**INTERNSHIP REPORT**

***Submitted by***

***R.T. Rishi Varman,***

***Thiagarajar College of Engineering, Madurai-625015***

**GSC Inc,Thiruchirapalli**

***Nov 25 – Dec 14, 2019***

**Abstract**

**Topic:Automatic License Plate Recognition(ALPR)**

Vehicles are widely used in all areas of production as well

as in our daily lives. Therefore, the vehicle License Plate (LP)

number is an efficient way to identify vehicles, which is unique

for each vehicle. With the rapidly increasing number of

vehicles, traffic violations appear more frequently in public

traffic, such as fraud tolls in highways or parking, speeding,

theft of cars, etc. Therefore, it is necessary to identify the LPs

of vehicles for safety. The extracted information from an LP

can be used for several purposes,

Automatic license plate recognition (ALPR) is used in many domains, such as parking, border control and motorway road tolling. Its importance has increased the recent years, with many new applications. Therefore, it is necessary to identify the LPs of vehicles for safety. The extracted information from an LPs can be used for several purposes, like access and flow control, monitoring border crossings and highway toll stations,searching for suspicious vehicles or even fighting crime, etc High prediction accuracy and speed of ALPR is vital. Our objective is to create shared library file which is used run a C/C++ Application which detects the licence plate in a video/image and stores the number. To fulfil our goal we use DEEP LEARNING methods. The library we use to train the model is the Tensorflow Object Dedection API.

Vehicles are widely used in all areas of production as well

as in our daily lives. Therefore, the vehicle License Plate (LP)

number is an efficient way to identify vehicles, which is unique

for each vehicle. With the rapidly increasing number of

vehicles, traffic violations appear more frequently in public

traffic, such as fraud tolls in highways or parking, speeding,

theft of cars, etc. Therefore, it is necessary to identify the LPs

of vehicles for safety. The extracted information from an LP

can be used for several purposes, like access and flow control,

monitoring border crossings and highway toll stations,

searching for suspicious vehicles or even fighting crime, etc

Vehicles are widely used in all areas of production as well

as in our daily lives. Therefore, the vehicle License Plate (LP)

