioned here: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/installation.md
7. Now if you want to run on cloud go to this path in your computer: "Python36\Lib\site-packages\tensorflow\models\research"
   And make the following changes in the required files:
   - Replace setup.py with the same file I putted in scripts library.
   - In object_detection directory, line 184 of evaluator.py, change

     ```
     tf.train.get_or_create_global_step()
     ```
     to
     ```
     tf.contrib.framework.get_or_create_global_step()
     ```

- In “samples\cloud” change the runtime version to 1.2, at the time I am writing this guide version 1.4 is released but it has some defects which leads to errors when you run Faster R-CNN and RFCN models.
- In object_detection/utils/visualization_utils.py, line 24 (before import matplotlib.pyplot as plt) add:

```
import matplotlib
matplotlib.use('agg')
```

- Finally, in line 103 of object_detection/builders/optimizer_builder.py, change

```
tf.train.get_or_create_global_step()
```

to

```
tf.contrib.framework.get_or_create_global_step()
```

8.
```
# From tensorflow/models/research/
protoc object_detection/protos/*.proto --python_out=.
```

9.
```
You should adds these paths to your system environment:
“C:\Program Files\Python36\Lib\site-packages\tensorflow\models”
“C:\Program Files\Python36\Lib\site-packages\tensorflow\models\research”
“C:\Program Files\Python36\Lib\site-packages\tensorflow\models\research\slim”
“C:\Program Files\Python36\Lib\site-
packages\tensorflow\models\research\object_detection”
```

   Easily, add a new system variable called PYTHONPATH, add these paths to it, go to the “path” in your system variables and add “%PYTHONPATH%” to it.
10. Test the installations: python object_detection/builders/model_builder_test.py
11. Set up the GCP project and bucket (name it something memorable) — and it should create ‘your first project’. You should do all the things mentioned in this link: https://cloud.google.com/solutions/creating-object-detection-application-tensorflow step by step. Then in your google cloud platform environment, from the storage in left hand nav bar, you could make a new bucket with a unique name.
12. Enable Machine Learning (ML) Engine API’s — should be in the left hand nav bar of your project
13. Install the Google Cloud SDK on your local machine, and do the authentication and choose your project by the id of project you made in step 11.
14. In model config files in “requirements” directory in current place you are in github, you should replace “object_detection_bucket_samin” with the name of your own bucket. Then, In your bucket upload all the folders included in the directory “requirements”.
    If you want to run the training locally instead of on google cloud, save them into a new folder in:
    “Python36\Lib\site-packages\tensorflow\models\research\object_detection”
    Again do not forget to change the paths in model config files, to the local paths to train and test records (They are saved in “data” directory that you downloded).
15. Package the object detection config files on you local machine from the path:
    “C:\Program Files\Python36\Lib\site-packages\tensorflow\models\research”
    By running:
    python setup.py sdist
    cd slim
    python setup.py sdist

16. To train a model on cloud, go to :
    "C:\Program Files\Python36\Lib\site-packages\tensorflow\models\research"
    Open git bash, run the following commands, but you should change them as you wish:

```
gcloud ml-engine jobs submit training `whoami`_object_detection_eval_`date +%s`
--job-dir= gs://object_detection_bucket_samin /train \
--packages dist/object_detection-0.1.tar.gz,slim/dist/slim-0.1.tar.gz \
--module-name object_detection.train  \
--region us-central1  \
 --config object_detection/samples/cloud/cloud.yml \
--
 --train_dir=gs:// object_detection_bucket_samin /train   \
--
pipeline_config_path=gs://object_detection_bucket_samin/data/faster_rcnn_incepti
on_v2_coco.config  \
```

    The job-dir and train-dir indicate the directory of training. If you do not want to train any model, and you just want to observe the plots and histograms of my training jobs you could do so, in this manner you should open the google cloud shell as shown bellow from the nav of your google cloud platform:

And run: tensorboard --logdir=gs://${GCP_BUCKET_NAME} --port=8080

If you want to train a model locally, run:

```
python object_detection/train.py \
    --logtostderr \
    --pipeline_config_path=${PATH_TO_YOUR_PIPELINE_CONFIG} \
    --train_dir=${PATH_TO_TRAIN_DIR}
```

Where path to pipeline config is the path to one of the model config files in "requirments" you downloaded previously (faster_rcnn_inception_v2_coco.config or anyone else).

You can't run tensorboar on windows to see the plots and analysis of your training job. But if you are not on windows: `tensorboard --logdir=${PATH_TO_MODEL_DIRECTORY}`

17. After training finished you should turn your desired checkpoint of model into a frozen interface graph to use it for testing on test images. So, go to the path that you have your "export_inference_graph.py" (it is included in requirements directory you downloaded) and you should run this command:

py -3 export_inference_graph.py --input_type image_tensor --pipeline_config_path ./ssd_mobilenet_v1_coco.config --trained_checkpoint_prefix ./models/ train5_rfcn /model.ckpt-100004 --output_directory ./fine_tuned_model_5classes

Do not forget to change the paths.

18. To test your images, you could use a prepared file called "object_detection_tutorial.ipynb" in the path "Python36\Lib\site-packages\tensorflow\models\research\object_detection". You should open this file in Jupyter and follow the instructions in commands. You should change some paths to frozen file etc. It's straight forward.