

Lyft – 3D Object Detection for autonomous vehicles



Project by,
Alisha Fernandes
Haritha Maheshkumar
Khezen Yang
Sijo VM
Thirumurugan Vinayagam



Agenda

- Problem Overview
- Data Details
- Exploratory Data Analysis
- Data Transformation
- Model Building
- Kaggle Competition
- Evaluation and Results
- Challenges and Learnings

1

Problem Overview

Why Object Detection & Why Lyft?



Competition

Tech Companies



Car Manufacturers



Full System or Retrofit



Automation of driving process has evolved over the years and is currently in the 5th stage of evolution – Complete Automation

Driver Assistance



Vehicle can assist with some functions like braking

Partial Automation



Vehicle can assist with steering or acceleration, however, driver is still responsible for critical functions

Conditional Automation



Vehicle controls monitoring of the environment using sensors; human interaction is not required at speeds below 37 MPH

High Automation



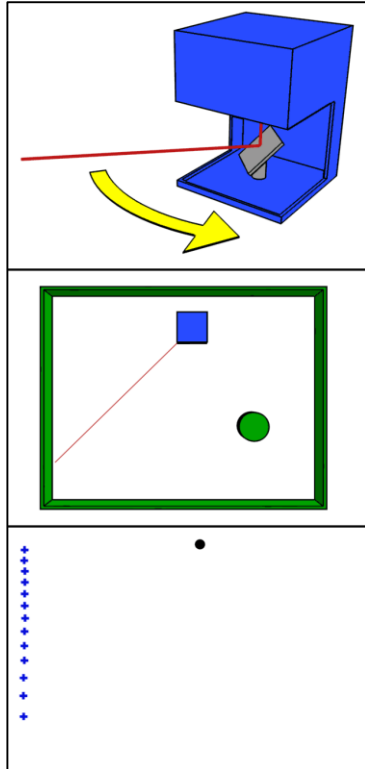
Vehicle can control steering, braking, accelerating, lane changes and use of signals. However, it cannot determine dynamic driving situations

Complete Automation



This level requires no human intervention. The vehicle can identify most unique driving conditions

The LiDAR method consists of multiple sensors which captures light from different parts of the object, and the times recorded by the sensors would be different...



Laser beams are shot in all directions by a laser. The beams reflect off the objects in their path and the reflected beams are collected by the sensor.

The Distance traveled by the light is then calculated using the formula:

$$\text{Distance} = (\text{Speed of Light} \times \text{Time of Flight}) / 2$$

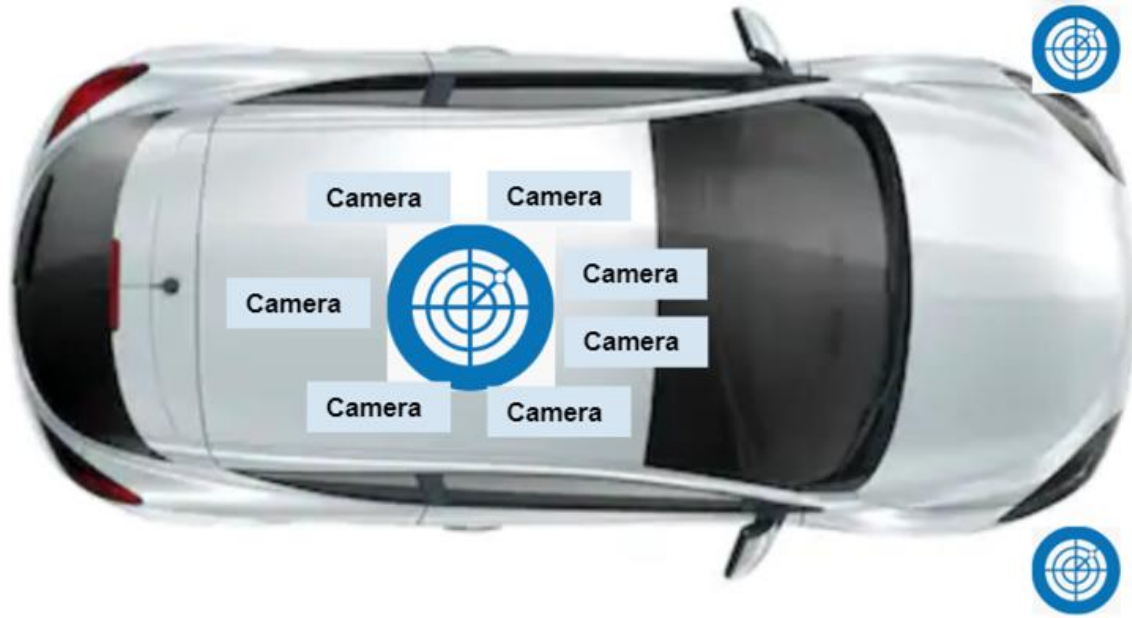
The Flash LiDAR camera is used to create 3D maps using information from these sensors

