Object Detection in an Urban Environment

Andres Murube Lindahl

2022/02/05

Index

- **Project overview**
- This section should contain a brief description of the project and what we are trying to achieve. Why is object detection such an important component of self driving car systems?
- **Dataset**
- **Dataset analysis**
- This section should contain a quantitative and qualitative description of the dataset. It should include images, charts and other visualizations.
- **Cross validation**
- This section should detail the cross validation strategy and justify your approach.
- **Training**
- **Reference experiment**
- This section should detail the results of the reference experiment. It should includes training metrics and a detailed explanation of the algorithm's performances.
- **Improve on the reference**
- This section should highlight the different strategies you adopted to improve your model. It should contain relevant figures and details of your findings.

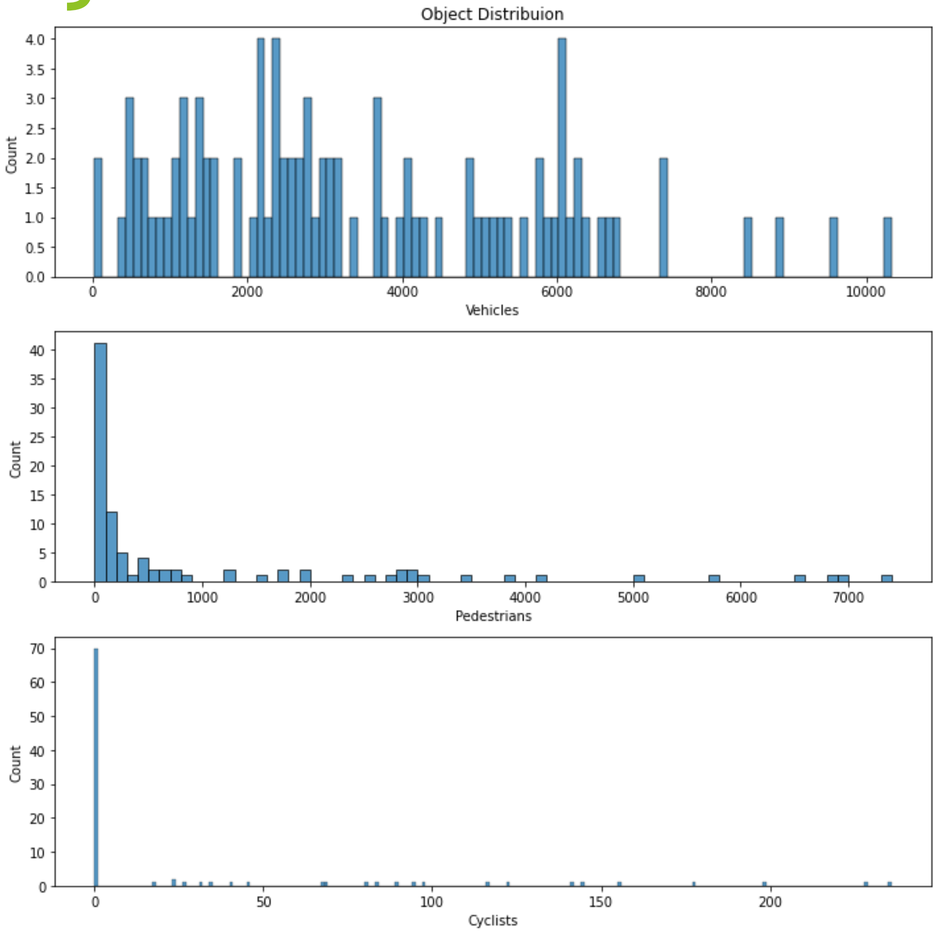
Project Overview

- The project goal is to use transfer learning techniques to accurately predict
  - vehicles, pedestrians and cyclists and
  - their position in the image

in video scenes that have been recorded.

