

Object Detection in an Urban Environment

1. Project Overview

The goal of the project is to detect object in an urban environment. For this project, the waymo open dataset is used for sensor data. The waymo dataset is stored into .tfrecord files. A dataset of images of urban environments containing annotated cyclists, pedestrians and vehicles were provided. In the first task, we need to perform an Exploratory Data Analysis (EDA), like object occlusions, label distribution etc. Later we need to train a Single Shot detector, a pertained model of SSD Resnet 50 640x640 model is given. At first we need to train and evaluate the pre-trained model and then based on the results we need to improve the performance of object detection by modifying the hyper-parameters of the CNN in config files.

The modifications to the config file need to reduce the loss of model and better detection. At last step, after satisfactory results, the model will be exported and inference video is created.

2. Set Up

- i) The jupyter notebook is launched using the script ./launch_jupyter.sh
- ii) Download the pretrained model and move to the /home/workspace/experiments/pretrained_model/ folder and extract the tar.gz as described in the below steps

```
cd /home/workspace/experiments/pretrained_model/  
  
wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz  
  
tar -xvzf ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz  
  
rm -rf ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz
```

- iii) Edit the config file so a new config file is created with name **pipeline_new.config** . Move this file to the /home/workspace/experiments/reference/ .

```
cd /home/workspace/  
python edit_config.py --train_dir /home/workspace/data/train/ --  
eval_dir/home/workspace/data/val/ --batch_size 2 --  
checkpoint/home/workspace/experiments/pretrained_model/ssd_resnet50_v1_fpn_  
640x640_coco17_tpu-8/checkpoint/ckpt-0 --  
label_map/home/workspace/experiments/label_map.pbtxt
```

- iv) Training Process

```
python experiments/model_main_tf2.py --model_dir=experiments/reference/ --  
pipeline_config_path=experiments/reference/pipeline_new.config
```

Monitor the process by executing the below command.

```
python -m tensorboard.main --logdir experiments/reference/
```

Evaluation Process

```
python experiments/model_main_tf2.py --
model_dir=experiments/reference/ --
pipeline_config_path=experiments/reference/pipeline_new.config --
checkpoint_dir=experiments/reference/
```

v) Export the trained model

```
python experiments/exporter_main_v2.py --input_type image_tensor --
pipeline_config_path experiments/reference/pipeline_new.config --
trained_checkpoint_dir experiments/reference/ --output_directory
experiments/reference/exported/
```

Modify the config file and train and export the trained model when you find the results are satisfactory.

3. Dataset

- Dataset Analysis:

The dataset is divided into train, test, val and stored into respective folders. We use the train folder to do dataset analysis.

Below are some images of with its bounding boxes

The screenshot shows a Jupyter Notebook titled "Exploratory Data Analysis" with a toolbar at the top. Below the toolbar, a code cell is visible with the following text:

```
plt.show()
```

Below the code cell, there is a section titled "Display 10 images" with a description: "Using the dataset created in the second cell and the function you just coded, display 10 random images with the associated bounding boxes. You can use the methods `take` and `shuffle` on the dataset."

The code cell output shows the following code:

```
In [4]: # Display 10 random images in dataset
dataset = dataset.shuffle(30)
batch = dataset.take(10)
display_images(batch.as_numpy_iterator())
```

Below the code, two images are displayed side-by-side. Each image shows a street scene with several cars. Red bounding boxes are drawn around the cars in both images. The left image shows a street with a white car in the foreground and several other cars further down the road. The right image shows a street with a white car in the foreground and several other cars further down the road. The images are displayed with a vertical axis on the left side of each image, ranging from 200 to 500.

0.0.0.0:3002/notebooks/Exploratory Data Analysis.ipynb


Logout

jupyter Exploratory Data Analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

In [4]:
Display 10 random images in dataset
dataset = dataset.shuffle(30)
batch = dataset.take(10)
display_images(batch.as_numpy_iterator())



Additional EDA


In this last part, you are free to perform any additional analysis of the dataset. What else would like to know about the data? For example, think about data distribution.

jupyter Exploratory Data Analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

display_images(batch.as_numpy_iterator())



Additional EDA

In this last part, you are free to perform any additional analysis of the dataset. What else would like to know about the data? For example, think about data distribution. So far, you have only looked at a single file...

root@5635c4b... Exploratory Da...

display_images(batch.as_numpy_iterator())



