# Object Detection in an Urban Environment

**Goal**

To classify and localize the cars, pedestrians and cyclists in camera input feed. This is useful for self-driving cars.

**Setup**

The code repository can be found at : https://github.com/Sunny-DV/Object-Detection-in-Urban-environment

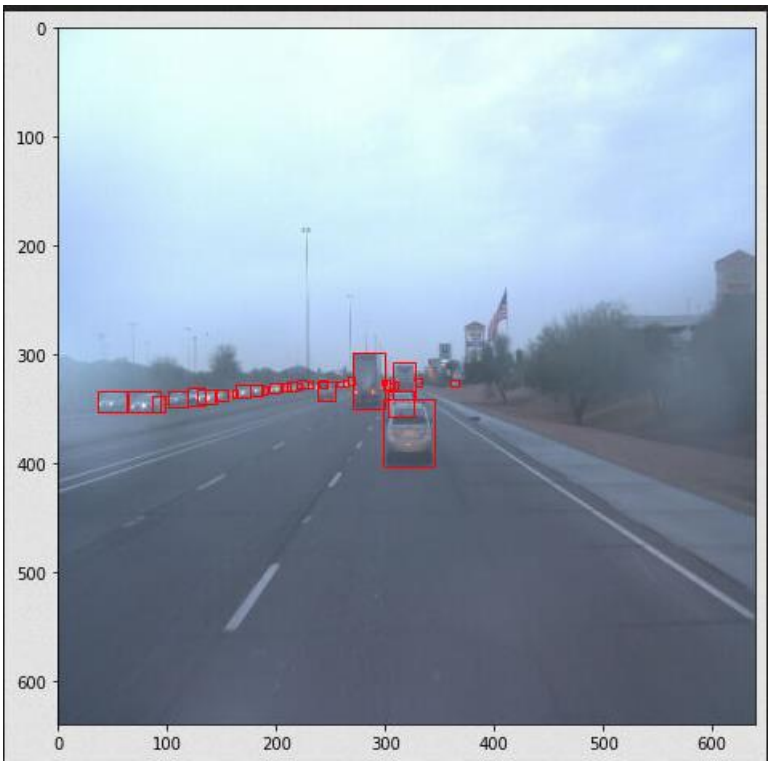
All the instruction to run the repository can be found in the README.md file present in the repository.

The config for Experiment 1 gives the best results.

**Dataset**

The dataset contains high resolution traffic images with three classes : vehicles, pedestrians and cyclists.

Below are some sample images from the dataset .



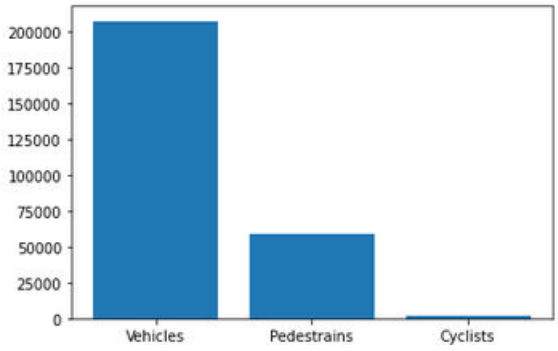

**Dataset Analysis**

The class distribution is 2070 : 586 : 148 (Vehicles : Pedestrians : Cyclists)

The class distribution is not even

Some images are taken at night and have low brightness.

Below is the class distribution graph.



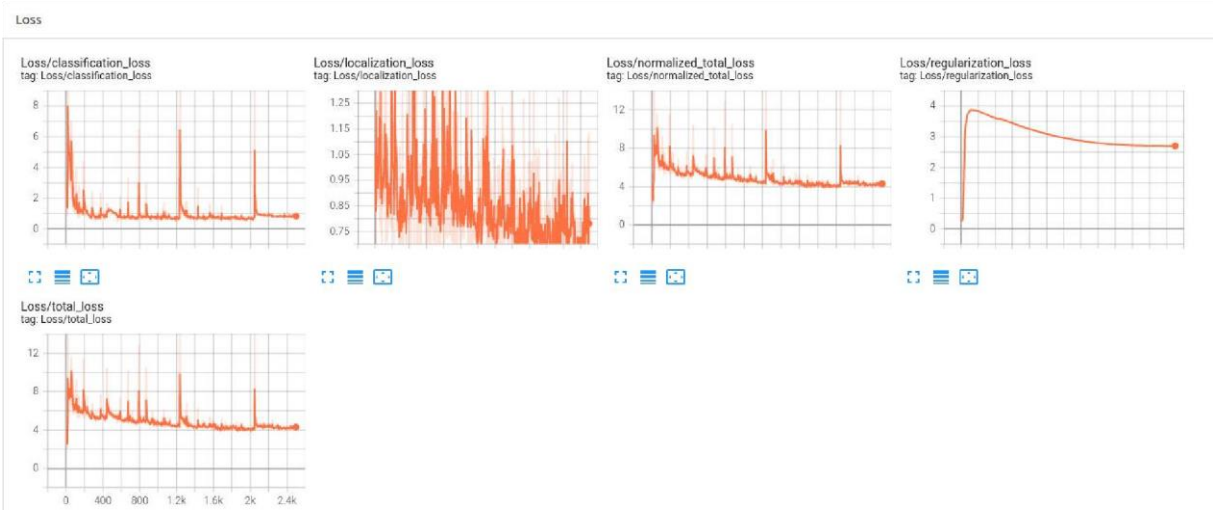
**Training**

1. *Reference experiment*: The training loss as well as the classification loss were high for this experiment. Below are the AP and recall logs :

Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.007
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.005
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.053
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000

INFO:tensorflow:Eval metrics at step 2000
I1215 21:40:06.577259 140628795279104 model_lib_v2.py:988] Eval metrics at step 2000
INFO:tensorflow: + DetectionBoxes_Precision/mAP: 0.000040
I1215 21:40:06.579910 140628795279104 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP: 0.000042
INFO:tensorflow: + DetectionBoxes_Precision/mAP@.50IOU: 0.000113
I1215 21:40:06.581568 140628795279104 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP@.50IOU: 0.000113
Loss/total_loss: 4.230478

In tensorboard logs for reference experiment the loss is on the higher side and this was expected as the dataset is hard to train and would require augmentation methods and some parameter tuning for good results.




2. *Improve on the reference*: I changed the optimizer to Adam and lowered the learning rate from 0.004 to 5e-4 . I also added some data augmentation methods like random horizontal flip, brightness adjustment (Ref. : https://github.com/tensorflow/models/blob/master/research/object_detection/protos/preprocessor.proto) etc. These were done as the dataset objects are present in mostly the centre or the periphery of the individual images. This all changes were made in Experiment 2.

   Here are the data augmentation examples :

I tried the base settings in  the reference experiment.

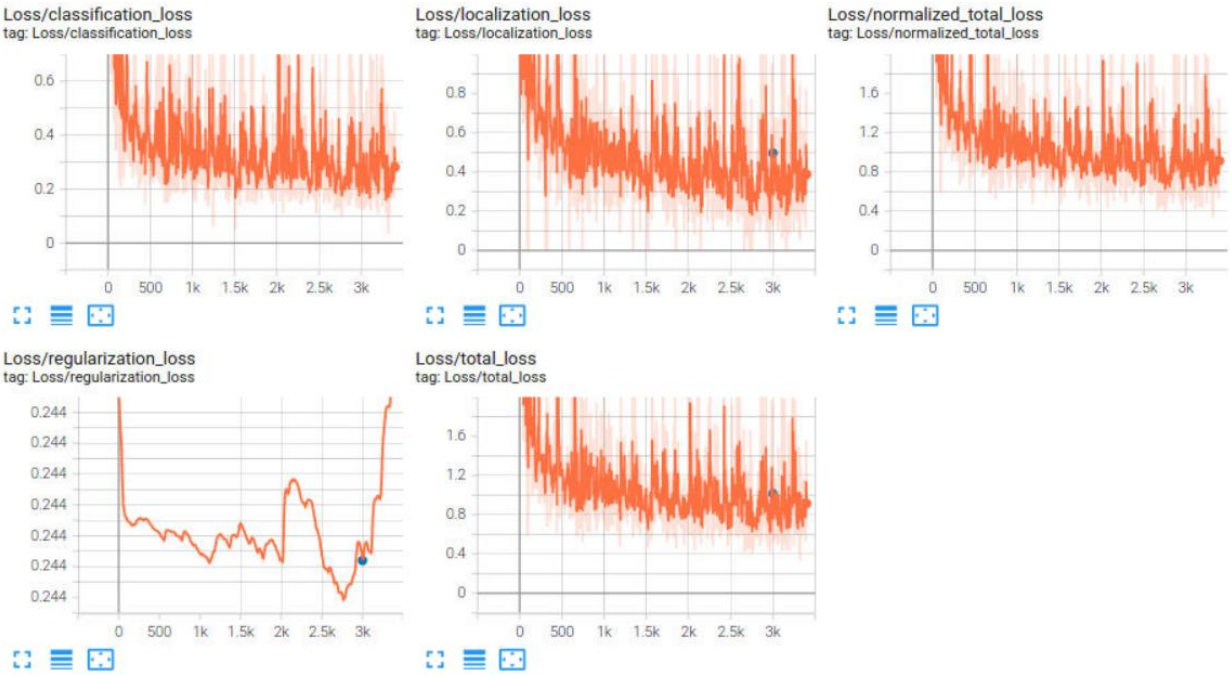
The training results were much better in Experiment 1 as compared to the reference model!

Below are the AP and Recall logs for Experiment 1:

Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.086
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.186
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.073
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.030
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.379
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.470
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.024
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.098
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.132
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.071
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.477
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.574
INFO:tensorflow:Eval metrics at step 3000
I1230 17:45:54.119803 140466852169536 model_lib_v2.py:988] Eval metrics at step 3000
INFO:tensorflow: + DetectionBoxes_Precision/mAP: 0.086412
I1230 17:45:54.129273 140466852169536 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP: 0.086412
Loss/total_loss: 1.016260

The total loss is down from 4.20 in the reference experiment to 1.016260 in Experiment 1.

Below are the tensorboard logs for Experiment 1.




It is clear that the Experiment 1 model works the best!

Below are some snippets from the animation generated by the model of Experiment 1 :


vehicle: 38.0%
vehicle: 31.0%


vehicle: 32.0%