- This task is crucial in self-driving vehicles to ensure the correct and safe trajectory planning of the ego vehicle

Exploratory Data Analysis
Object distribution over the TFRecords

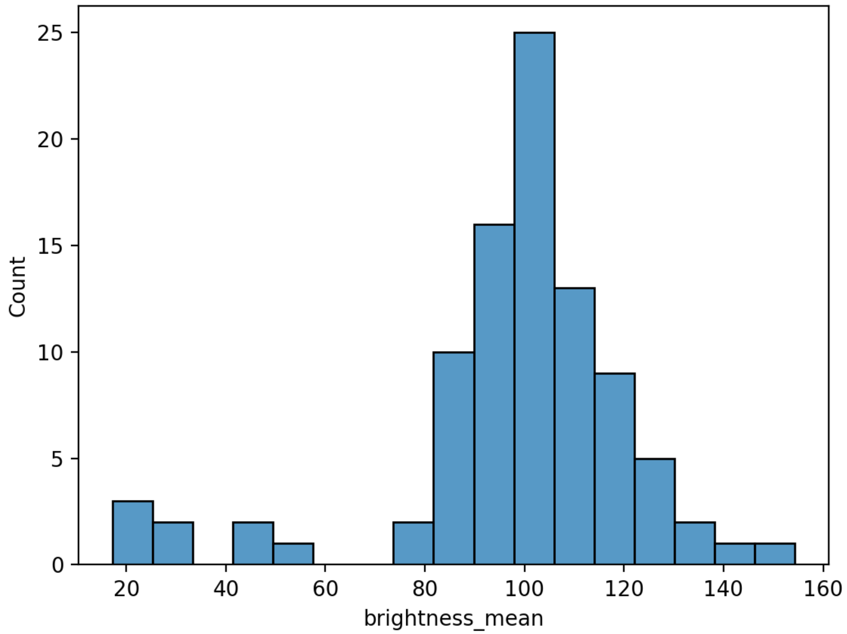


- The figure shows the distribution of the objects that can be detected for all TFRecords available
- It can clearly be observed that the #vehicles is widely distributed, whereas many frames do not have any cyclists or pedestrians
- The resulting split should try to cover these distributions as well

Cars are well represented in the dataset. Cyclists and Pedestrians are less present

Exploratory Data Analysis
Image Brightness (Day/Night scenes)

- From the calculated perceived brightness, we can clearly identify what images have been taken during the day or night.
- We want to take this into consideration when training the model. Otherwise, there is the risk of underperforming in night scenarios

