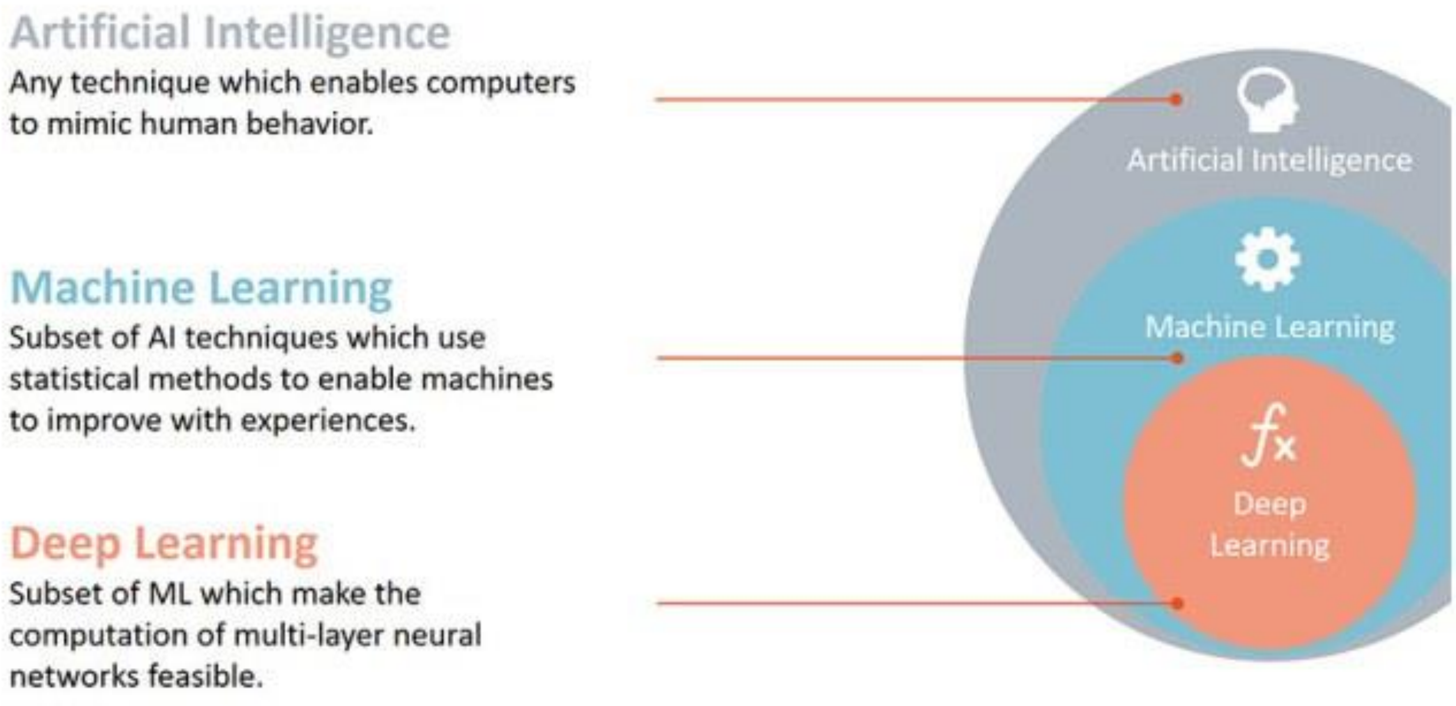
DLSC Hackathon

AAMER ABDUL RAHMAN

Contents

What is Deep Learning?



Examples

Artificial Intelligence:

- Rule-based/fuzzy logic Systems
- NLP

Machine Learning:

- Linear Regression
- Logistic Regression
- Random Forest Algorithm

Deep Learning:

- RNN (LSTM, GRU)
- CNN
- GAN
- CRNN / YOLO / SSD

Why Deep Learning?

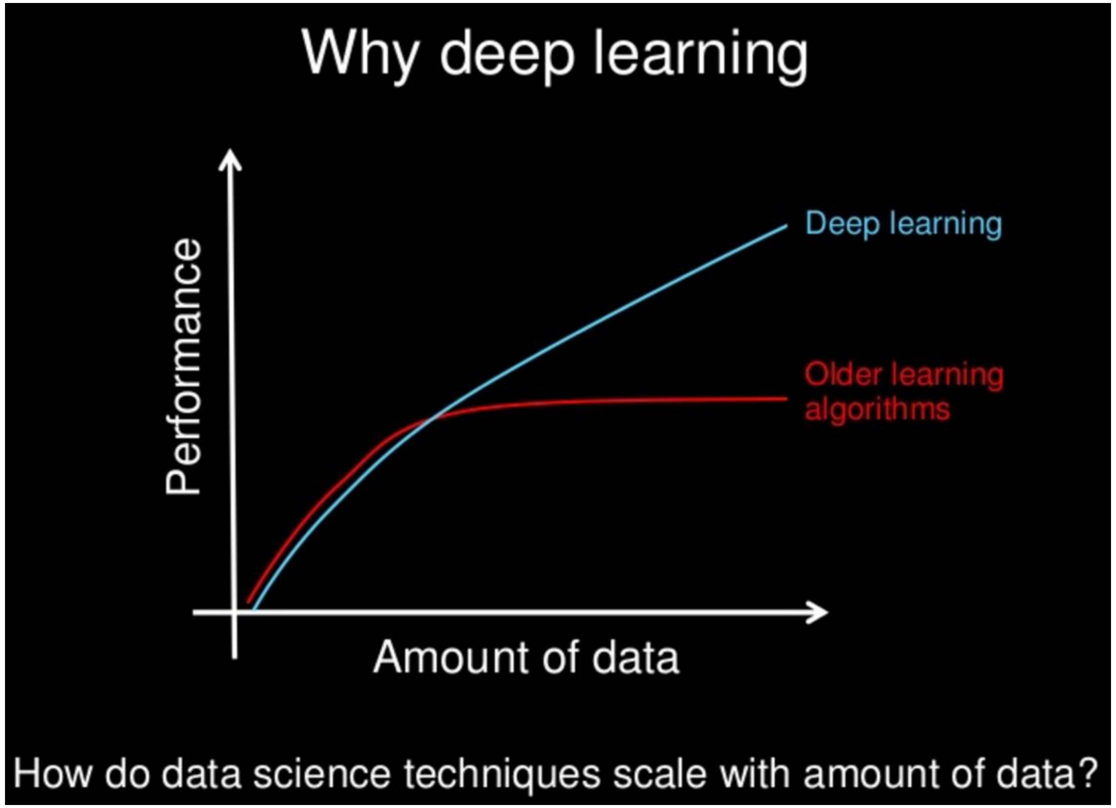
Why Deep Learning?

Convolutional neural networks, also called ConvNets, were first introduced in the 1980s by Yann LeCun.

Deep Learning went through a hibernation period till the recent advancements in:

- 1) Computational performance
- 2) Data availability

Why Deep Learning?



https://machinelearningmastery.com/what-is-deep-learning/

GPU vs CPU

A GPU is designed to quickly render or process high-resolution images and video concurrently.

Because GPUs can perform parallel operations on multiple sets of data, they are also commonly used for non-graphical tasks such as **deep learning** and **scientific computation(eg. Matrix operation and high dimensional vector operation)**.

Designed with hundreds of processor cores running simultaneously, GPUs enable massive parallelism where each core is focused on making efficient calculations.

In simple terms, if an image is provided to CPU and GPU then CPU process each pixel one by one(sequential) whereas GPU process hundreds of pixel together (concurrently/parallelism) with it’s hundreds of cores.

https://mc.ai/cpu-vs-gpu-for-deep-learning/

Applications of Vehicle Detection via Satellite Imagery

Traffic and congestion detection

Future Application:

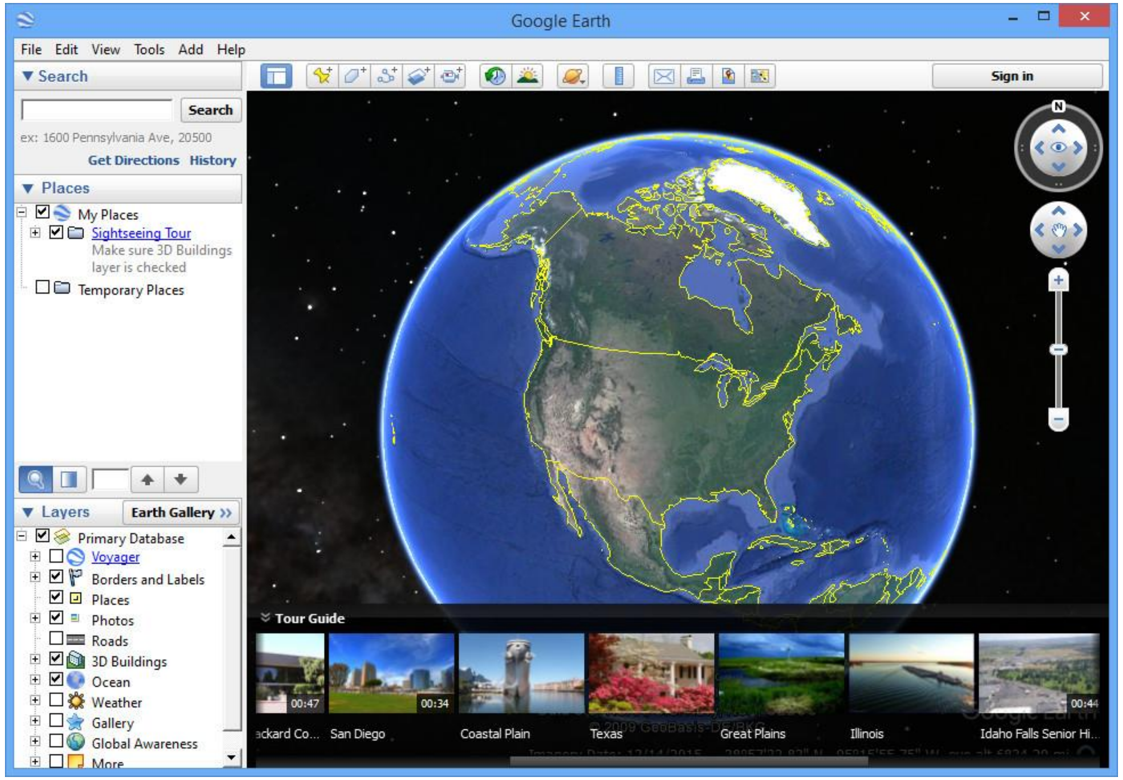
- Provide centralised data for a road system consisting of autonomous vehicle