Additional EDA

In this last part, you are free to perform any additional analysis of the dataset. What else would like to know about the data? For example, think about data distribution. So far, you have only looked at a single file...

display_images(batch.as_numpy_iterator())



Additional EDA

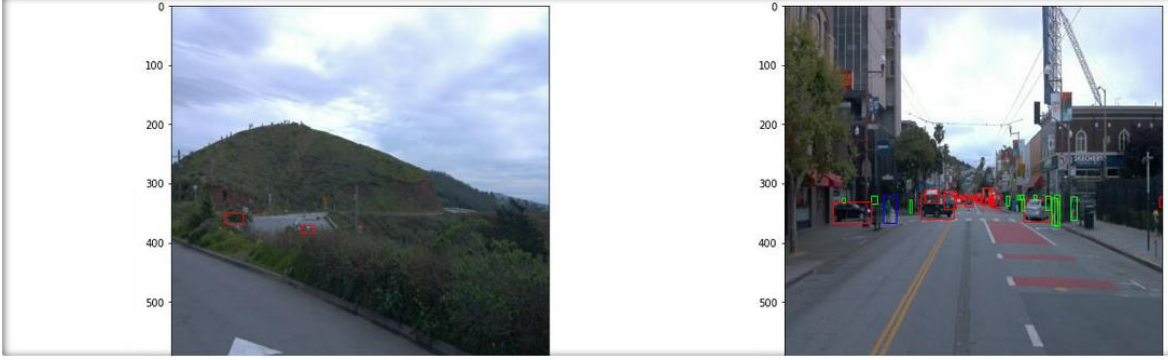
In this last part, you are free to perform any additional analysis of the dataset. What else would like to know about the data? For example, think about data distribution. So far, you have only looked at a single file...

jupyter Exploratory Data Analysis (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

the methods `take` and `shuffle` on the dataset.

```
In [6]: # Display 10 random images in dataset
dataset = dataset.shuffle(30)
batch = dataset.take(10)
display_images(batch.as_numpy_iterator())
```



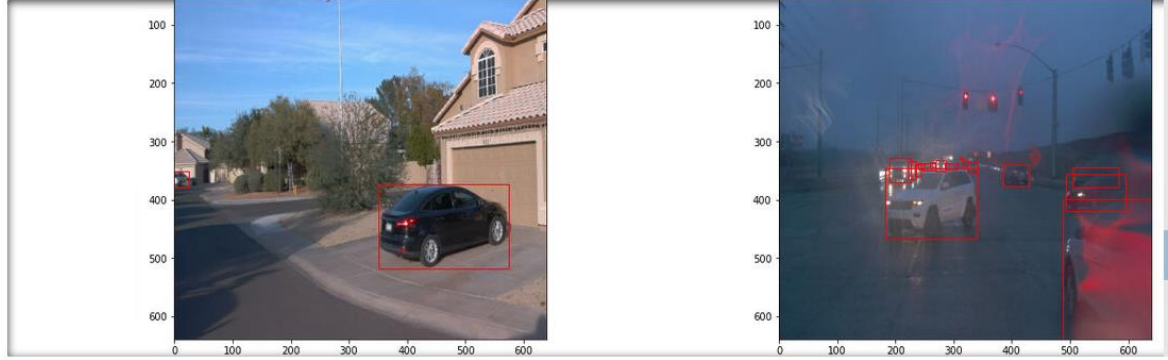
root@5635c4b... Exploratory Da...

