

Self Driving Car Engineer

Udacity Nanodegree

Quan Bach

Project 1

**Object Detection in an Urban
Environment**

Project Overview

The objective of this project is performing data exploratory analysis and train a single-shot detection model to detect objects in an urban environment using the Waymo dataset.

Set Up

- The Jupyter Notebook QuanBach-Exploratory Data Analysis is the deliverable for the Exploratory Data Analysis Part
- The Jupyter Notebook QuanBach_Explore_augmentations is the implementation for data augmentation

Dataset

Data Analysis

After implemented the draw bounding boxes for the ground truth classes, I have learnt that the ground truth classes are not 1: vehicles, 2: pedestrian, and 3: cyclist as the initial assumption but the correct classes are 1: vehicles, 2: pedestrian, and 4: cyclist.

Further data exploration and analysis from sampling 100,000 batches of the dataset indicate that cyclists appear significantly less in the training dataset compared to the other two. This would result in the model having problem detecting cyclists.

From the display sample images, most of the objects appear to be occluded. Therefore, adding occlusions as an data augmentation method would be beneficial for better performance.

Training

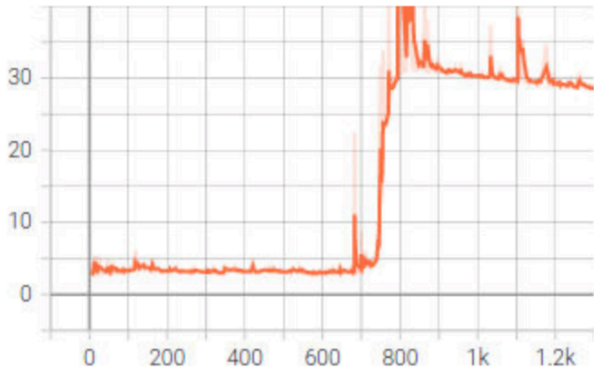
Reference Experiment

The Reference Experiment results in poor performance with the Total Loss never got below 20.

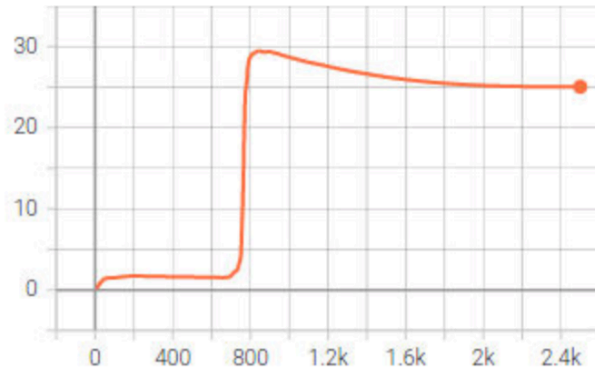


The Normalized Loss and Regulation Loss remain low for half/one-quarter of the data respectively then increase sharply around 30 and become stagnation also.

Loss/normalized_total_loss
tag: Loss/normalized_total_loss

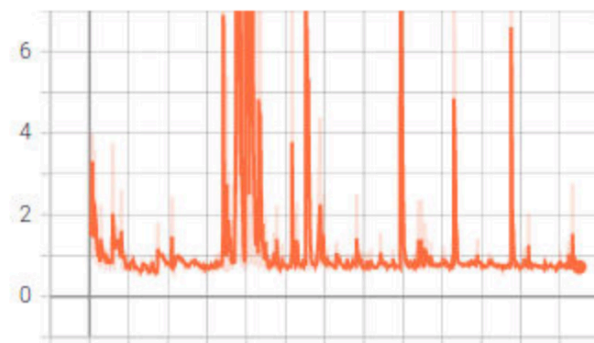


Loss/regularization_loss
tag: Loss/regularization_loss

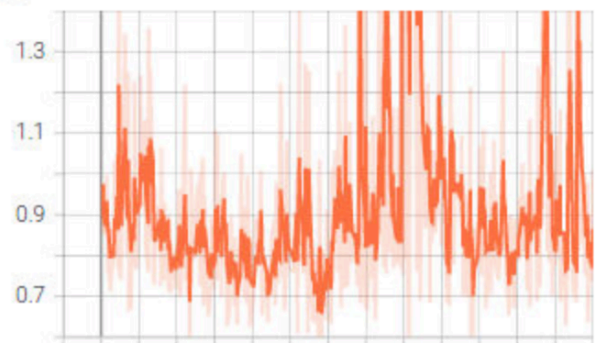


The Classification Loss and Localization Loss oscillated and never become logarithmic.