GPU vs CPU

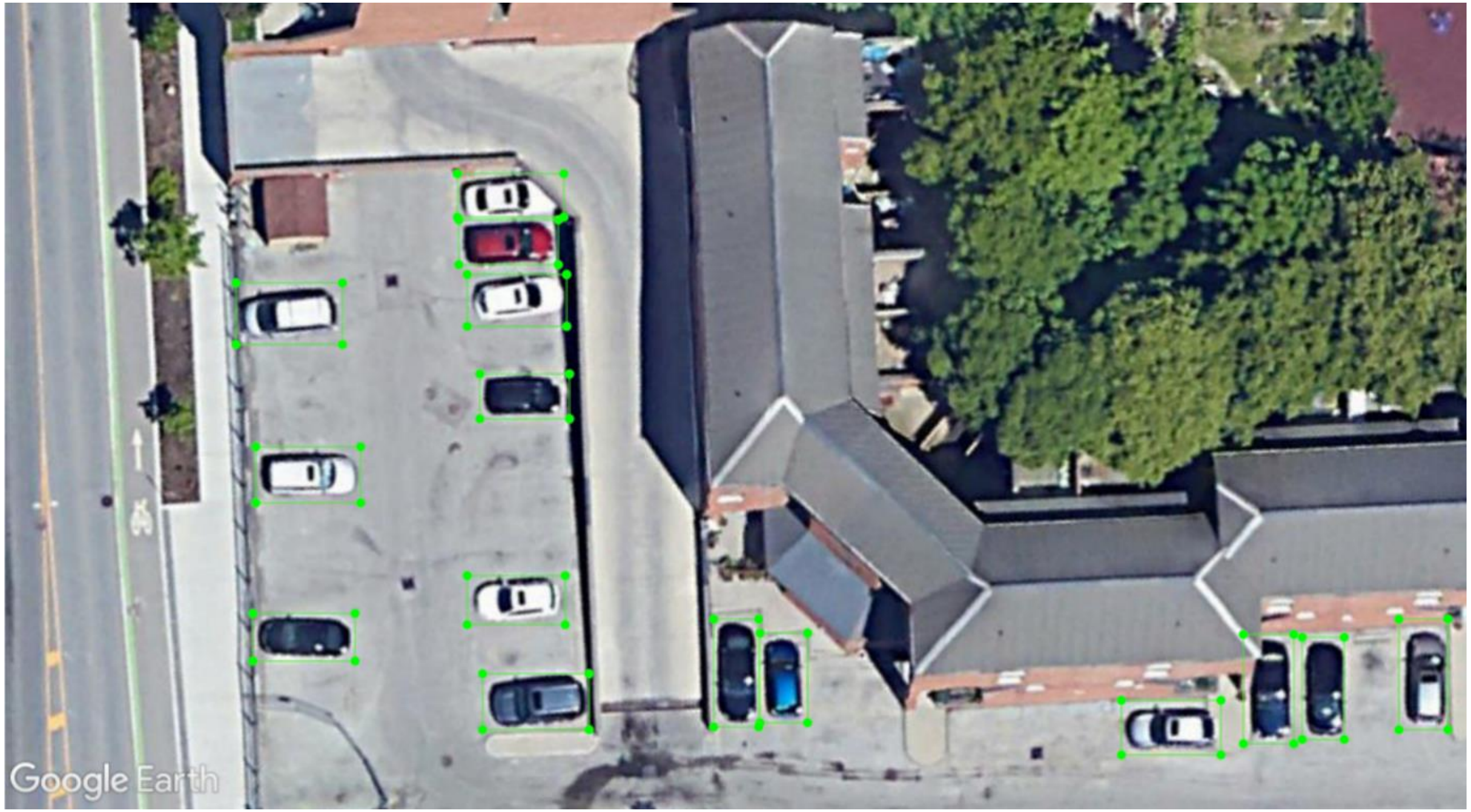
The GPU provided by colab: Nvidia Tesla K80 GPU

The CPU provided by colab: Intel(R) Xeon(R) CPU @ 2.30GHz

Data Collection



Data Preparation



Different Deep Learning Object Detection Techniques

YOLO

RCNN, Fast-RCNN, Faster-RCNN

SSD

Pretrained YoloV3 (Transfer Learning)

Transfer Learning: Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks.

Train / Validation / Test Split – Experiment 1

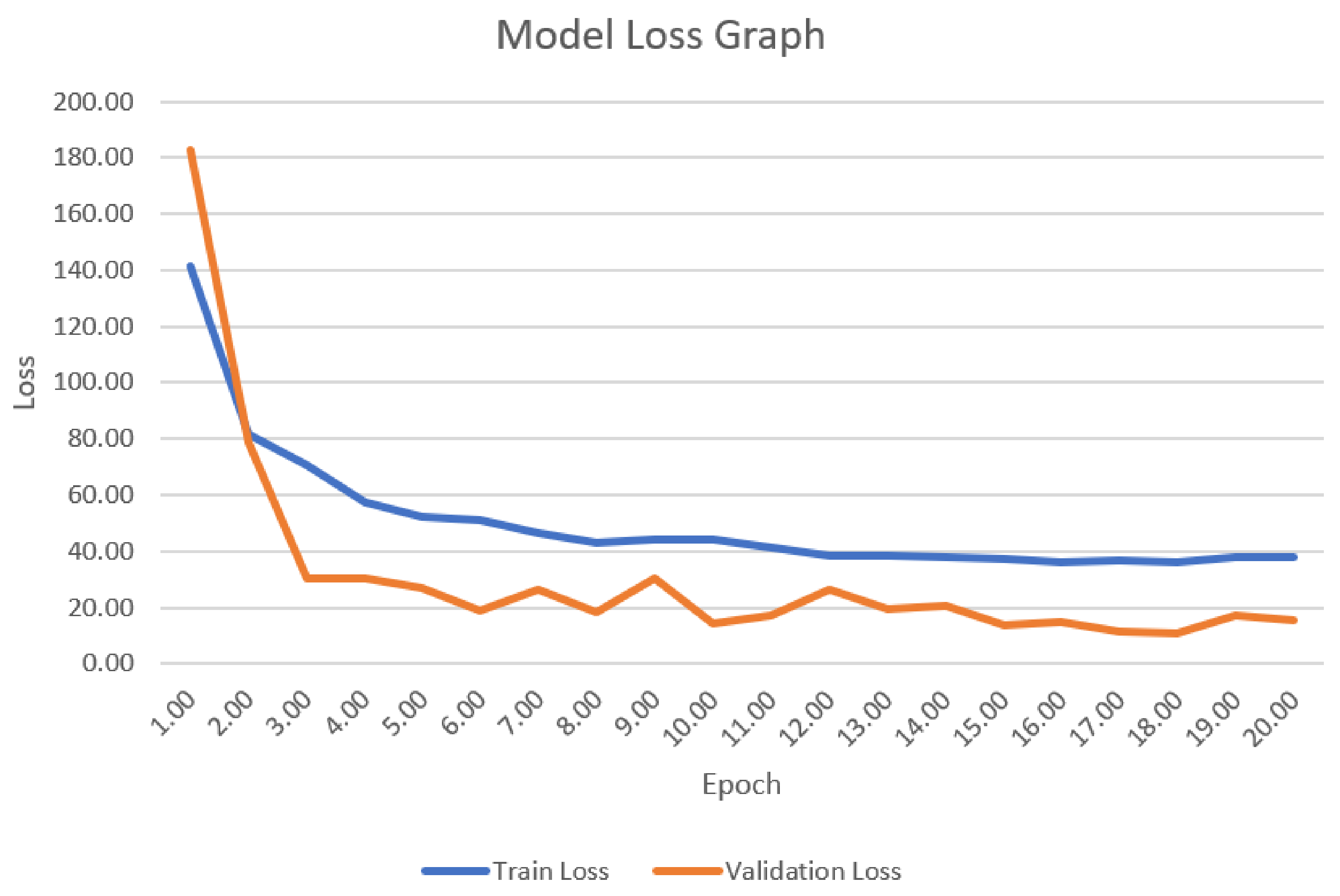
Initial Experiment:

- Train : 12 images
- Eval : 3 images
- Test: 5 images

Total :