number is an efficient way to identify vehicles, which is unique

for each vehicle. With the rapidly increasing number of

vehicles, traffic violations appear more frequently in public

traffic, such as fraud tolls in highways or parking, speeding,

theft of cars, etc. Therefore, it is necessary to identify the LPs

of vehicles for safety. The extracted information from an LP

can be used for several purposes, like access and flow control,

monitoring border crossings and highway toll stations,

searching for suspicious vehicles or even fighting crime, etc

**Introduction**

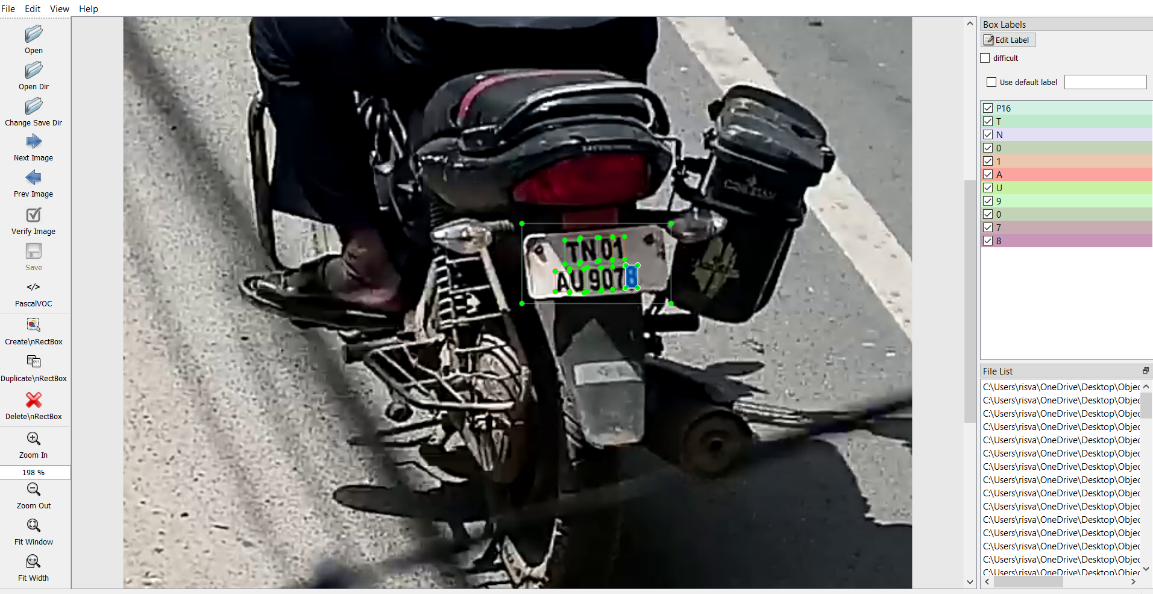
Vehicles are widely used in all areas of production as well as in our daily lives. Therefore, the vehicle License Plate (LP) number is an efficient way to identify vehicles, which is unique for each vehicle. With the rapidly increasing number of vehicles, traffic violations appear more frequently in public traffic , such as fraud tolls in highways or parking, speeding, theft of cars, etc. it is essential to create an Lisence Plate recognition application.

**Requirements analysis and specification:**

* A Protobuf Compiler
* Windows/LINUX platform
* Tensorflow library – To train the model
* Vehicle images(Splitted as Test and Train)
* Anoconda – To create the Python Environment
* System with a good GPU support – For training the model faster
* Bazel – To build a dynamic Tensorflow Library

**Methodology and steps involved:**

**Ground Truth Extraction:**

* The License plate and the numbers in it of the vehicle images are labeled using a tool LabelImg.exe and stored as an xml file.
* The xml file and is coverted to a .csv file using a Python Program containing the position of all the labels in all the images (separately for both test and train images) .They csv files for both test and train images are then conveted to a RECORD FILE using a(Tesnsorflow records) which is a binary data file of the labelled images.They are used during the training process.

**Training the Framework(model)**

* We use a pretrained model ssd\_mobilenet\_v1\_coco instead of creating our own model architecture as the training speed and accuracy is better in the pretrained model.
* The model is configured based on our needs:

They are:

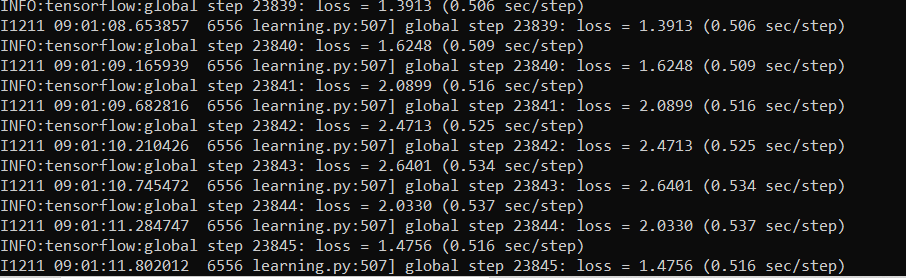
1.The number of classes are changed to 39(A-Z,0-9,’-’,’.’)

2.The checkpoint of the model is changed to the correct path.

3.The train records and the test records path are set.

4.The test images count are modifed

* The training is then started by running train.py from the object detection folder as well as specyfying the config file directory.

****

* The inference graph is exported from the model checkpoint by running export\_inferencegraph.py.