Loss/classification_loss
tag: Loss/classification_loss



Loss/localization_loss
tag: Loss/localization_loss

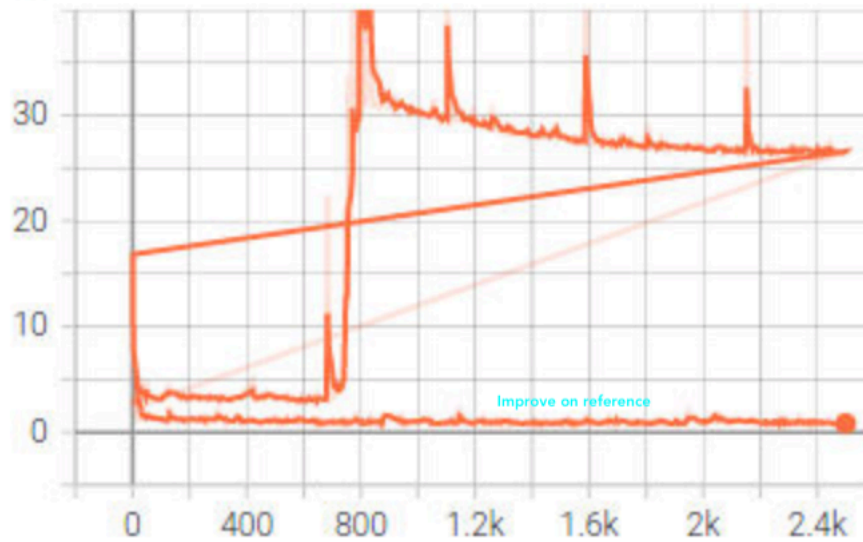


Improve On Reference Experiment

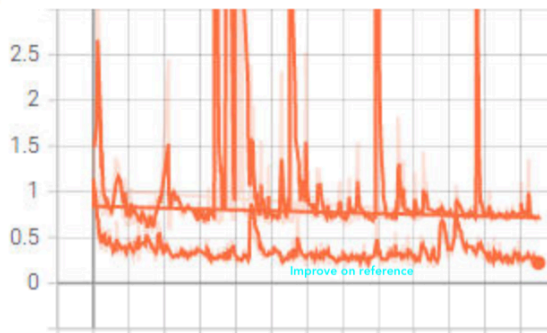
I have made multiple changes in the training configurations to improve the training. The first change is the optimizer. Instead of using the momentum optimizer I have chosen to use ADAM optimizer because Adam doesn't roll so fast after jumping over the minimum but decreasing the velocity a little bit for carefully search. I also have different learning rates at different step in the training; details can be found in the pipeline_new.config. Furthermore, I have included two more data augmentation methods: random black patches and random color distort. These two data augmentations are chosen due to the discovery that the objects in the training data usually being occluded; and for some sample data, the rain has distort the color of the objects. The results of the improvement can be found in the Loss graphs below (with highlighted

'improve on reference' to indicate the new experiment and the others were the initial experiment losses)

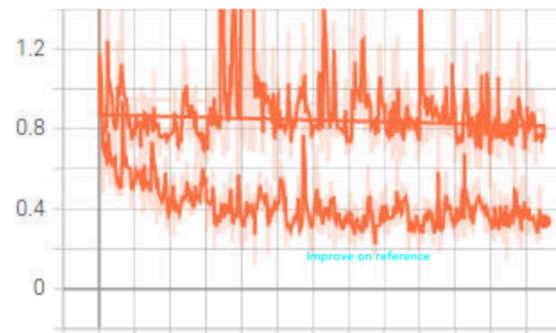
Loss/total_loss
tag: Loss/total_loss



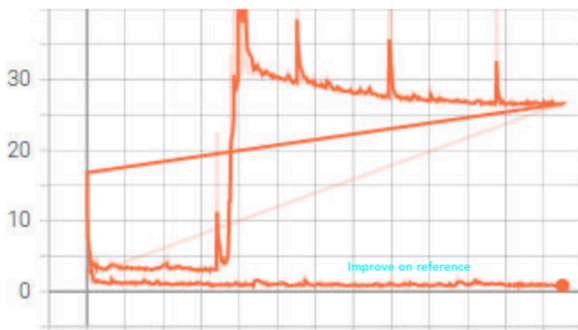
Loss/classification_loss
tag: Loss/classification_loss



Loss/localization_loss
tag: Loss/localization_loss



Loss/normalized_total_loss
tag: Loss/normalized_total_loss



Loss/regularization_loss
tag: Loss/regularization_loss