23 Images

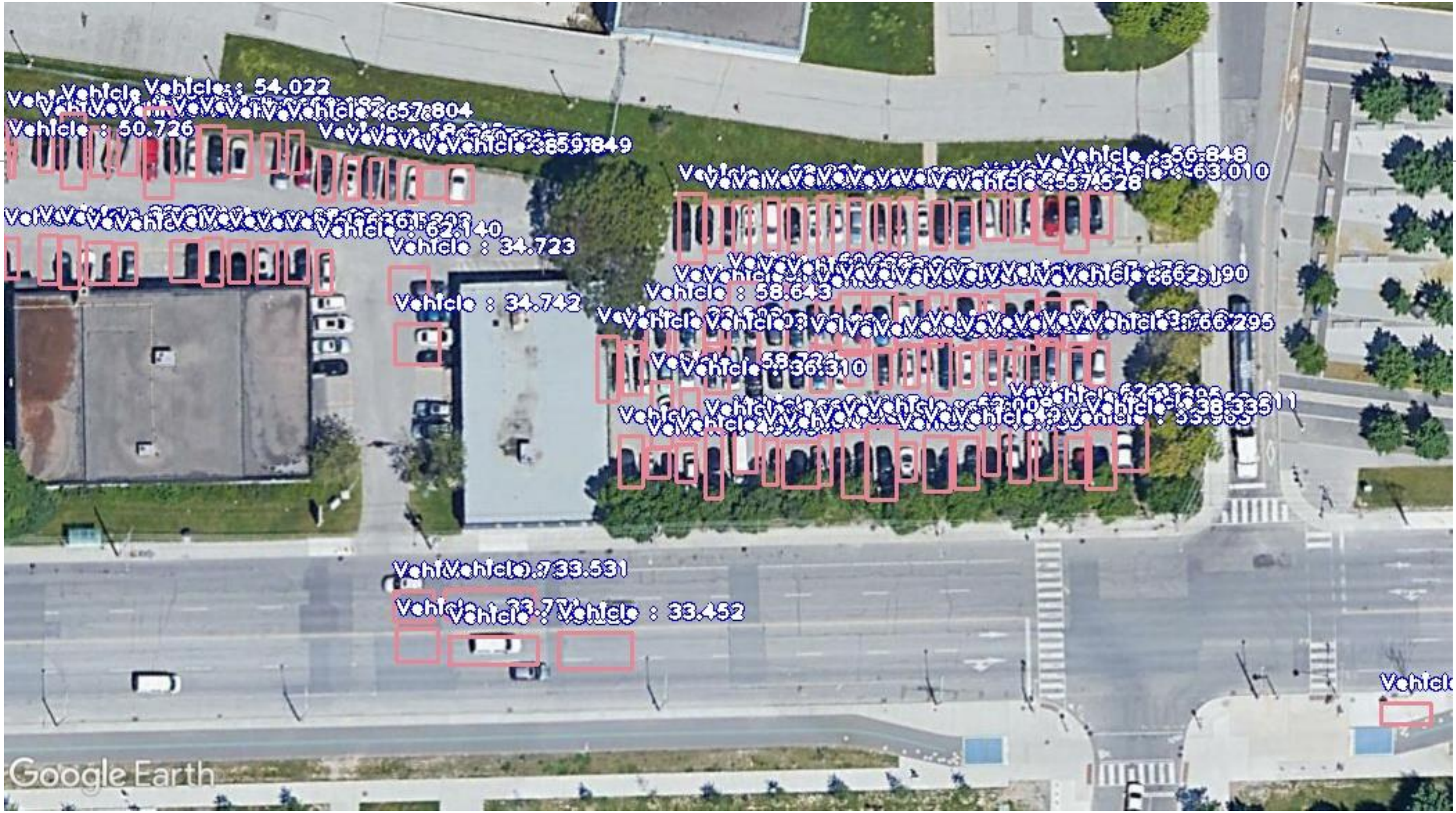
500~ objects



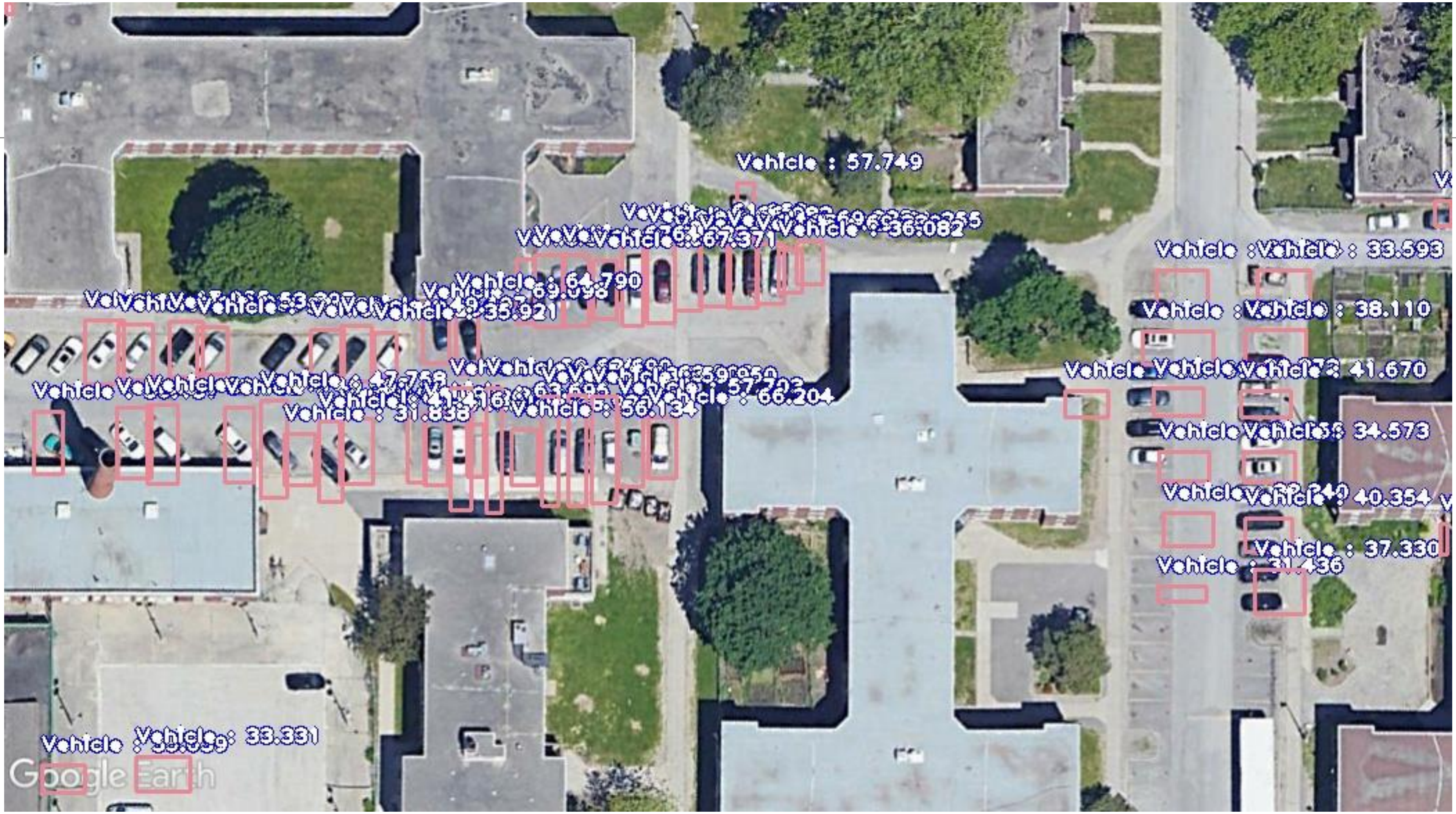
Model Loss Graph

BEST MEAN AVERAGE PRECISION: 0.12

Vehicle : 62.690
Vehicle : 68.894
Vehicle : 66.847
Vehicle : 62.556
Vehicle : 37.505
Vehicle : 36.119
Vehicle : 39.614
Vehicle : 43.414
Vehicle : 31.857
Vehicle : 33.771
Google Earth


Vehicle : 54.022
Vehicle : 50.726
Vehicle : 62.140
Vehicle : 34.723
Vehicle : 34.742
Vehicle : 58.643
Vehicle : 56.848
Vehicle : 33.452
Google Earth


Vehicle : 35.294
Vehicle : 40.109
Vehicle : 56.081
Vehicle : 63.309
Vehicle : 59.865
Vehicle : 63.054
Vehicle : 51.918
Vehicle : 33.005
Vehicle : 50.938
Vehicle : 57.903
Vehicle : 37.098
Vehicle : 72.763
Vehicle : 37.850
Google Earth


Vehicle : 57.749
Vehicle : 36.082
Vehicle : 33.593
Vehicle : 38.110
Vehicle : 41.670
Vehicle : 34.573
Vehicle : 40.354
Vehicle : 37.330
Vehicle : 64.790
Vehicle : 35.921
Vehicle : 66.204
Vehicle : 56.134
Vehicle : 31.838
Vehicle : 33.331
Google Earth

