

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 and other road safety systems.

Setup

The code repository can be found at :

<https://github.com/Vaibhav-ML/Udacity-Data-Scientist-Nanodegree>

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

The config for Experiment 2 gives the best results.

Dataset

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

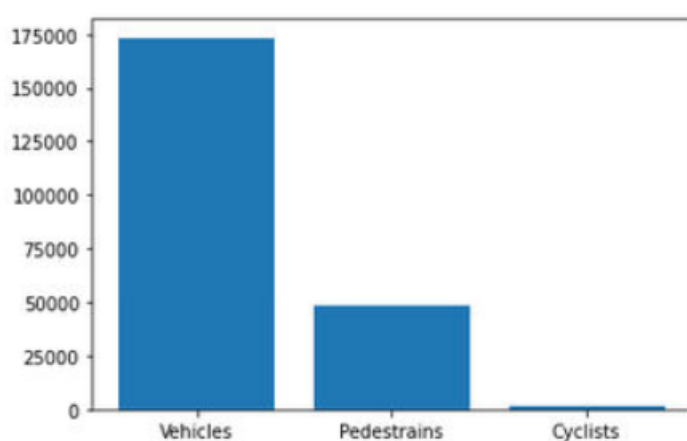
Dataset Analysis

The class distribution is roughly : 1730 : 488 : 12 (Vehicles : Pedestrians : Cyclists)

It is a heavily unbalanced class distribution.

Some images are taken at night and hence have low brightness.

Below is the class distribution graph.



Training

1. *Reference experiment*: The total training loss as well as the classification loss was really 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: 0.000042

I1215 21:40:06.579910 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP: 0.000042

INFO:tensorflow: + DetectionBoxes_Precision/mAP@.50IOU: 0.000114

I1215 21:40:06.581568 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP@.50IOU: 0.000114

INFO:tensorflow: + DetectionBoxes_Precision/mAP@.75IOU: 0.000000

I1215 21:40:06.583297 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP@.75IOU: 0.000000

INFO:tensorflow: + DetectionBoxes_Precision/mAP (small): 0.000000

I1215 21:40:06.584882 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP (small): 0.000000

INFO:tensorflow: + DetectionBoxes_Precision/mAP (medium): 0.006679

I1215 21:40:06.586670 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP (medium): 0.006679

INFO:tensorflow: + DetectionBoxes_Precision/mAP (large): -1.000000

I1215 21:40:06.588349 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Precision/mAP (large): -1.000000

INFO:tensorflow: + DetectionBoxes_Recall/AR@1: 0.000000

I1215 21:40:06.592261 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@1: 0.000000

INFO:tensorflow: + DetectionBoxes_Recall/AR@10: 0.000000

I1215 21:40:06.593793 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@10: 0.000000

INFO:tensorflow: + DetectionBoxes_Recall/AR@100: 0.004706

I1215 21:40:06.595592 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@100: 0.004706

INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (small): 0.000000

I1215 21:40:06.597191 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@100 (small): 0.000000

