1. First label your data with designated objects.

Note -- All of these instructions are should be run from a root folder [in this case, CTG_ObjectDetection]

2. After labeling, please distribute the labeled images in train and test in 90:10 ratio and move these 2 folders inside the images folder.

3. You will find one xml_to_csv.py in CTG_ObjectDetection directory which will change the xml files to csv files present in train and test directory. [You must have 2 new empty folders called data and training folder before doing this].

Run: python3 xml_to_csv.py

4. Now, you will have generate_tfrecord_train.py and generate_tfrecord_test.py. Make these changes if necessary-

(i) First change the number of labels if necessary in class_text_to_int function.

Now, run these commands:

```
python generate_tfrecord_train.py --csv_input=data/train_labels.csv  --output_path=data/train.record

python generate_tfrecord_test.py --csv_input=data/test_labels.csv  --output_path=data/test.record
```

After this command, you will have your tfrecords files.

Note: - These commands will assume that you have eveyrthing in place along with object detection API.

5. Now, download the model and its respective configuration file from

Model: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

Config: https://github.com/tensorflow/models/tree/master/research/object_detection/samples/configs

6. Make the necessary changes in config files like num_classes, batch size and move into training folder. Replace the bottom of the file with this section:

```
train_input_reader: {
  tf_record_input_reader {
    input_path: "data/train.record"
  }
  label_map_path: "data/object-detection.pbtxt"
}

eval_config: {
  num_examples: 2000
  # Note: The below line limits the evaluation process to 10 evaluations.
  # Remove the below line to evaluate indefinitely.
  max_evals: 10
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "data/test.record"
  }
  label_map_path: "data/object-detection.pbtxt"
  shuffle: false
  num_readers: 1
}
```

You also need to set path to your model for "fine_tune_checkpoint" variable as this step will take the pre-existing model as its starting checkpoint.

PATH_TO_BE_CONFIGURED points to the directory with the model files ie ssd_mobilenet_v1_coco_11_06_2017

7. Now, Move the necessary folders like data, images, training into object_detection inside models/research folder.

8. Start the training by running this command:

```
			python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/model_config
```

```
			April 30, try: python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/ssd_mobilenet_v1_coco.config
```

9. Exporting Graph

```
python export_inference_graph.py --input_type image_tensor --
pipeline_config_path=training/ssd_mobilenet_v1_coco.config --
trained_checkpoint_prefix=training/model.ckpt-11082 --output_directory output_inference_graph.pb
```

Wait for upto 8 to 10 k iterations.

10. Start running the jupyter notebook from models/research/object_detection directory and open object_detection_tutorial.ipynb. You can make a copy of existing notebook to make any changes.

Note: If you find any difficulty then go to https://www.youtube.com/watch?v=COlbP62-B-U&index=1&list=PLQVvvaa0QuDcNK5GeCQnxYnSSaar2tpku