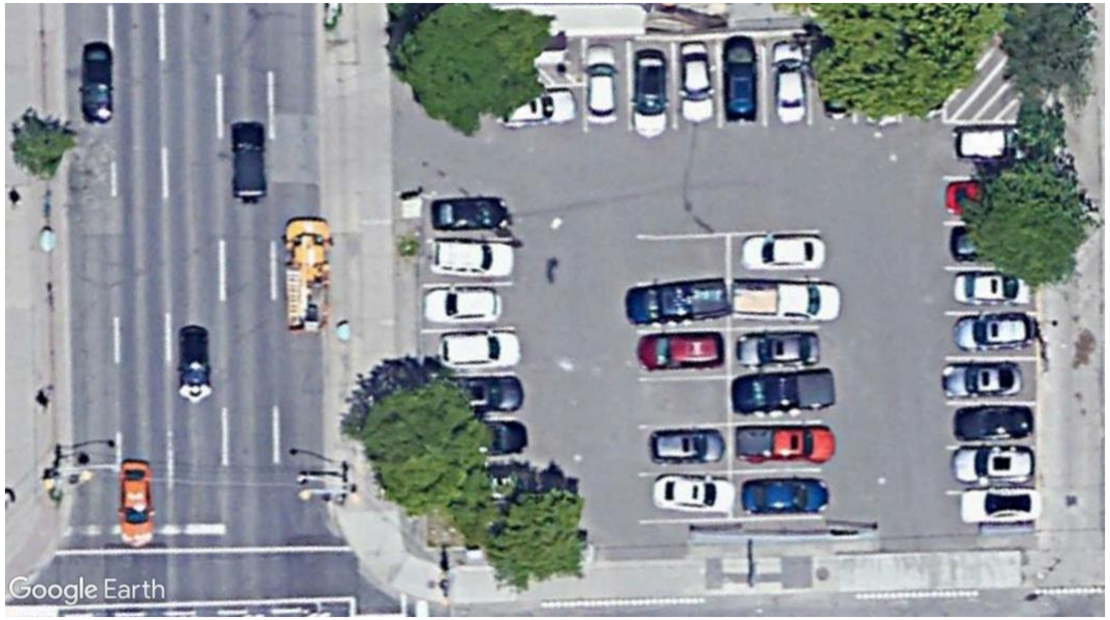

Vehicle : 48.385
Vehicle : 64.667
Vehicle : 47.366
Vehicle : 47.737
Vehicle : 33.065
Vehicle : 66.025
Vehicle : 54.659
Vehicle : 37.230
Vehicle : 62.641
Google Earth

Experiment 2: Changing the data distribution between the Train and Val set

Best Mean Average Precision:

- Fold 1: 0.12
- Fold 2: 0.039
- Fold 3: 0.148

Experiment 3: Recollect Data



Old Image

New Image

Zoomed In Dataset Distribution

Total: 40 images , 350 – 400 objects

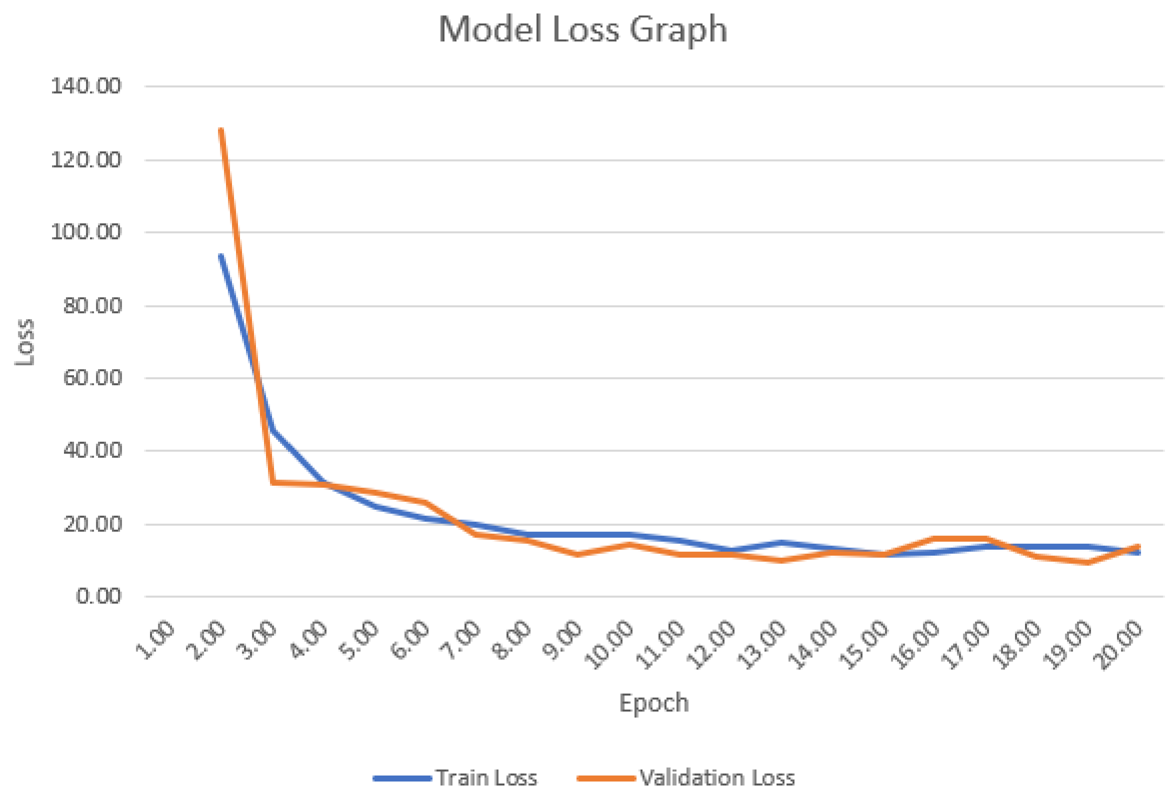
Train: 30 images

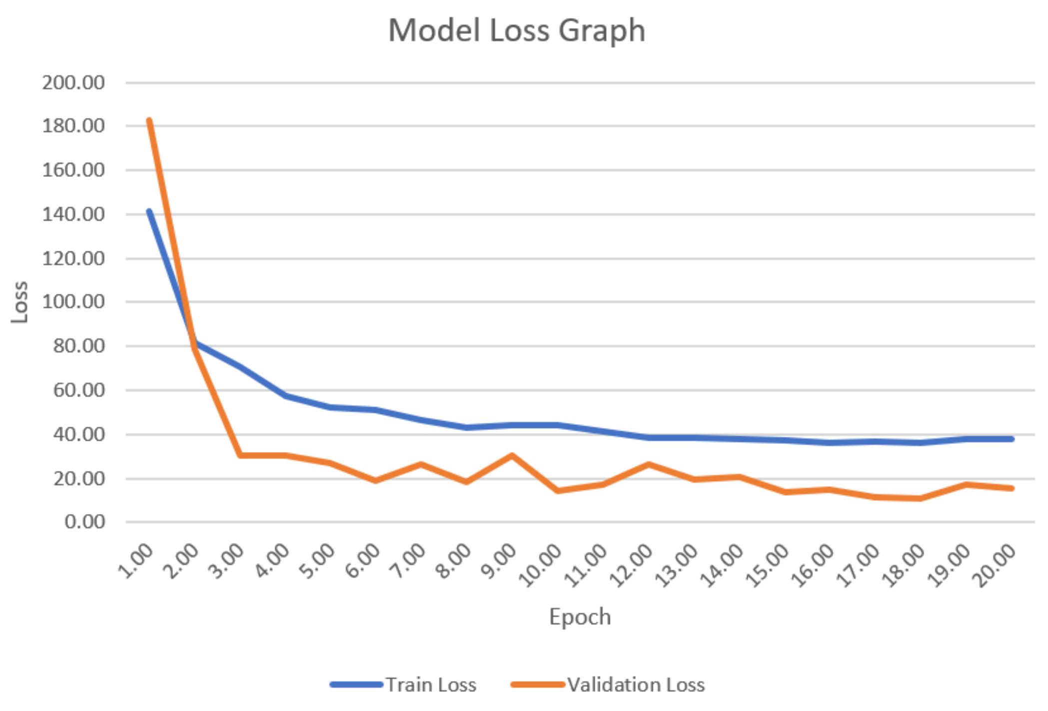
Val: 5 images

Test: 5 images

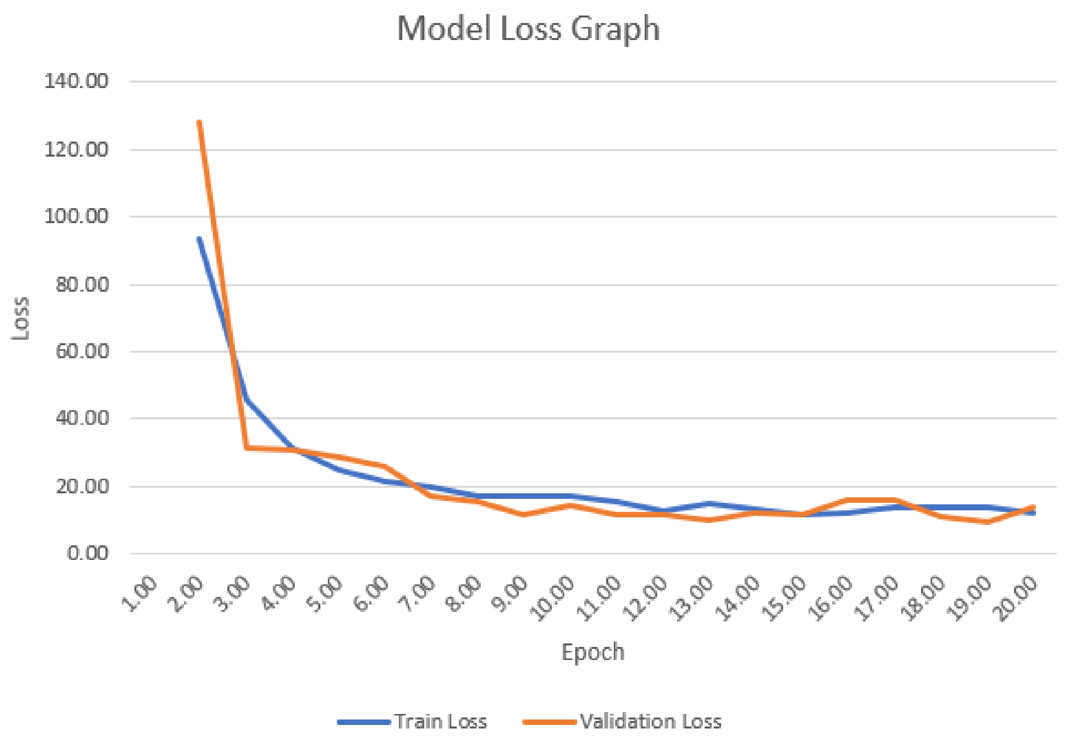
Experiment 2 results

Best Mean Average Precision: 0.1125





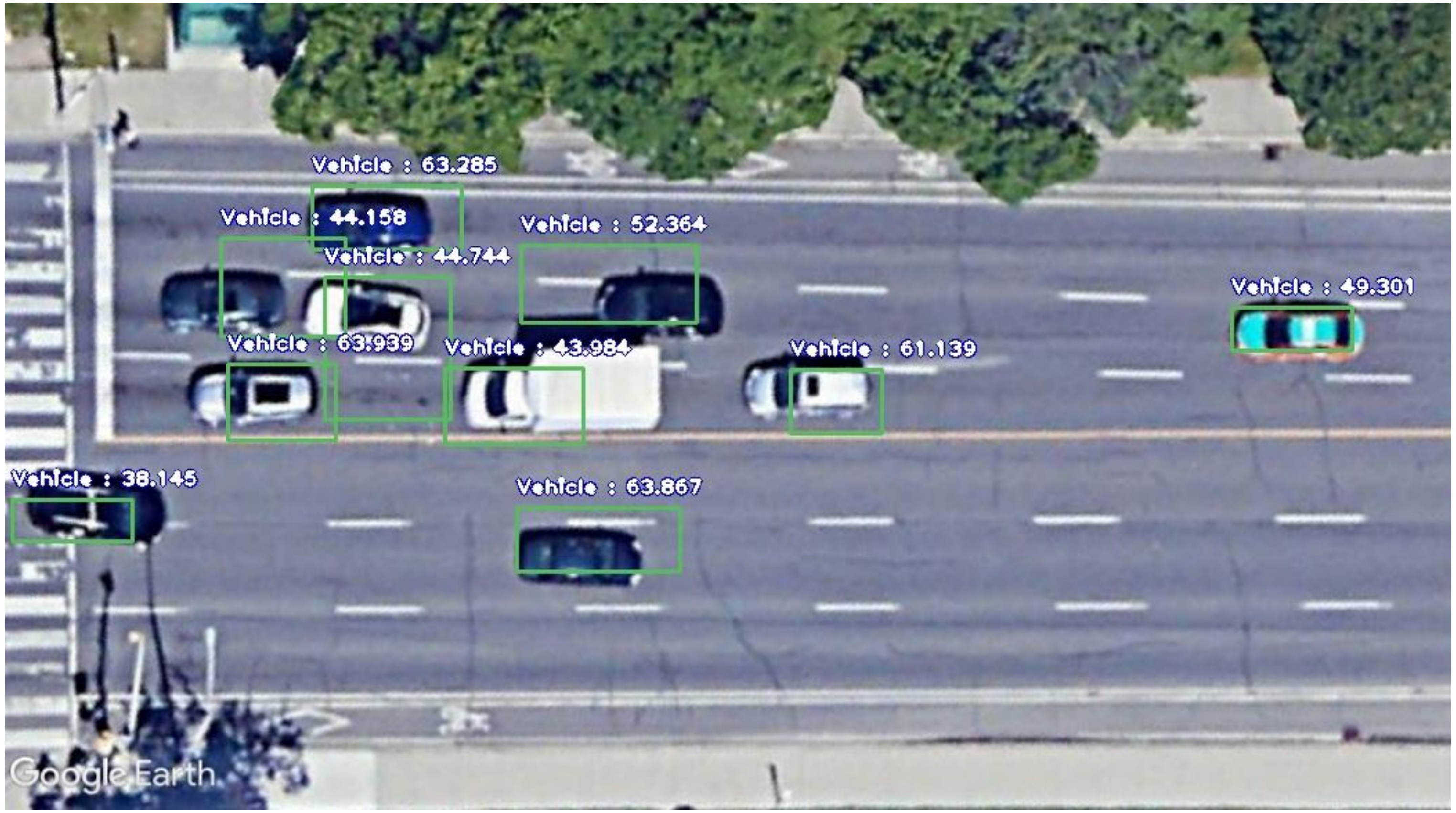