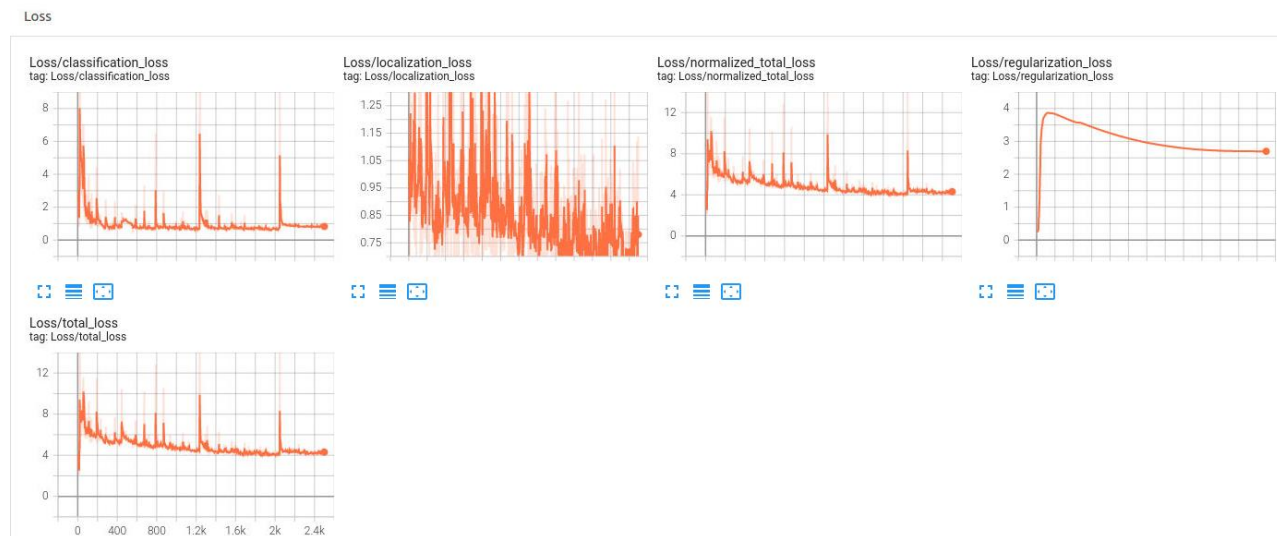
INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (medium): 0.053333

```

I1215 21:40:06.598931 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@100 (medium): 0.053333
INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (large): -1.000000
I1215 21:40:06.600528 140628795279104 model_lib_v2.py:991] +
DetectionBoxes_Recall/AR@100 (large): -1.000000
INFO:tensorflow: + Loss/localization_loss: 0.757059
I1215 21:40:06.601971 140628795279104 model_lib_v2.py:991] +
Loss/localization_loss: 0.757059
INFO:tensorflow: + Loss/classification_loss: 0.687994
I1215 21:40:06.603411 140628795279104 model_lib_v2.py:991] +
Loss/classification_loss: 0.687994
INFO:tensorflow: + Loss/regularization_loss: 2.715815
I1215 21:40:06.604907 140628795279104 model_lib_v2.py:991] +
Loss/regularization_loss: 2.715815
INFO:tensorflow: + Loss/total_loss: 4.160868
I1215 21:40:06.606426 140628795279104 model_lib_v2.py:991] +
Loss/total_loss: 4.160868

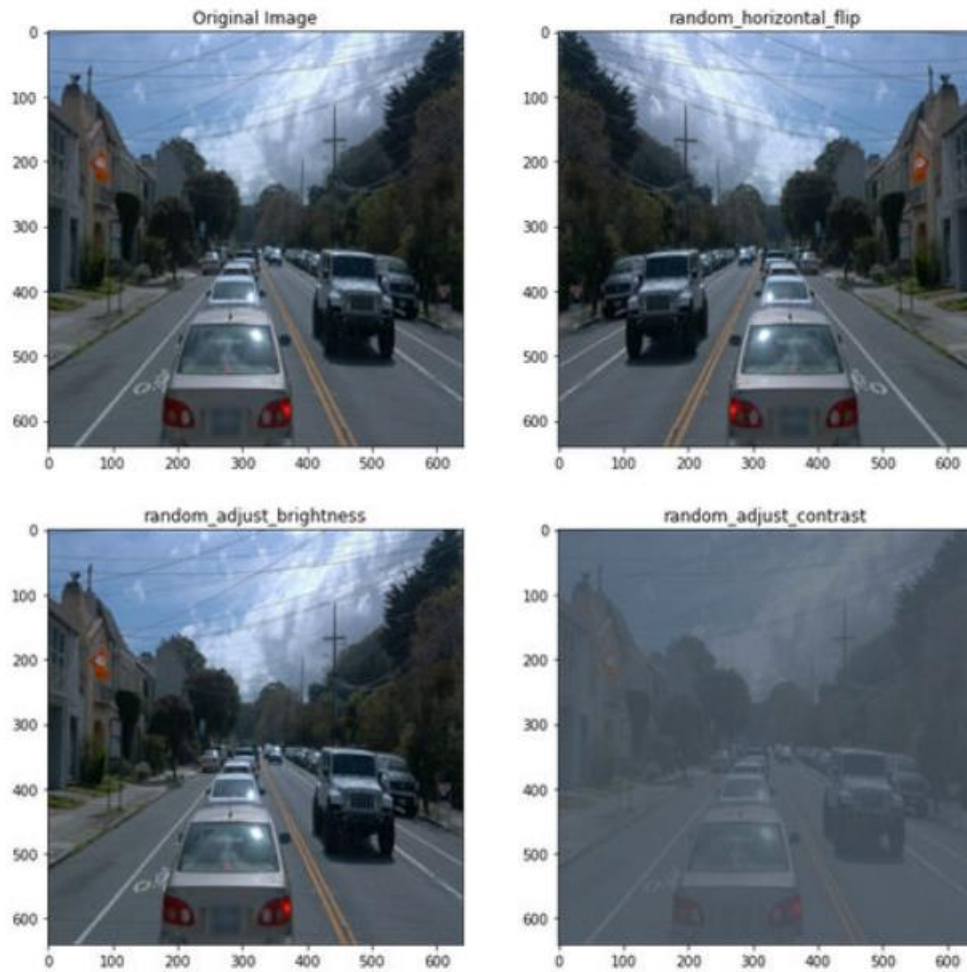
```