jupyter Exploratory Data Analysis (unsaved changes) Logout

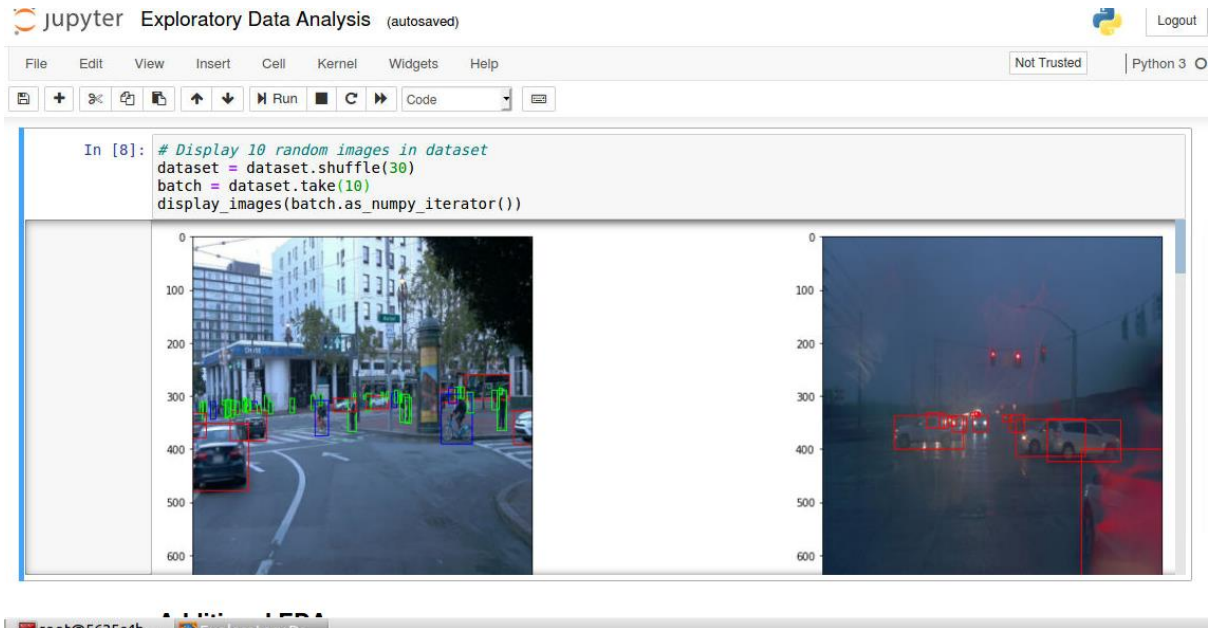
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

the methods `take` and `shuffle` on the dataset.

```
In [6]: # Display 10 random images in dataset
dataset = dataset.shuffle(30)
batch = dataset.take(10)
display_images(batch.as_numpy_iterator())
```



root@5635c4b... Exploratory Da...



The dataset consists of images with

Traffic: heavy traffic, no traffic, no detection of any vehicle

Types of roads: Highways, hill areas, Signal junctions, Residential area, Zebra crossing