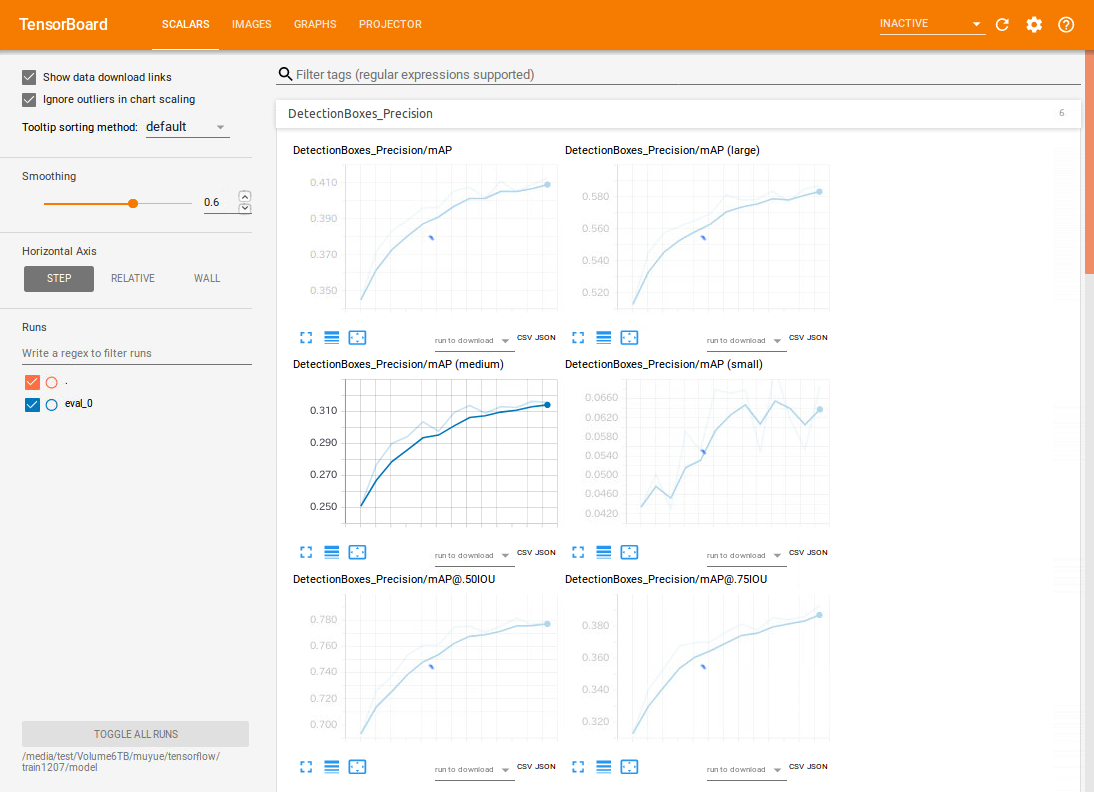
Therefore we get the .pb file which contains our trained framework.