The tensorboard logs depict the same story . The loss is on the higher side and this was expected as the dataset is complicated and would require augmentation strategies as well as hyperparameter tuning for obtaining better 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 architecture as well as an SSD Resnet 101 V1 architecture as part of Experiment 1. Although it gave better results than the reference model but it was slower and less accurate than the model in Experiment 2.

The training results were far better in Experiment 2 as compared to the reference model!

Below are the AP and Recall logs for Experiment 2:

Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.107

Average Precision (AP) @[IoU=0.50 | area= all | maxDets=100] = 0.233

Average Precision (AP) @[IoU=0.75 | area= all | maxDets=100] = 0.089

Average Precision (AP) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.041

Average Precision (AP) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.444

Average Precision (AP) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.428

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 1] = 0.027

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 10] = 0.113

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.162

Average Recall (AR) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.097

Average Recall (AR) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.540

Average Recall (AR) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.670

INFO:tensorflow:Eval metrics at step 2500

I1215 19:02:29.361037 140411958515456 model_lib_v2.py:988] Eval metrics at step 2500

INFO:tensorflow: + DetectionBoxes_Precision/mAP: 0.106636

I1215 19:02:29.369823 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP: 0.106636

INFO:tensorflow: + DetectionBoxes_Precision/mAP@.50IOU: 0.233006

I1215 19:02:29.371702 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP@.50IOU: 0.233006

INFO:tensorflow: + DetectionBoxes_Precision/mAP@.75IOU: 0.088867

I1215 19:02:29.373291 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP@.75IOU: 0.088867

INFO:tensorflow: + DetectionBoxes_Precision/mAP (small): 0.041406

I1215 19:02:29.374817 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP (small): 0.041406

INFO:tensorflow: + DetectionBoxes_Precision/mAP (medium): 0.443651

I1215 19:02:29.376449 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP (medium): 0.443651

INFO:tensorflow: + DetectionBoxes_Precision/mAP (large): 0.428136

I1215 19:02:29.378222 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Precision/mAP (large): 0.428136

INFO:tensorflow: + DetectionBoxes_Recall/AR@1: 0.027315

I1215 19:02:29.379950 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@1: 0.027315

INFO:tensorflow: + DetectionBoxes_Recall/AR@10: 0.113490

I1215 19:02:29.382022 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@10: 0.113490

INFO:tensorflow: + DetectionBoxes_Recall/AR@100: 0.161706

I1215 19:02:29.383764 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@100: 0.161706

INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (small): 0.097009

I1215 19:02:29.385457 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@100 (small): 0.097009

INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (medium): 0.540135

I1215 19:02:29.387208 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@100 (medium): 0.540135

INFO:tensorflow: + DetectionBoxes_Recall/AR@100 (large): 0.670400

I1215 19:02:29.389199 140411958515456 model_lib_v2.py:991] + DetectionBoxes_Recall/AR@100 (large): 0.670400

INFO:tensorflow: + Loss/localization_loss: 0.449796

I1215 19:02:29.390682 140411958515456 model_lib_v2.py:991] + Loss/localization_loss: 0.449796