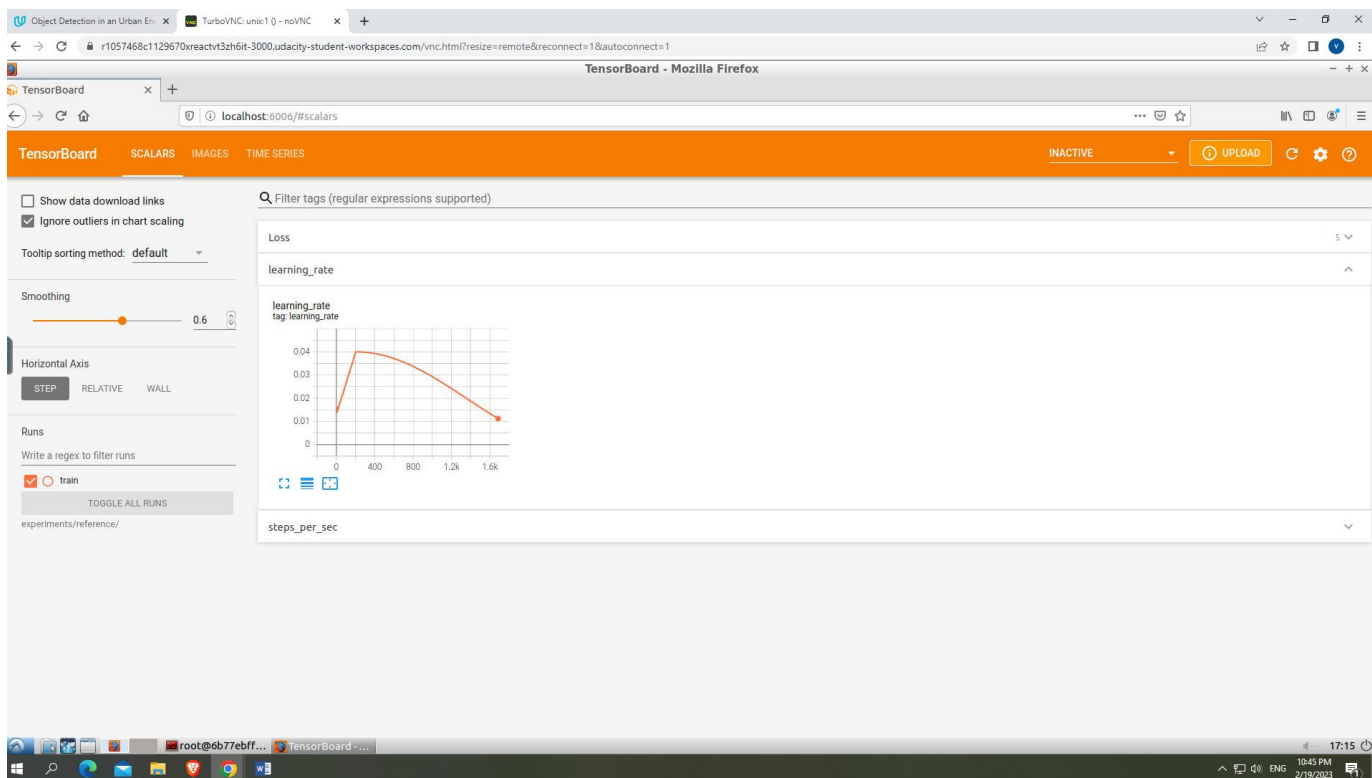
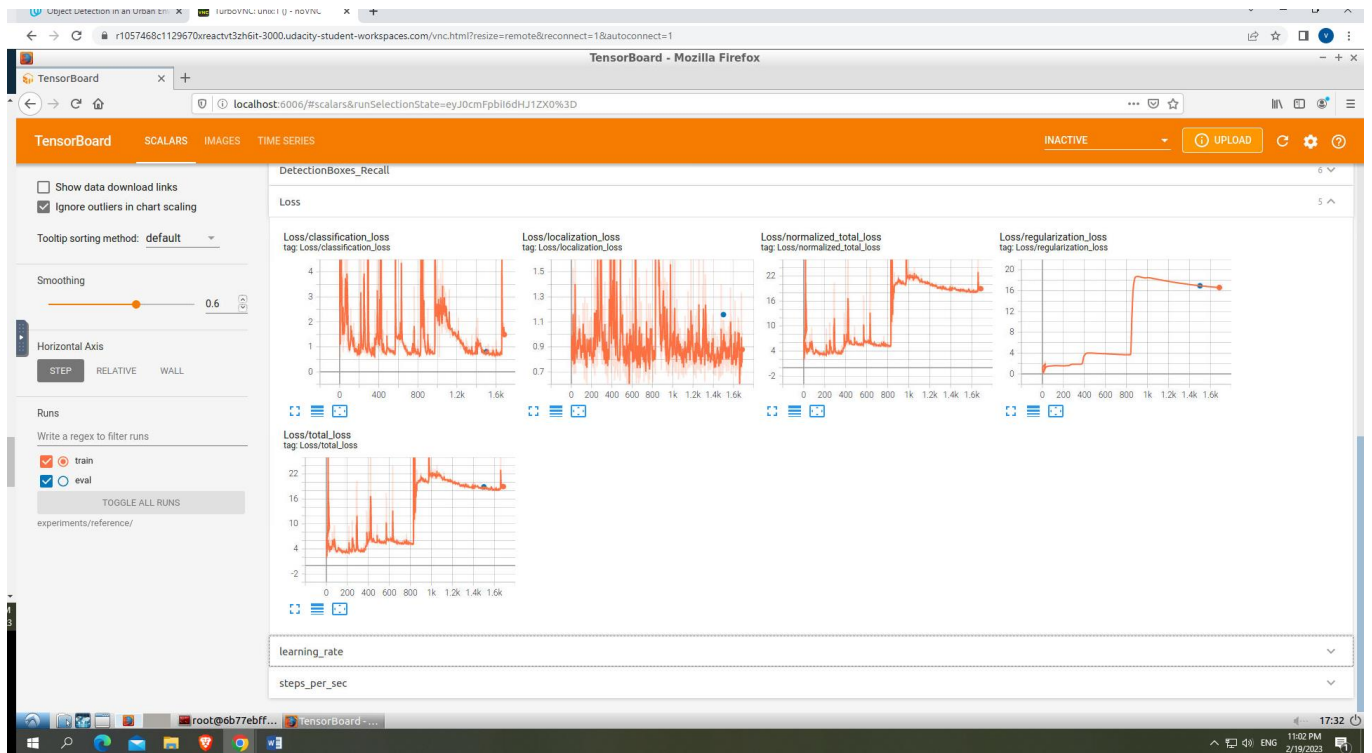
Environment: variation of sun light, Foggy, Night etc.