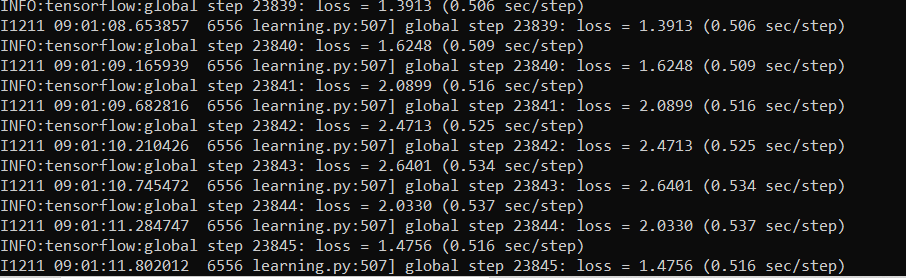
**Building the shared Library:**

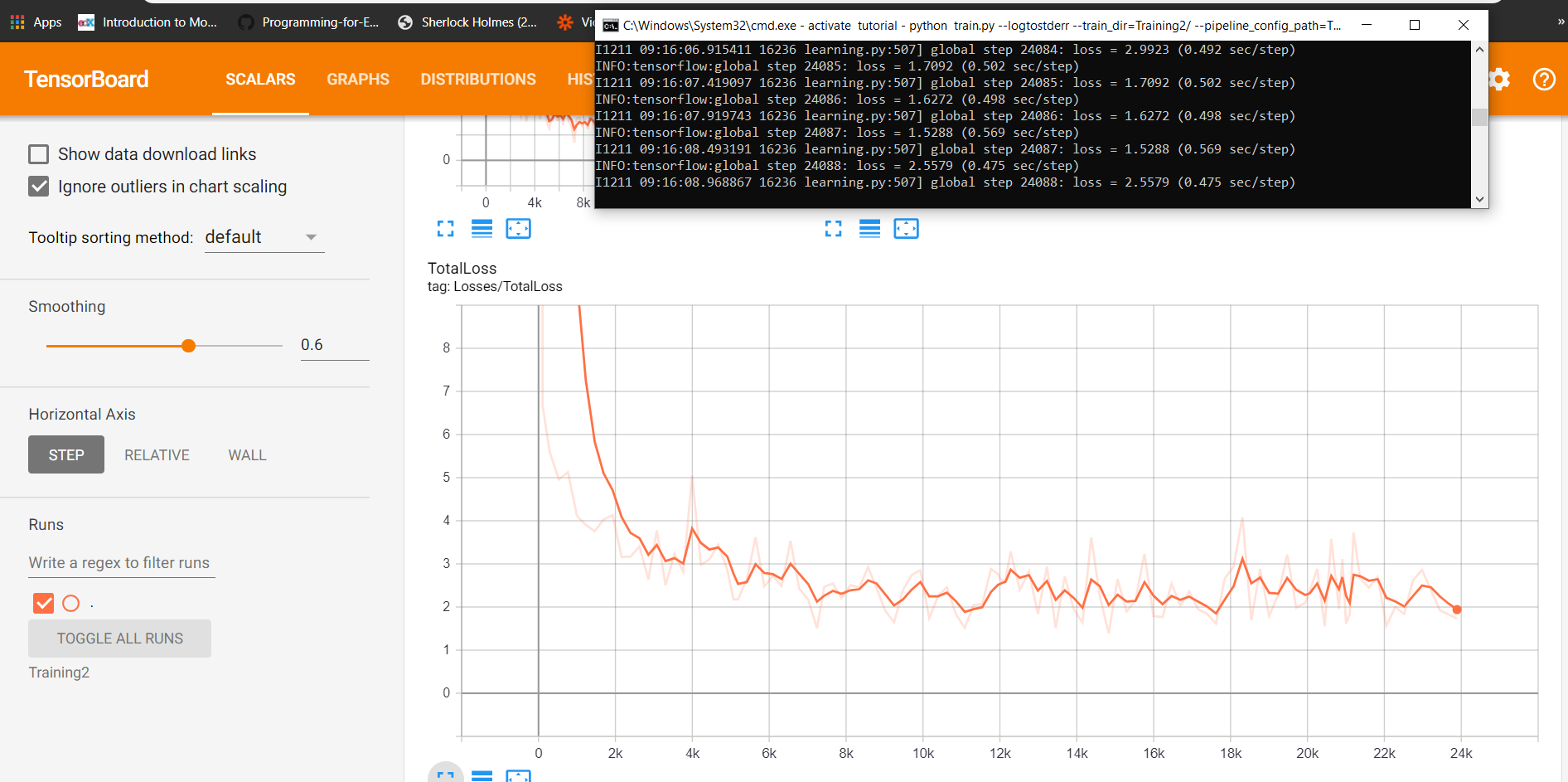
* Bazel tool is used to create the tensorflow shared library file.
* After the .so(shared library) file is built the libraries are used in the CPP Program in which the inference graph file is imported and implemented to detect the Lisence Plate in any image/video and store the results.