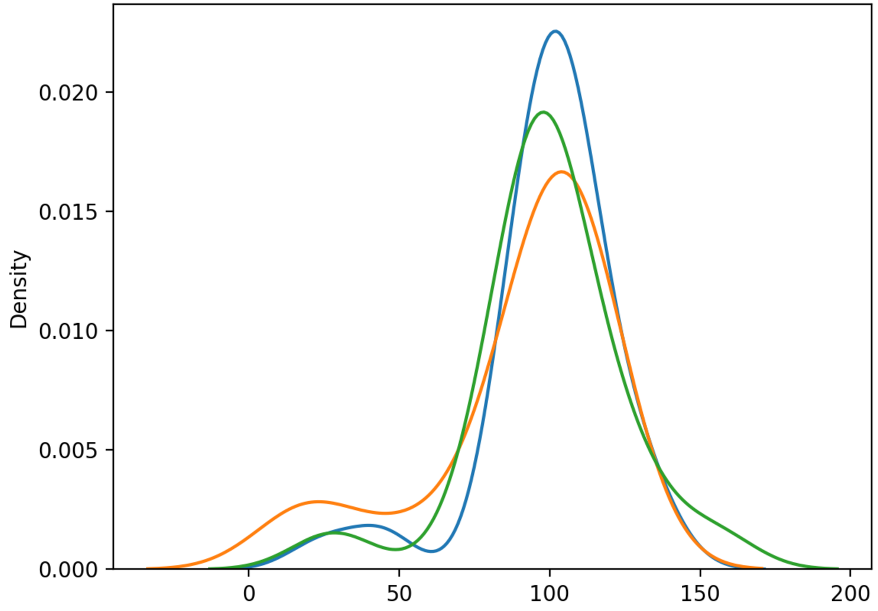


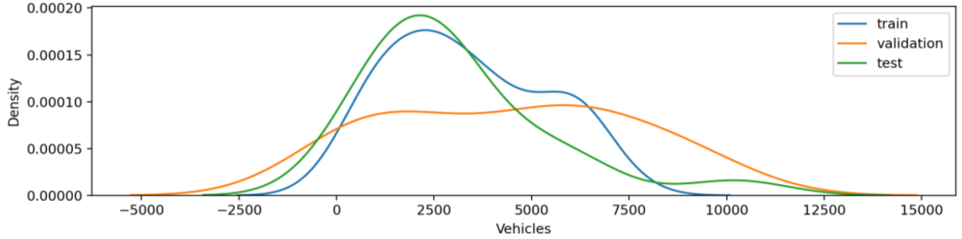
The great majority of the sequences are taken during the day. Only 9 out 100 were during the night

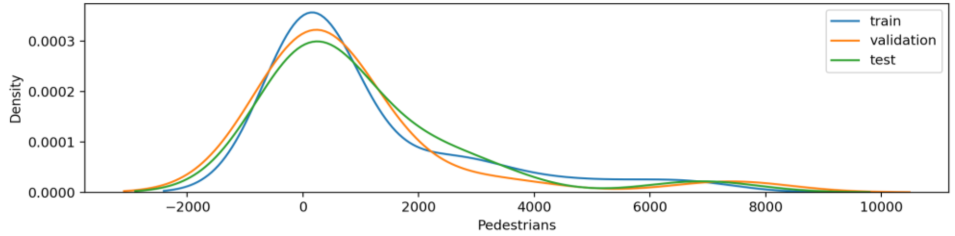
Split Creation

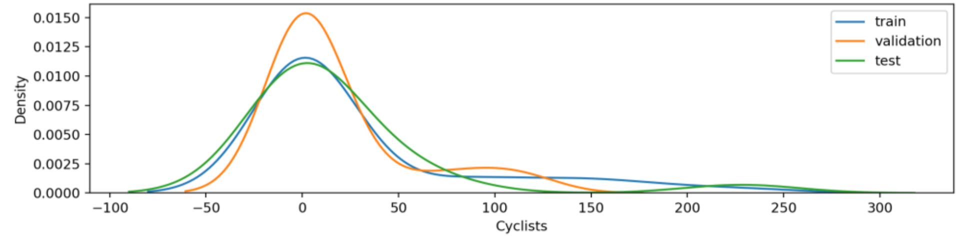
- For the creation of the different splits, we will take the classical splitting approach of 60/20/20 and evaluate whether we get similar probability densities for the classes that we evaluated before. This split can be chosen as we have a fairly large number of measurements

- From the probability distributions it can be observed that the shuffle procedure create similar datasets for training, validation and test. It will be necessary to evaluate later, whether it is sufficient or if we must make use of data augmentation to increase the features that are currently under-represented