Best Mean Average Precision: 0.12



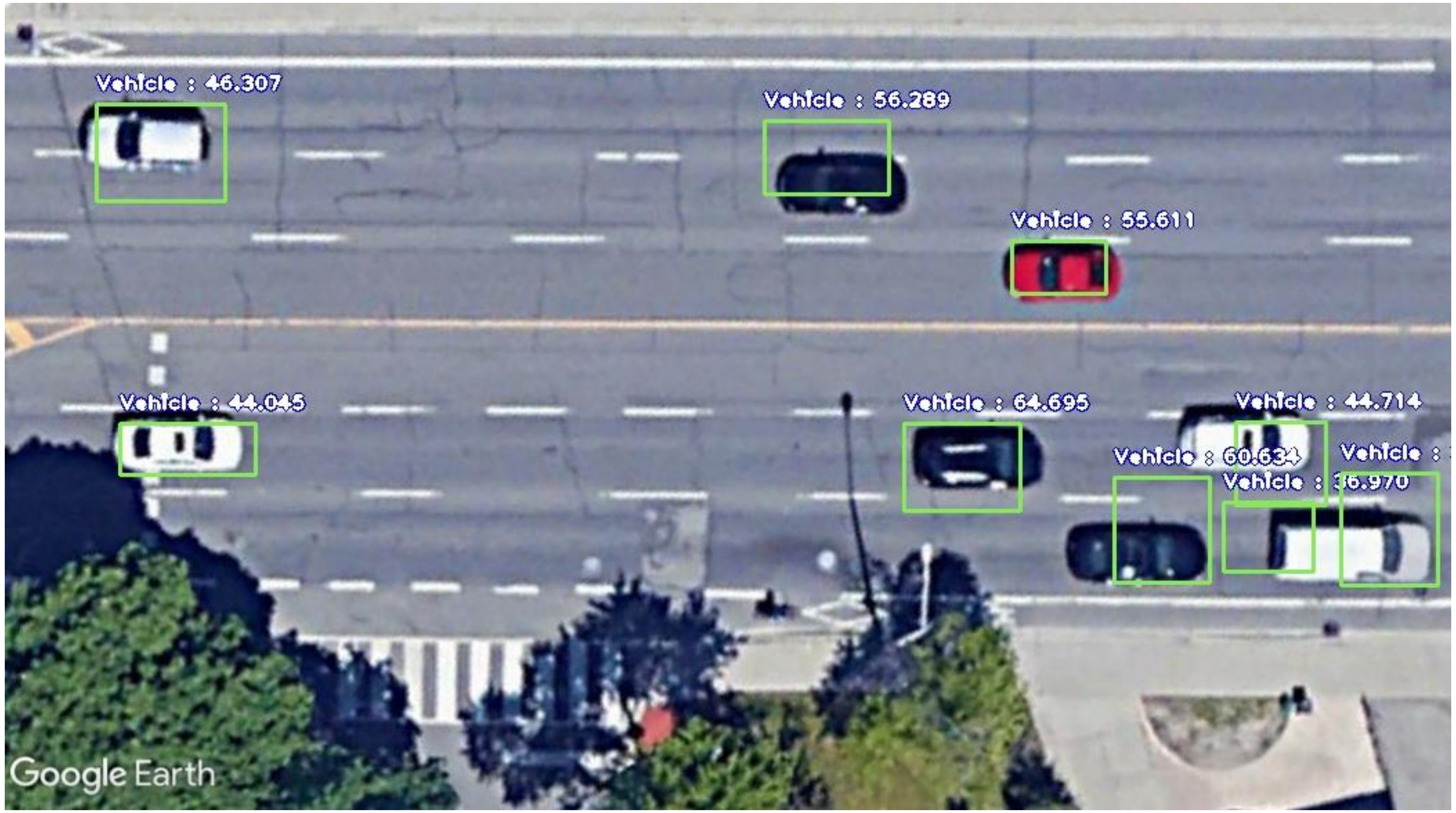
Best Mean Average Precision: 0.1125

Exp 1 vs Exp 2

Test Result – Exp 3



Test Result – Exp 3



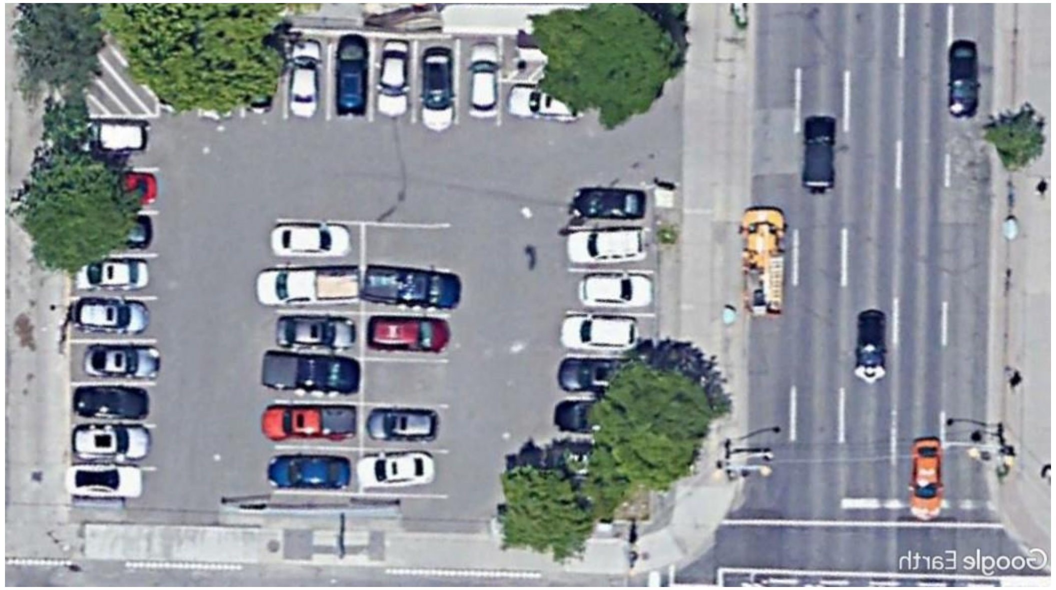
Data Augmentation

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks.

Techniques used for this study:

- Horizontal Flipping (Discussed in workshop)
- Contrast
- Brightness
- Saturation
- Lighting Noise