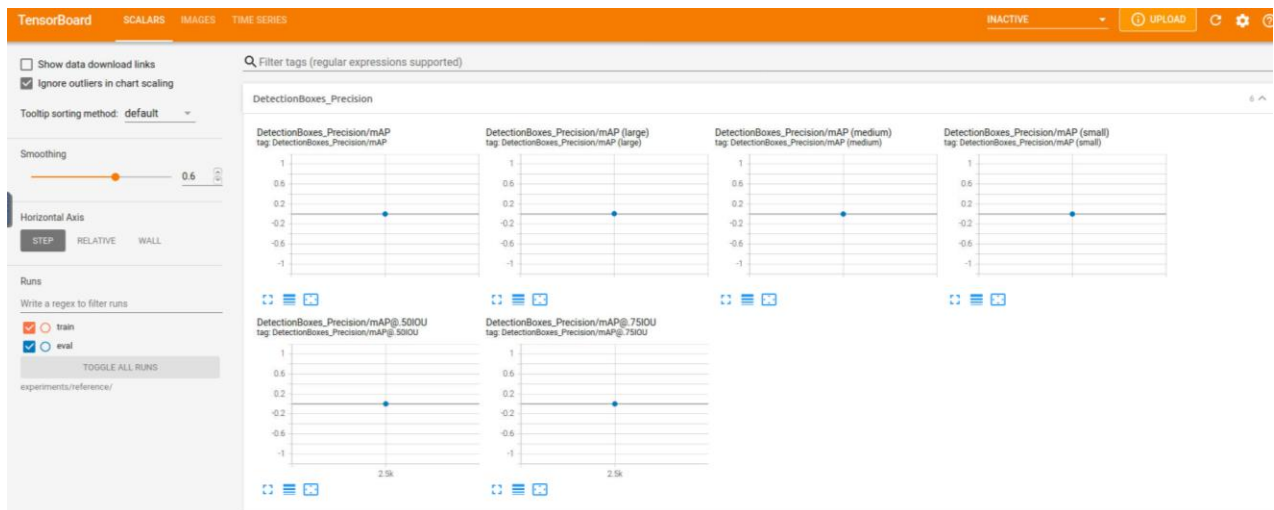
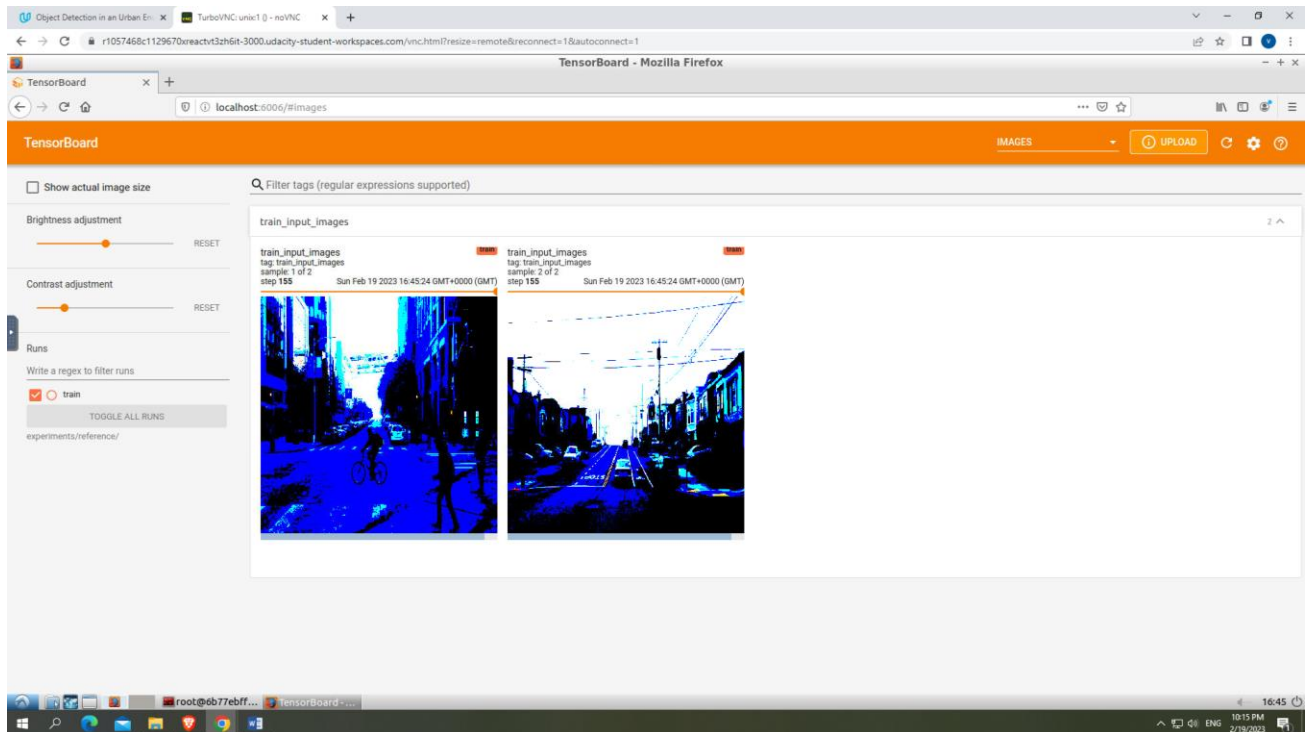
Heavy Vehicles, Light vehicles, bicycles etc.