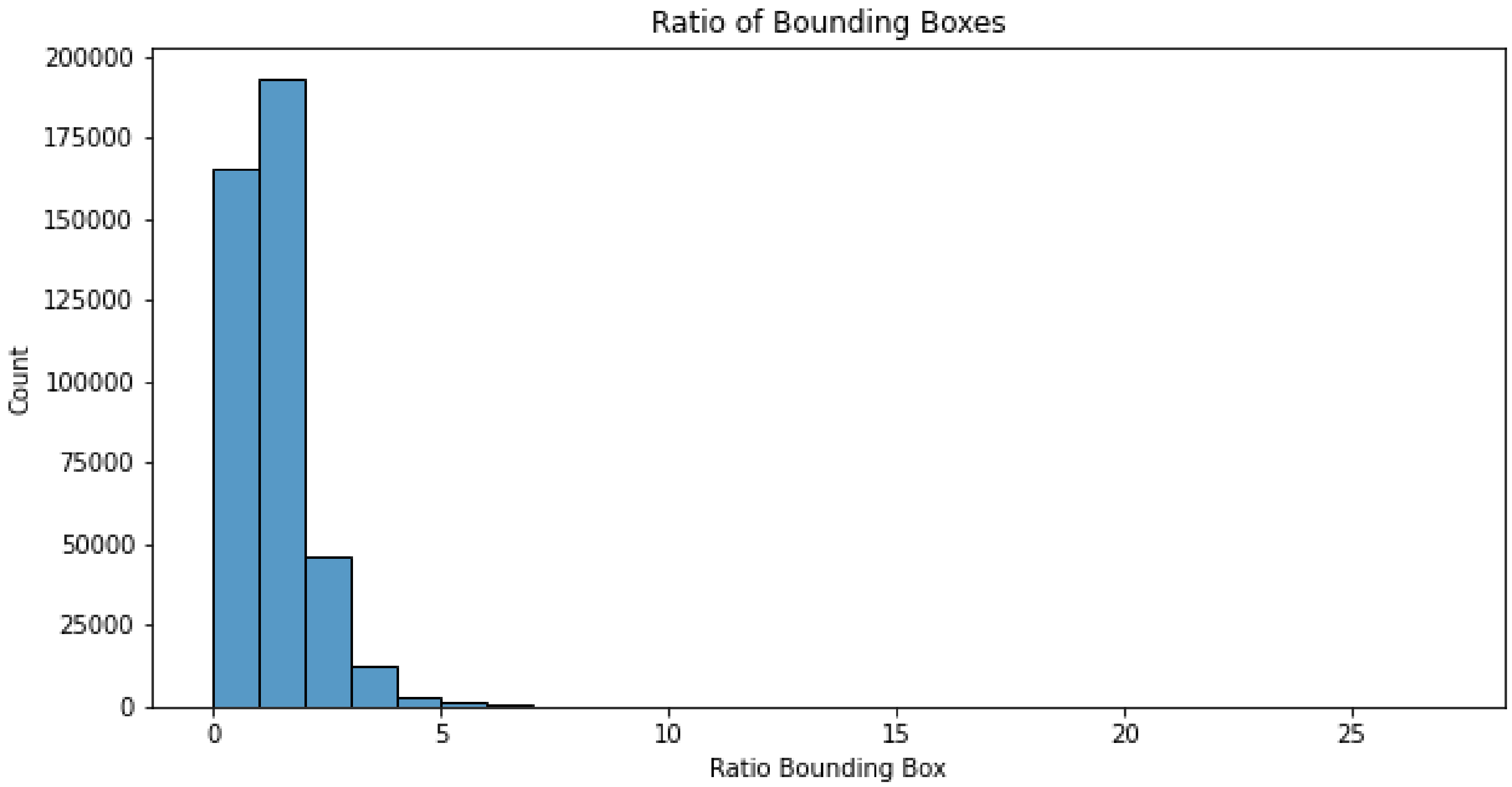


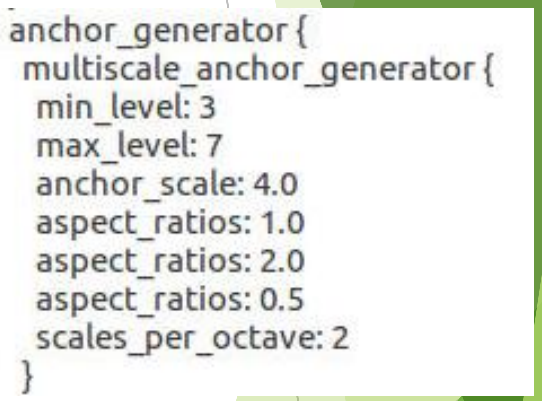






Bounding Boxes Ratio

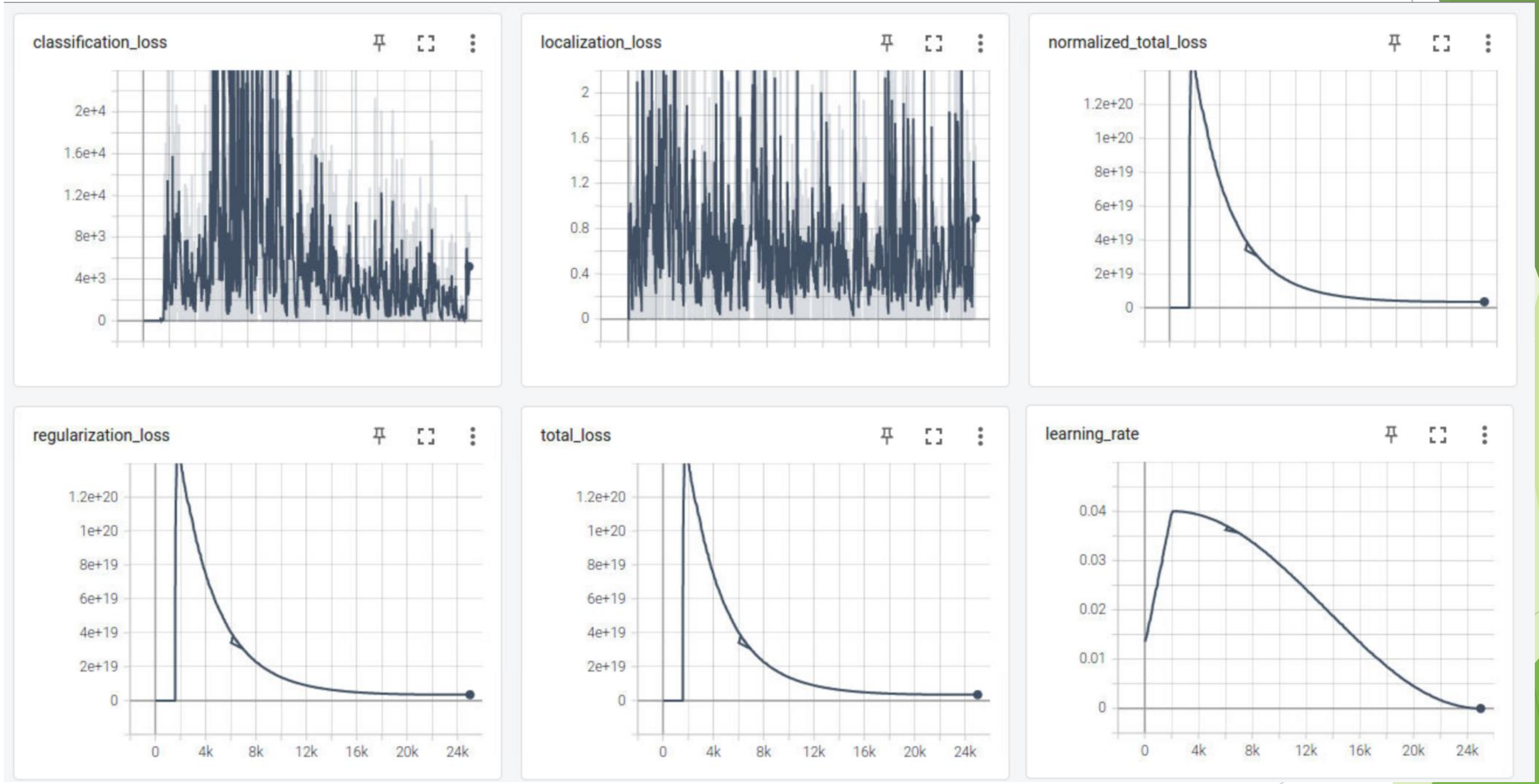




```
anchor_generator {
 multiscale_anchor_generator {
  min_level: 3
  max_level: 7
  anchor_scale: 4.0
  aspect_ratios: 1.0
  aspect_ratios: 2.0
  aspect_ratios: 0.5
  scales_per_octave: 2
 }
```

The pre-defined aspect ratios are a good starting point and don’t seem to require tweaking

Training of Reference Model: Evolution of Loss

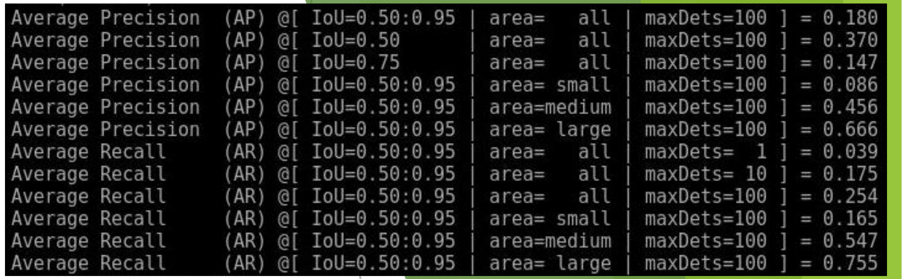


The huge losses are an indication that the learning value is too high!