Experiment 4 – Horizontal flipping



Expt 4: Dataset Distribution

Total: 70 images

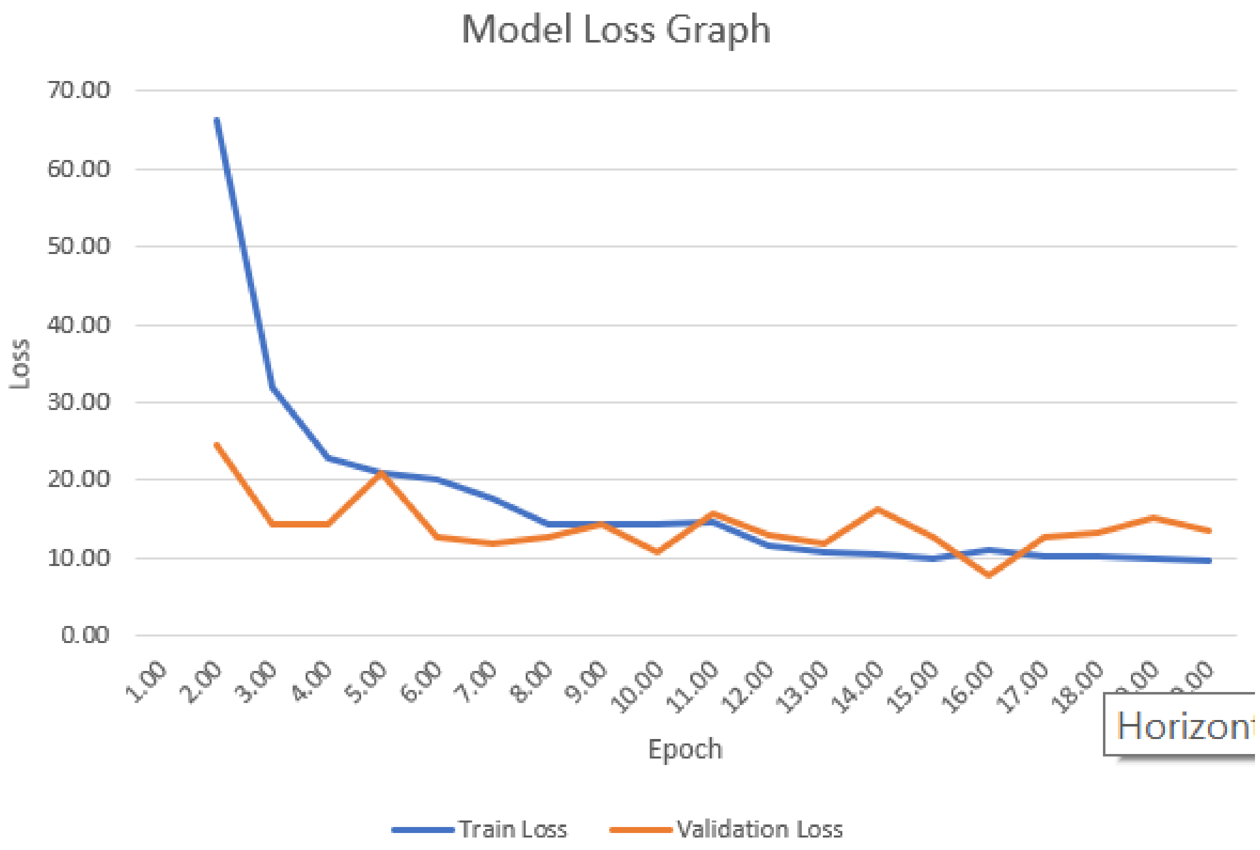
Train: 30 x 2 images

Val: 5 images

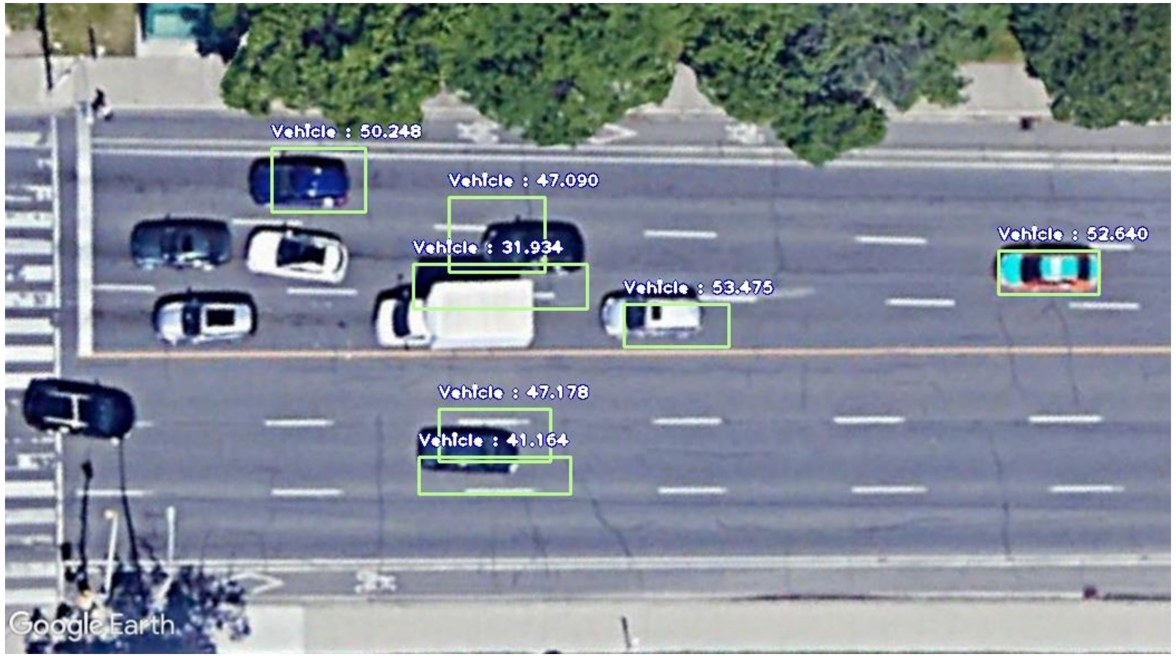
Test: 5 images

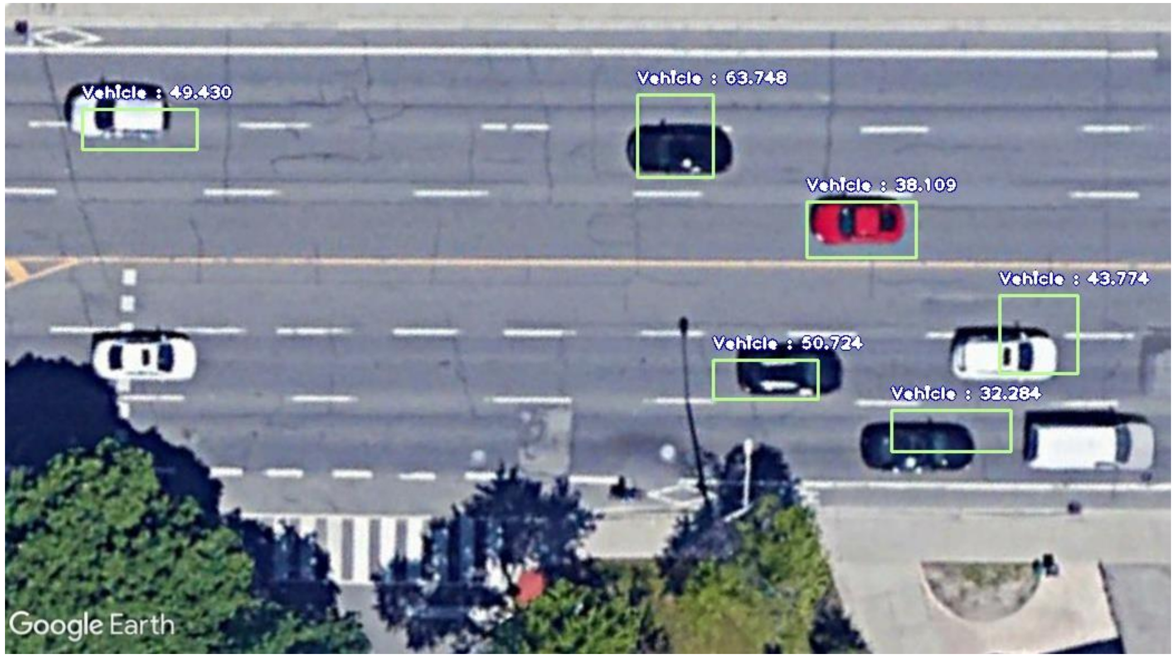
Exp 4: Model Loss Graph

Best Mean Average Precision: 0.80



Exp 4 Test Images

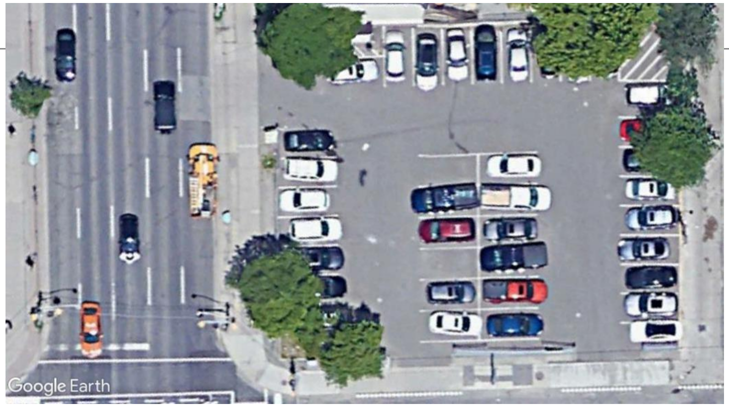




Data Augmentation – Exp 5

- Contrast
- Brightness
- Saturation
- Lighting Noise

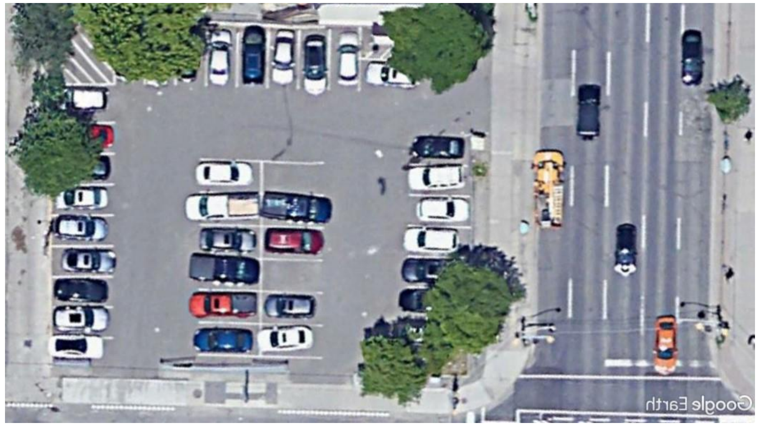
Lighting Noise



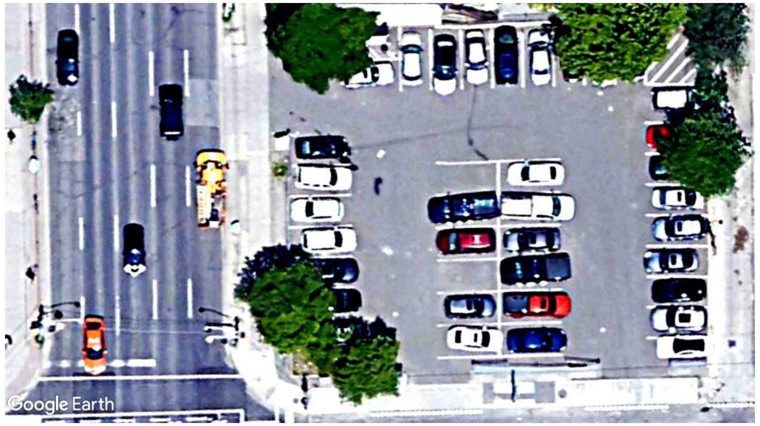
Original

Contrast

Brightness