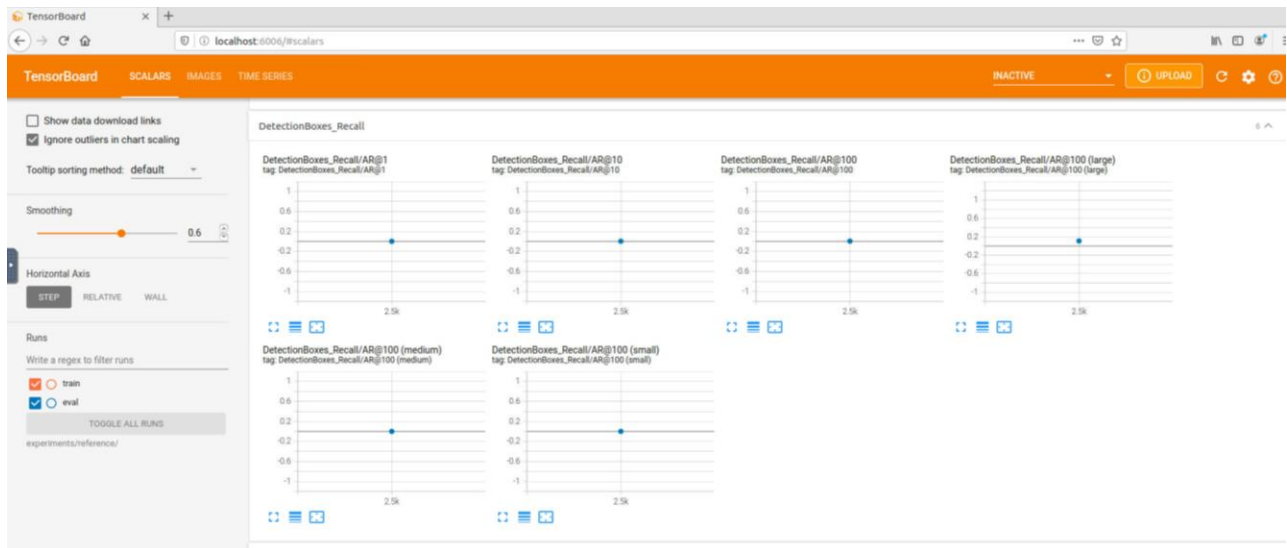
4. Training

- i) Reference Experiment: The loss and learning rate charts of the pre-trained model training and evaluation results shown below. Based on the observation, the more number of iterations, the more drop in the learning rate of the model and high increase in the loss.

The classification loss and localization loss decreased with more number of iterations. Whereas the normalized total loss, regularization loss has steep increase after 800 steps and slowly decreasing afterwards. The learning rate has been decreasing after 200 steps. The evaluation process is shown in blue dot as it ran for on epoch only.









ii) Improve on the Reference:

In the config file added the following data augmentations and decreased the learning rate. The loss is reduced ~3.

```
learning_rate {
  cosine_decay_learning_rate {
    learning_rate_base: 0.035
    total_steps: 2500
    warmup_learning_rate: 0.013333
    warmup_steps: 200
  }
}

data_augmentation_options {
  random_adjust_brightness {
    max_delta: 0.3
  }
}

data_augmentation_options {
  random_rgb_to_gray {
    probability: 0.2
  }
}

sync_replicas: true
```