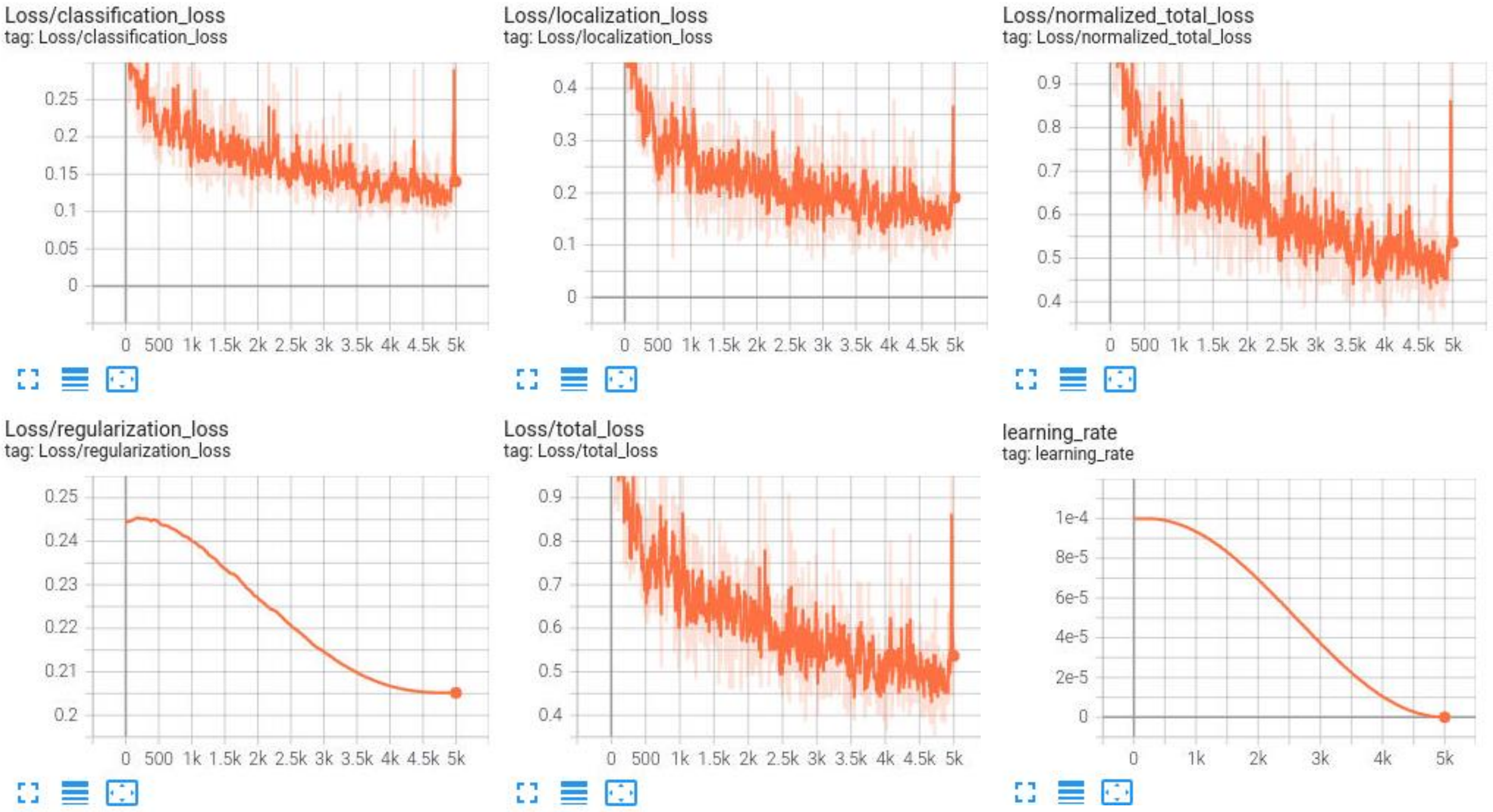
Training of Optimized Model: Changes to the configuration