INFO:tensorflow: + Loss/classification_loss: 0.244869

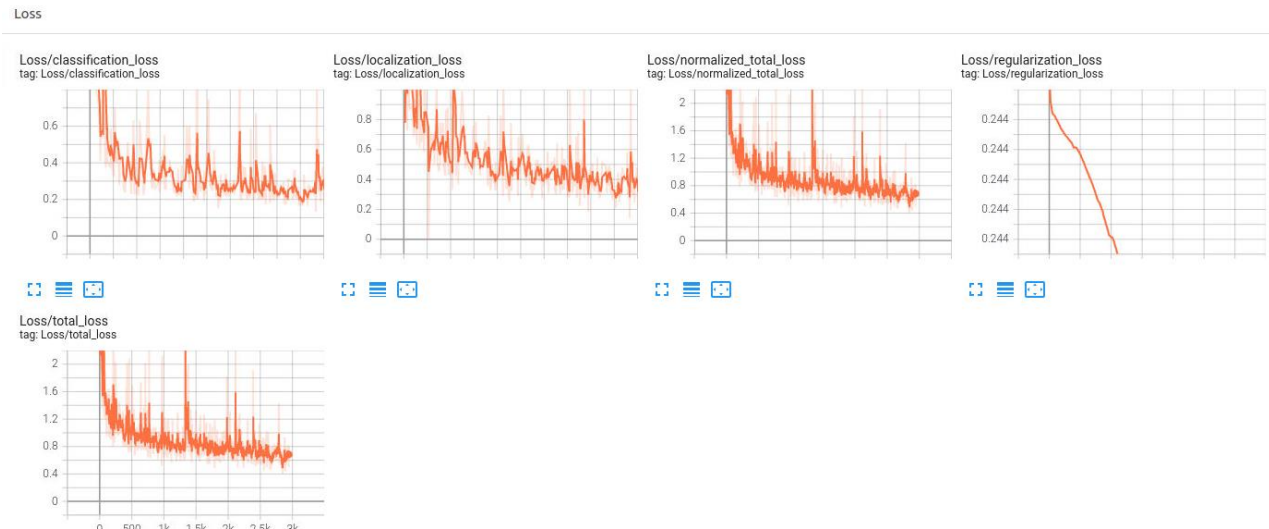
I1215 19:02:29.392213 140411958515456 model_lib_v2.py:991] + Loss/classification_loss: 0.244869

INFO:tensorflow: + Loss/regularization_loss: 0.242850

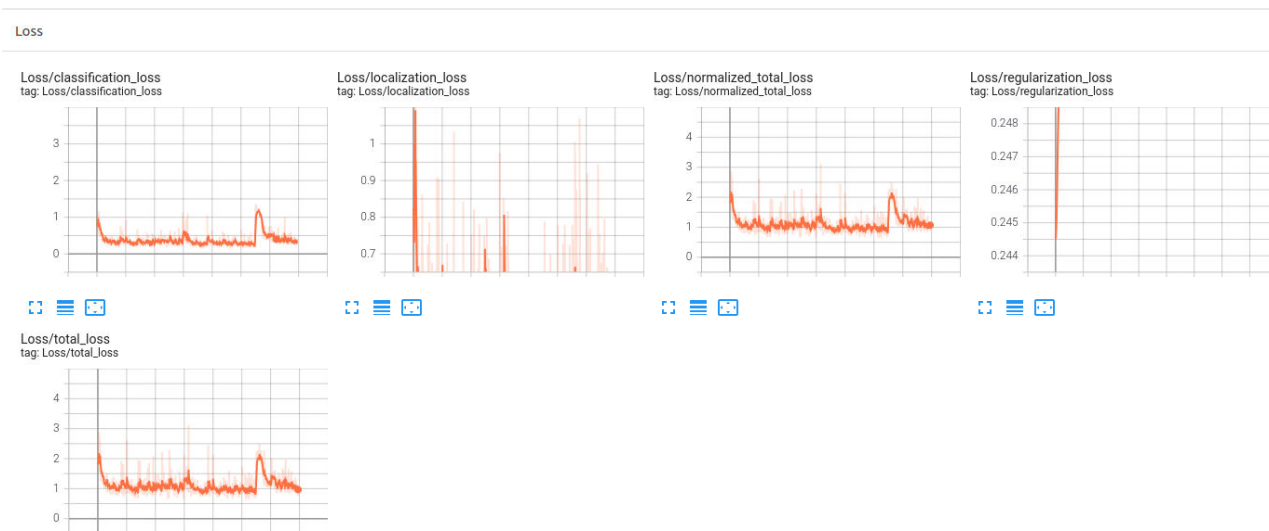

```
I1215 19:02:29.393750 140411958515456 model_lib_v2.py:991] +  
Loss/regularization_loss: 0.242850  
INFO:tensorflow: + Loss/total_loss: 0.937514  
I1215 19:02:29.395201 140411958515456 model_lib_v2.py:991] +  
Loss/total_loss: 0.937514
```

The total loss is down from 4.16 in the reference experiment to 0.9375 in Experiment 2.

Below are the tensorboard logs for Experiment 2.



Below are the tensorboard logs for Experiment 1.



It is evident that the Experiment 2 model works the best!

Below you can find some snippets from the animation generated by the model of Experiment 2 :