**My Progress:**

Training 1:

I trained the framework using the above methodology with 90 train and 10 test image by getting a Precesion value of 58 percent

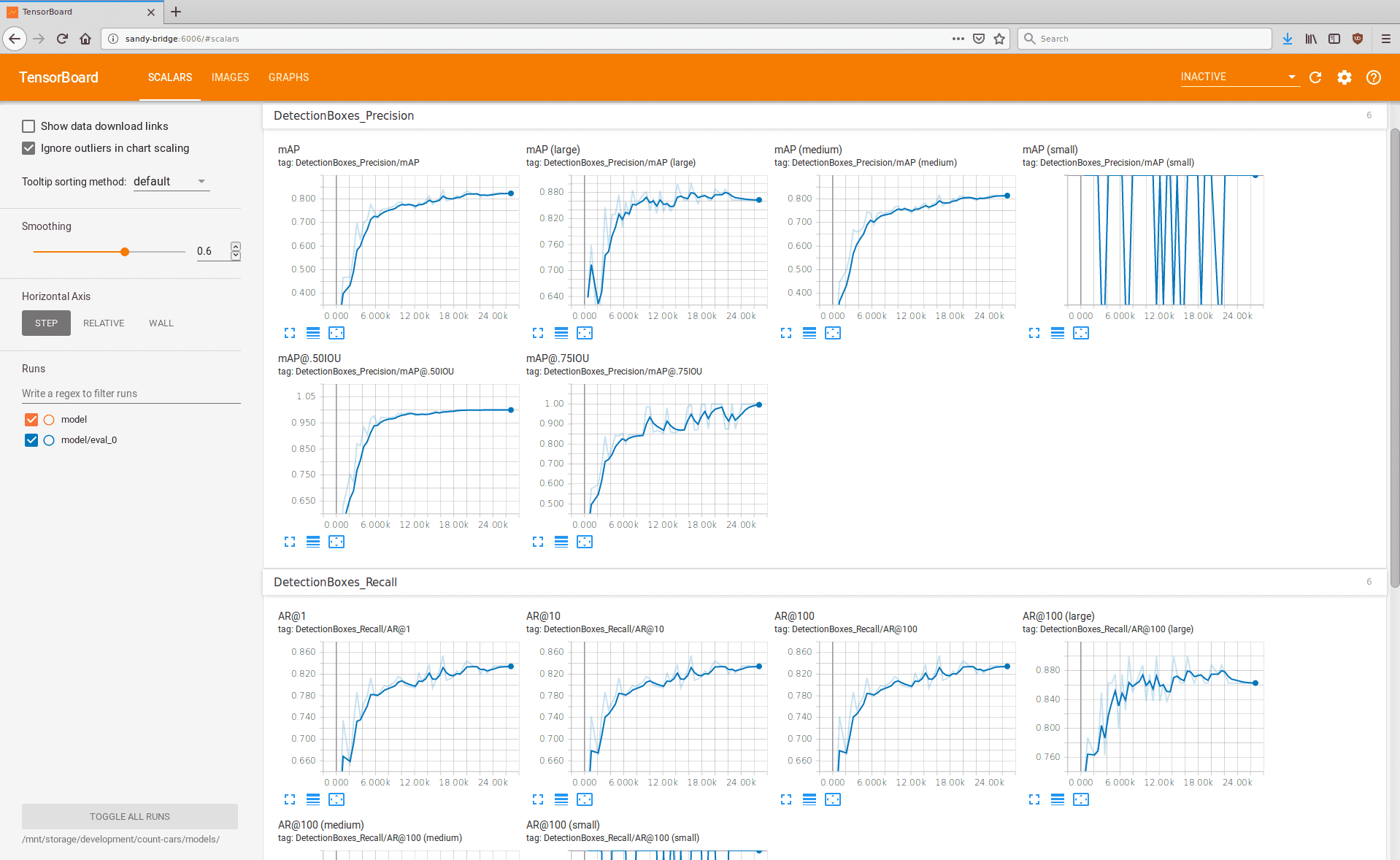
The total lass value was 1.47

The total loss graph was less smoother.