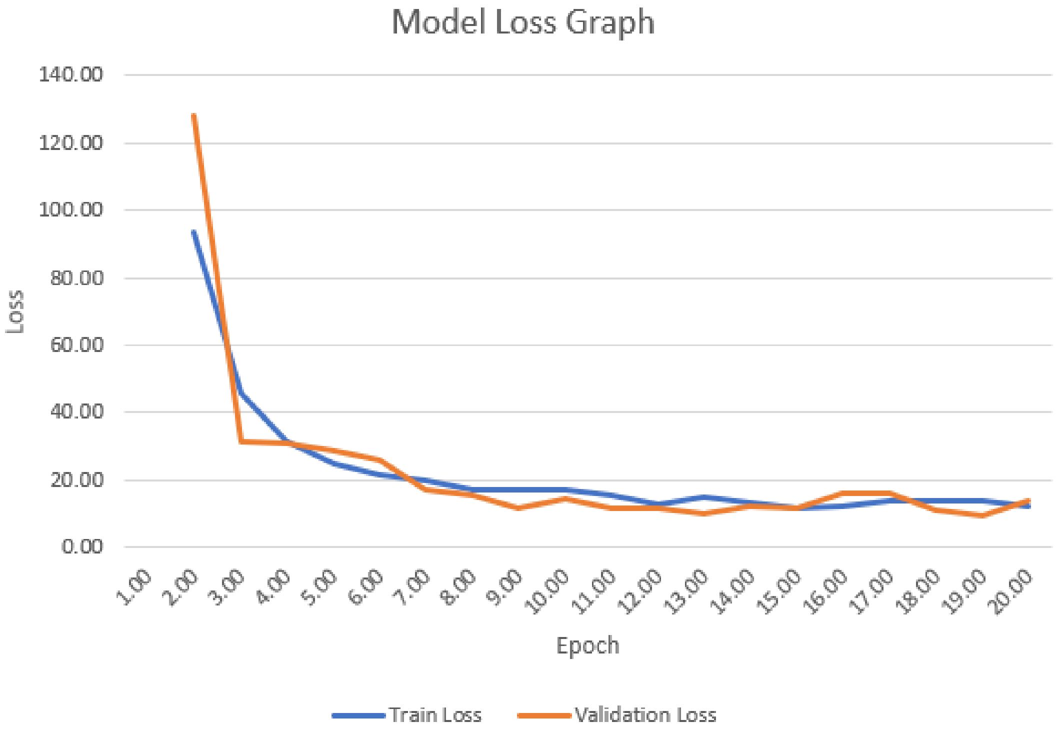


Lighting Noise

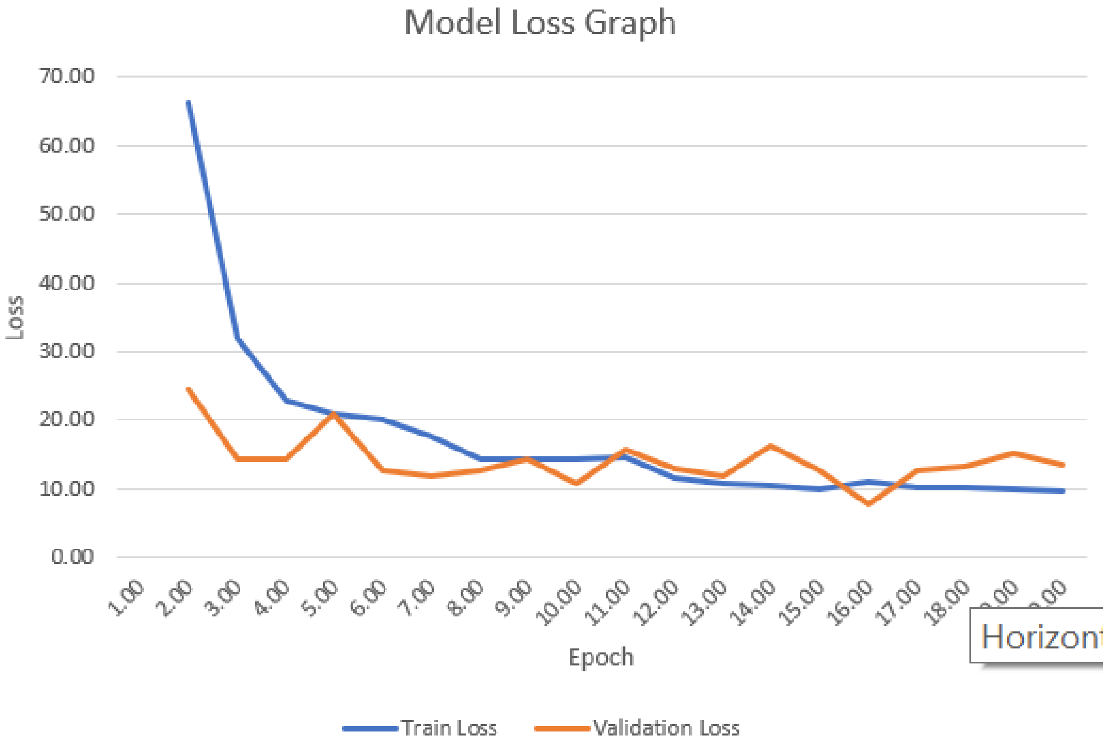


Saturation

Results

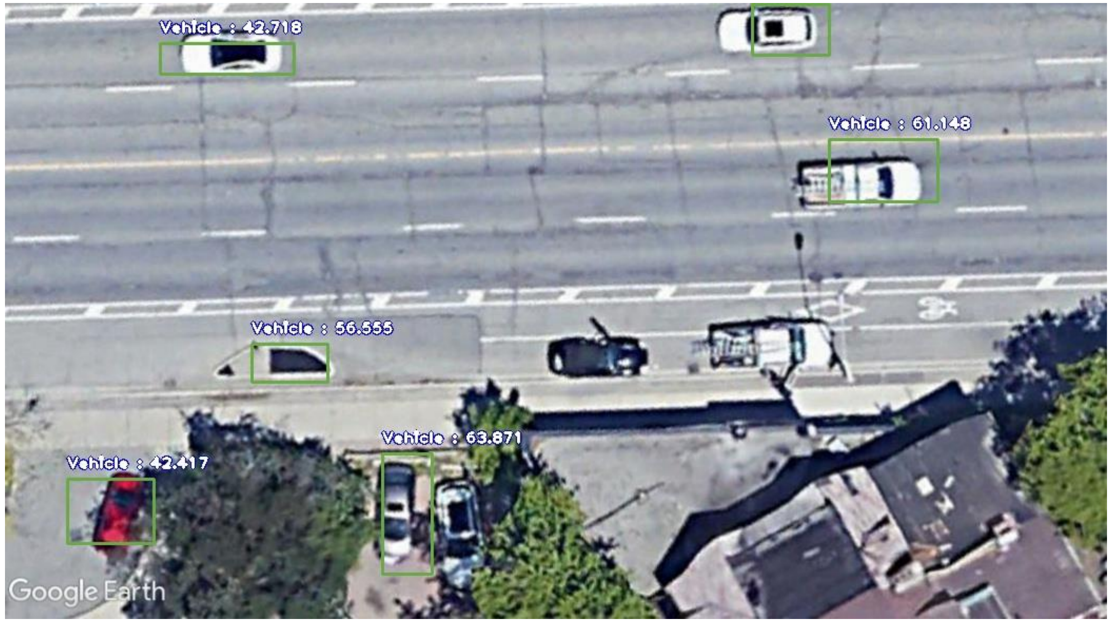


Without horizontal tilt augmentation
Best Mean Average Precision: 0.1125

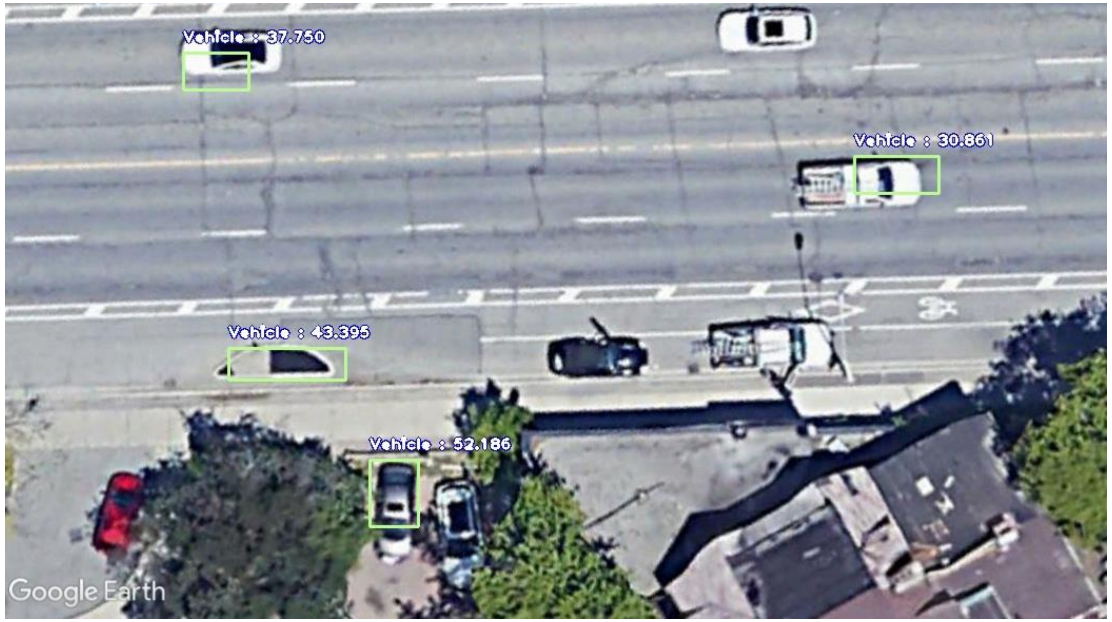


Without horizontal tilt augmentation
Best Mean Average Precision: 0.80

Results

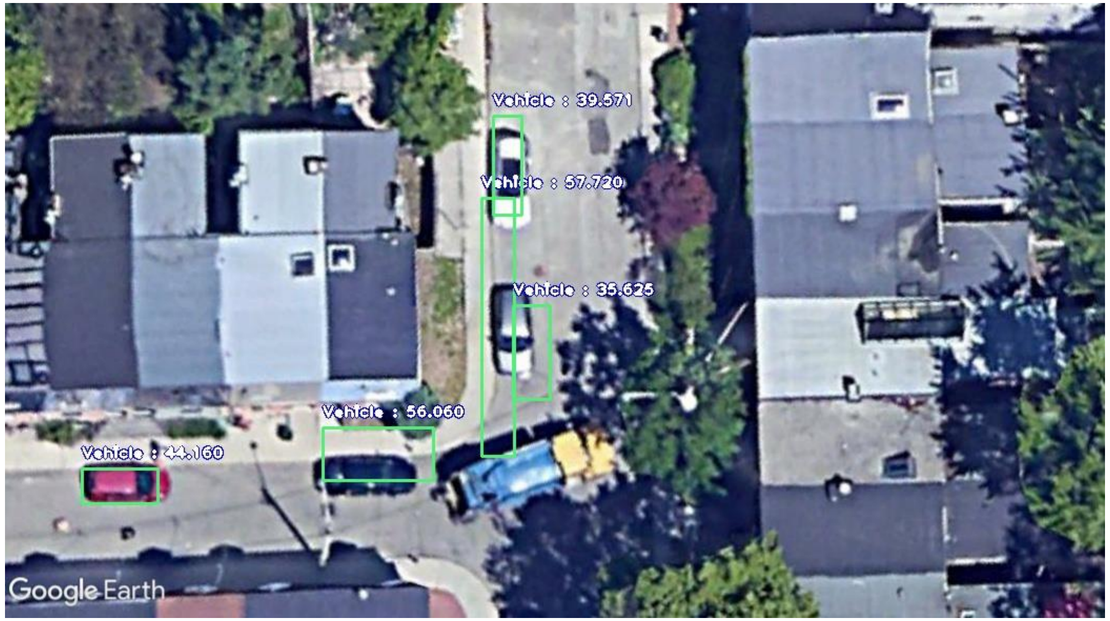


Without horizontal tilt augmentation

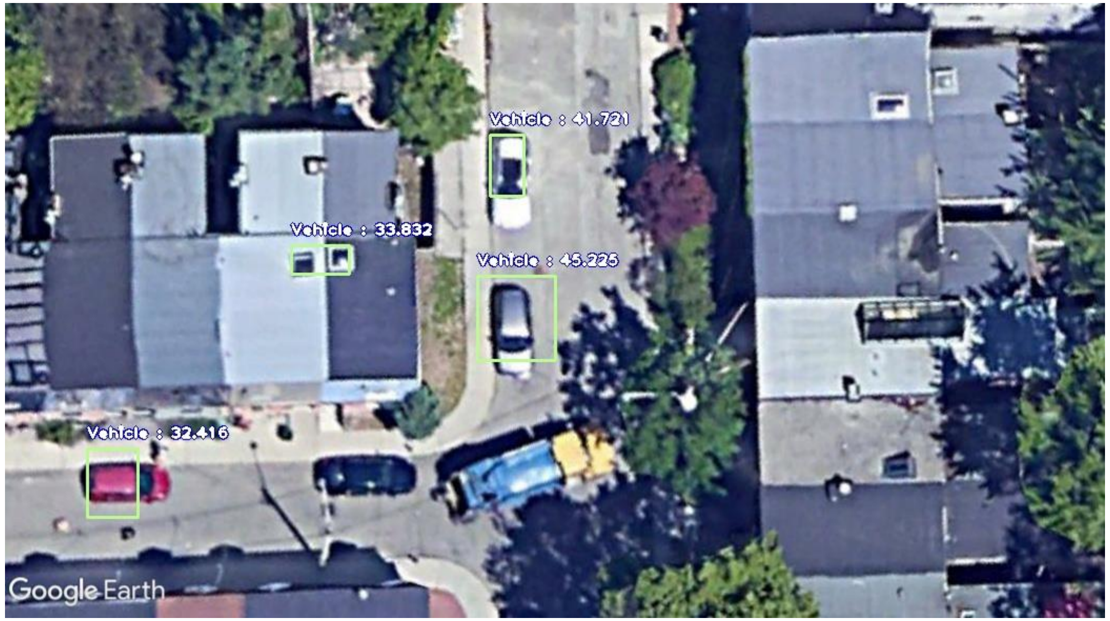


With horizontal tilt augmentation

Results



Without horizontal tilt augmentation



With horizontal tilt augmentation