- The following changes were done to improve the performance of the model:
  - Reduced the learning rate to avoid wrong learning of the model → large losses
  - Change of the optimization algorithm to ADAM
  - Include augmentations
    - random_rgb_to_gray
    - random_adjust_brightness
    - random_adjust_contrast
    - random_adjust_saturation

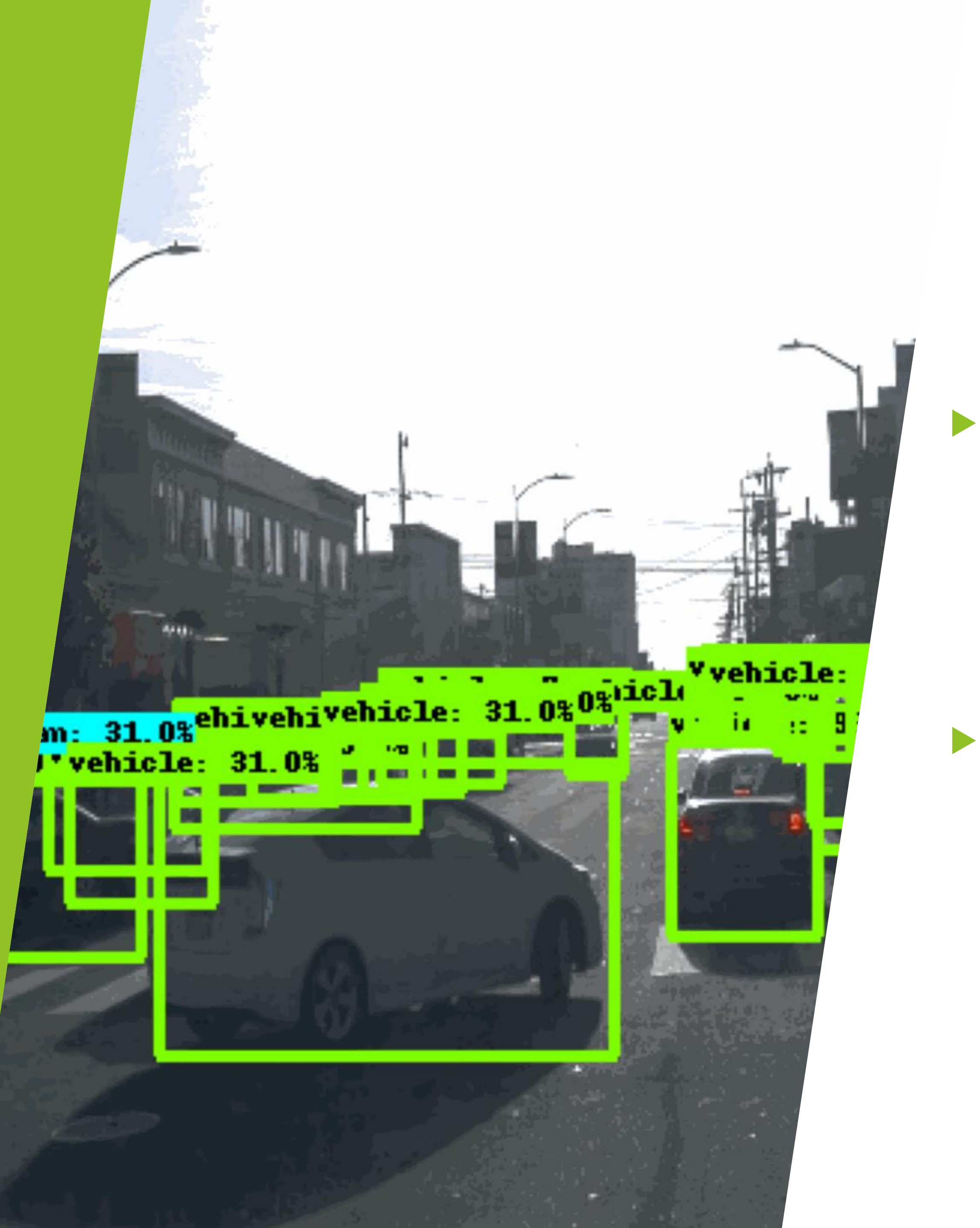
Training of Optimized Model: Evolution of Loss



```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.180
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.370
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.147
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.086
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.456
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.666
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.039
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.175
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.254
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.165
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.547
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.755
```