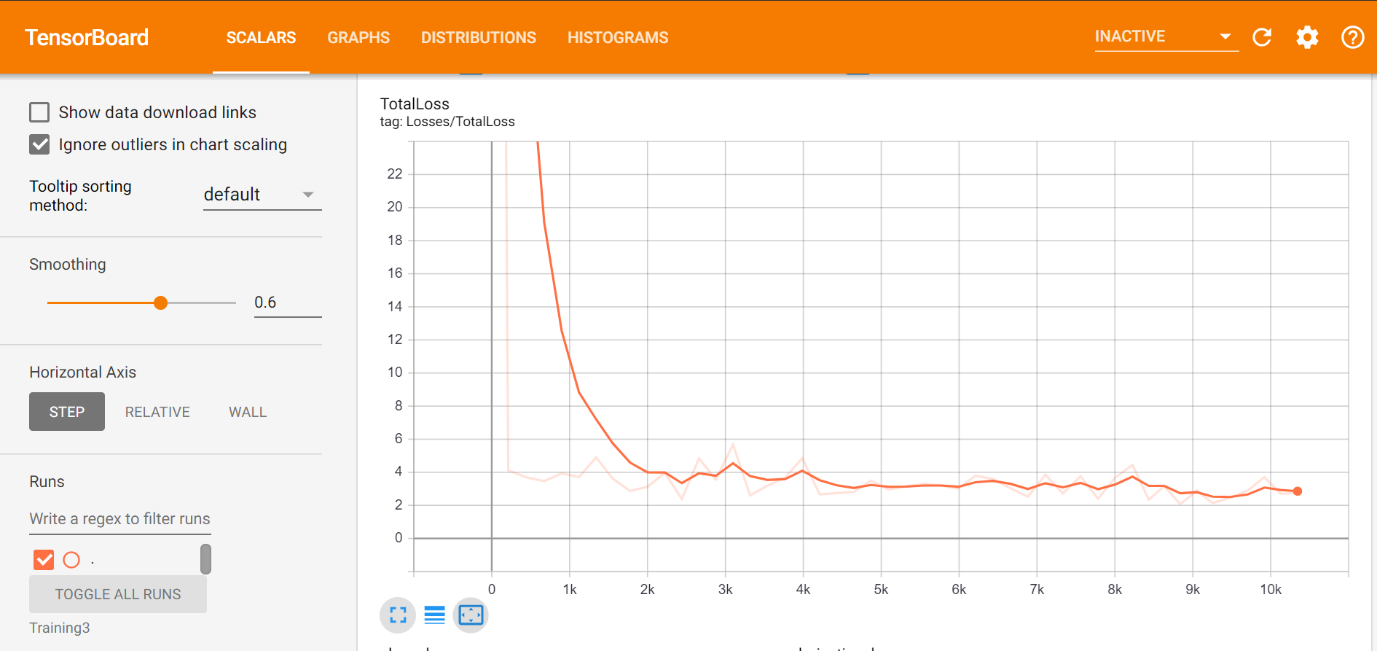
TRAINING 2:

I trained the framework again with an additional 500 images and exported the inference graph for the trained model.This can be easily done by changing the checkpoint file directory in the config file by replacing with the last chackpoint of the training.

The precision was much better.It was 79 percent and the total loss graph was much smoother.

