Sources For data (Labelled) -> Oxford IIT pets dataset <https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/running_pets.md>

Source for tutorial - <https://pythonprogramming.net/introduction-use-tensorflow-object-detection-api-tutorial/> ( I highly recommend to watch this series while doing the below steps)

<https://becominghuman.ai/tensorflow-object-detection-api-tutorial-training-and-evaluating-custom-object-detector-ed2594afcf73>

Follow tensorflow object detection api tutorial.

* Problems Faced and general steps:-

1. Install protobuf 3.4
2. Install PyQt5 from <https://www.riverbankcomputing.com/static/Docs/PyQt5/installation.html>

Also add it to path

1. In the models/research folder of the tflow object detection API (or object\_detection folder of this repo), run python setup.py install to install object-detection=0.1
2. Convert the labelled dataset (which contains of images and their labellings in xml files) into a csv file first. This csv file is obtained by passing all xml labellings of the dataset of images that we have.
3. Convert csv file to record file using a simple script.
4. Go to <https://github.com/tensorflow/models/tree/master/research/object_detection/samples/configs>, to get the desired model and it’s config file , which we want to train. It is advisable to use a pre trained model from the list of models given in the list. We can choose a slower running model which is more accurate, as our input data is static.

**Note**: For this project I have already converted all the images from the “images” folder to record files stored in “**object\_detection\legacy\data\**” folder

1. Make necessary changes to the config file of the chosen model. I have chosen the “**ssd\_mobilenet\_v1\_coco\_11\_06\_2017**” model (in config file “**ssd\_mobilenet\_v1\_pets.config** “ change statements like path to model, number of classes (2 in our case), batch size, etc, to suit our intentions).
2. Copy paste all the training and testing images, along with the model u have chosen and its config file to object-detection/legacy folder which also has train.py.

Make sure to also copy contents of object-detection/slim to the legacy folder to avoid errors.

1. Run train.py to train the model -> python train.py --logtostderr --train\_dir=training/ --pipeline\_config\_path=training/ssd\_mobilenet\_v1\_pets.config

In the above command , logtosterr saves the error, train\_dir stores all the training related files like meta, model files, tensorboard related info, etc.

1. From models/object\_detection, via terminal, you start TensorBoard with: tensorboard --logdir=’ C:\Acer\SCU Docs\Spring 2019 Quarter\Cloud Computing - COEN 241\Cloud\_Computing\_Project\models-master\research\object\_detection\legacy\training\ ' This runs on 127.0.0.1:6006 (visit in your browser) The file “events.out.tfevents.1555538636.LAPTOP-25TS0BEL”, records the training process which we can then see on tensorboard

Note: To resolve connection issues goto -> <https://stackoverflow.com/questions/40106949/unable-to-open-tensorboard-in-browser>

1. After sufficient training, extract the inference graph by running the script ->

python export\_inference\_graph \

--input\_type image\_tensor \

--pipeline\_config\_path path/to/ssd\_inception\_v2.config \

--trained\_checkpoint\_prefix path/to/model.ckpt \

--output\_directory path/to/exported\_model\_directory

12) In order to evaluate the model, use eval.py and run the following command in object-detection :-

python eval.py --logtostderr --pipeline\_config\_path=training/ssd\_mobilenet\_v1\_coco.config --checkpoint\_dir=training --eval\_dir=eval/

NOTE:- u will need to install pycocotools (coco api) to run above command^

Install Visual C++

Follow this link - <https://github.com/philferriere/cocoapi>

I cloned the above repo, and then ran a package called 2p3 to convert all python2 files in PythonAPI folder, to python3. Then just simply do python setup.py install.

Some errors I faced ->

invalid numeric argument '/Wno-cpp'  - <https://github.com/cocodataset/cocoapi/issues/51>

LINK : fatal error LNK1158: cannot run 'rc.exe' - <https://stackoverflow.com/questions/14372706/visual-studio-cant-build-due-to-rc-exe>

Slides Presentaion:-

<https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a>

<https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/#19ec3aa18d4b>

<https://github.com/tensorflow/models/tree/master/research/object_detection>

<https://github.com/kubeflow/examples/tree/master/object_detection>