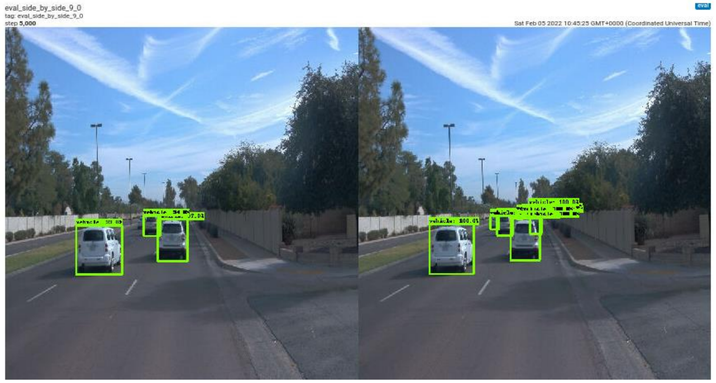
The model training convergence has significantly improved

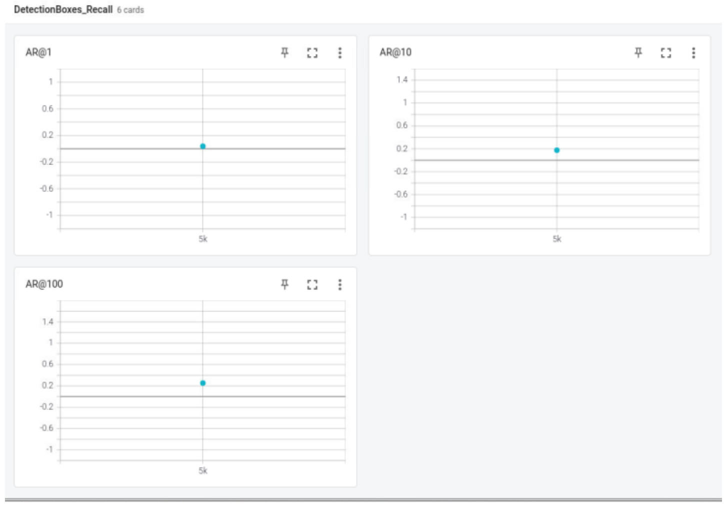


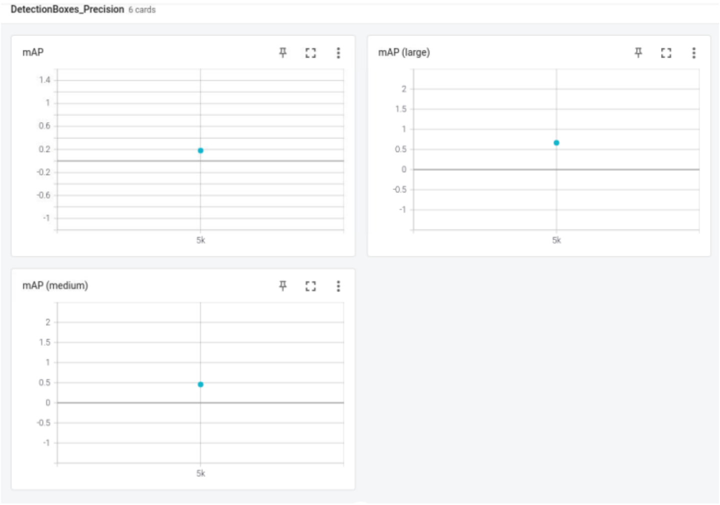
Further Possible Improvements

- Due to GPU time restrictions I could not perform the training for more than 5k epochs. Increase the values as originally configured would improve the performance of the model
- In the generated animations, it can be observed that the non-max suppression value was set too low (1e-8). Increasing this value to ~0.2 would reduce the number of windows significantly

Backup


eval_side_by_side_9_0
tag: eval_side_by_side_9_0
step 5,000
eval
Sat Feb 05 2022 10:45:25 GMT+0000 (Coordinated Universal Time)





Optimized Model Evaluation - Tensorboard

Evaluation was only possible at the end of the training as there were memory issues with the udacity workspace