******

***Precision***

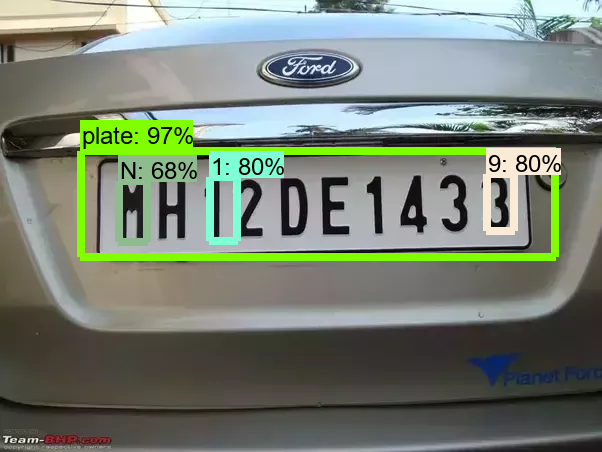
******

***TotalLoss***

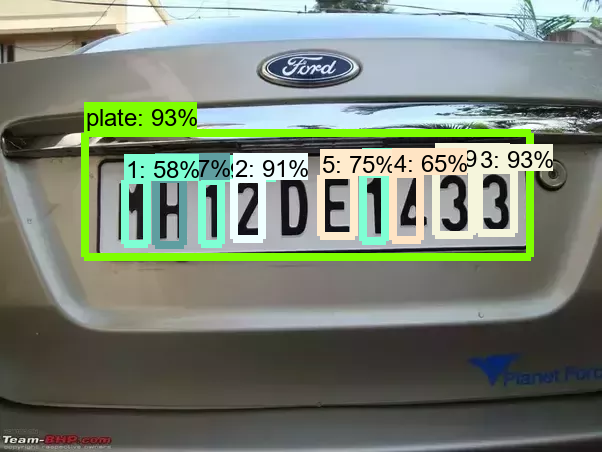
By this way we can retrain the framework an many times we collect extra images.

* I tested the framework using a python program on new images.The results were pretty good.

***Using the first Inference Graph***

******

***Using the second Inference Graph***

******

**Issues faced:**

I created the a cpp applcation for our objective.

The issue which I faced is, the tensorflow shared library build using bazel is not getting build.I tried many ways in facing the issues by trying with other packages like VCPKG.I progressed much better by installing the correct bazel version for the tensorflow version.The errors got reduced.But the dynamic Library(.so) Build is not getting completed. I need to solve the compile time errors and properly build the .so Library.

**Conclusion:**

Although some studies have been

conducted on LP detection and recognition, this research is

different from previous work studies because we have used a

deep learning architecture represented by a CNN model in

both LP detection and recognition.

Although some studies have been conducted on LP detection and recognition, this project is different from previous work studies because we have used a deep learning architecture represented by a CNN model in both LP detection and recognition. Future work will focus on the accuracy rate of improvement in the detection and recognition by adding on hidden layers in the CNN architecture.In the C++ implementation the Tensorflow Library must be properly builded.

***REFERENCES:***

THE WHOLE METHODOLODY-<https://pythonprogramming.net/introduction-use-tensorflow-object-detection-api-tutorial/>-

<https://medium.com/@WuStangDan/step-by-step-tensorflow-object-detection-api-tutorial-part-1-selecting-a-model-a02b6aabe39e>

Guide for installing the packages and commands-<https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/install.html#general-remarks>

Research paper: <https://arxiv.org/pdf/1506.02640.pdf>

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

Installing CUDA and setting the TF environment-<https://www.youtube.com/watch?v=KZFn0dvPZUQ>

Installing BAZEL: <https://docs.bazel.build/versions/master/install-ubuntu.html>

Blog for building tensorflow Library:<http://bitbionic.com/2017/08/18/run-your-keras-models-in-c-tensorflow/>

About CNN: <https://www.udemy.com/course/machine-learning-a-ztm-full-course/?utm_source=adwords&utm_medium=udemyads&utm_campaign=DSA_Catchall_la.EN_cc.INDIA&utm_content=deal4584&utm_term=_._ag_82569850245_._ad_398023114490_._kw__._de_c_._dm__._pl__._ti_dsa-392284169515_._li_9040217_._pd__._&matchtype=b&gclid=Cj0KCQiArdLvBRCrARIsAGhB_sw5jnGwy8kHIbGRG2ZxPfYWvWgJwAKu-NzRDx2QE1tiHa53EeBdJ0gaAizWEALw_wcB>