**DOCUMENTATION-STEP BY STEP:**

*OBJECT DETECTION MODEL:*

Shailesh Nanisetty

1. Firstly, Collect Images and its Pascal VOC XML annotations. Then, randomly distribute them into 2 folders – Train and Test in the ratio 8:2. Put these into 2 folders in images folder in: *C:\Object\_detection\models-master\research\object\_detection.*
2. Open Command Prompt in windows and set path to *C:\Object\_detection\models-master\research\object\_detection.*
3. Install all the required dependencies on your working environment (or) instead create a virtual environment with name ‘your\_name’ and install required things in it .We can activate virtual environment using the command ‘activate your\_name’.
4. While annotating, all the meter images should have height, width, x, y, w and h as in the pascal VOC format whereas w.r.t Non-meter and spoof images’ annotations should have height and width in the same format as the meter ones but x, y, w and h must be assigned = -1.
5. Check the attached modify\_xml.py to understand how to modify annotations for non-meter and spoof images.
6. After finishing annotations, run the command *python xml\_to\_csv.py*  (or) *xml\_to\_csv.py* depending on Ur python version on command prompt. This will convert image info in xml annotations to csv file format.
7. Check your both train and test csv files created in images folder in the path mentioned in (2). Make sure that the values in the ‘label column’ are having the same lower case value i.e. ‘display’ and all the empty cells should be replaced by -1 (as these correspond to the coordinates of non-meter and spoof images).
8. Next, We need to generate 2 files: *train.record* and *test.record* .These are must as we are working on tensorflow (Tf.2x version) platform.
9. Generate tfrecord file for train set by this command: *python generate\_tfrecord.py --csv\_input=images/train\_labels.csv --image\_dir=images/train --output\_path=train.record*  and Generate tfrecord file for testing by this command: *python generate\_tfrecord.py --csv\_input=images/test\_labels.csv --image\_dir=images/test --output\_path=test.record* .
10. Add the train.record and test.record files into the ‘Tf2-Object-detection-folder/object\_detection’ path either in the jupyter notebook or in the google colaboratory. These are the training and test sets respectively.
11. For training the model after adding images again and again, steps (6) to (10) has to be followed.
12. Then open either the Jupiter notebook or the google colaboratory in whichever you’re working.
13. Start running the cells and to modify the hyperparameters of the SSD Mobilenet 320x320 model, open Tf2-Object-detection-folder/object\_detection/training/SSD mobilenet 320x320.config and change accordingly.
14. DESCRIPTION OF THE CONTENTS IN THE PYTHON NOTEBOOK:

* Model\_dir=place where checkpoints of the model gets updated. It’s located at SSDnew.
* Output\_directory=place where model gets saved .It’s located as SSDmobileoutput.

Rest all are direct, just run the cells directly in the notebook and we’ll get the results.

Tf2-Object-Detection-Folder=to be used in jupyter notebook or in a google colaboratory.

Object\_detection.zip=Used for all steps before running Jupyter notebook/google colaboratory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_