The signal that is returned is processed by **embedded algorithms** to produce a nearly **instantaneous 3D rendering** of objects and terrain features within the field of view of the sensor

2

Dataset

Our data is captured by 10 host cars, which contain 7 cameras and 3 LiDAR sensors each..



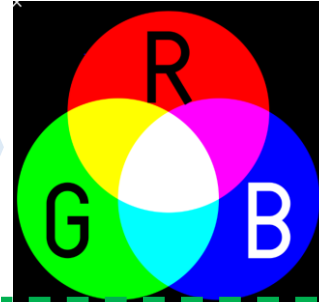
- ✓ Images are captured using 7 cameras
- ✓ Sensor data from 3 LiDAR sensors, producing 216,000 points at 10Hz
- ✓ Data captured from 10 host cars

Snapshots in our dataset capture 2 types of information - Image Data and LiDAR data

scene	25 - 45 seconds of a car's journey in a given environment. Each scene is composed of many samples.
sample	A snapshot of a scene at a particular instance in time. Each sample is annotated with the objects present.
sample_data	Data collected from a particular sensor on the car.
sample_annotation	An annotated instance of an object within our interest.
instance	An enumeration of all object instance we observed.
category	Taxonomy of object categories (e.g. vehicle, human).
attribute	Property of an instance that can change while the category remains the same.
visibility	(currently not used)
sensor	A specific sensor type.
calibrated sensor	Definition of a particular sensor as calibrated on a particular vehicle.
ego_pose	Ego vehicle poses at a particular timestamp.
log	Log information from which the data was extracted.
map	Map data that is stored as binary semantic masks from a top

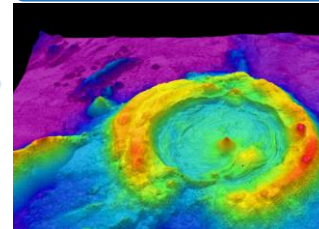


Image Data



- ✓ The 3 channels superimpose to form the final colored image
- ✓ They can be stored in 4-dimensional tensor with dimensions as batch_size, channels, width and height

LiDAR Data



- ✓ LiDAR method is used in generating accurate 3D representations of surroundings using laser light

The points captured by the LiDAR are used to estimate the x, y, z coordinates and derive the length, width and height measures

Measures of objects recorded by the LiDAR sensors

1 **Centre_x and centre_y** correspond to the x and y coordinates of an object's location (bounding volume). These coordinates represent the object's location on the xy plane

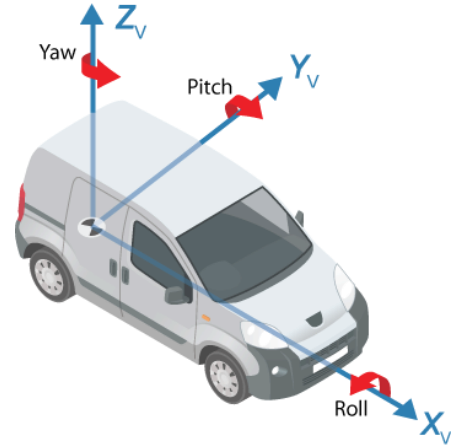
2 **center_z** corresponds to the z coordinate of the center of an object's location (bounding volume). This coordinate represents the height of the object above the xy plane.

3 **yaw** is the angle of the volume around the z -axis, making 'yaw' the direction the front of the vehicle / bounding box is pointing at while on the ground.

4 **width** is the width of the bounding volume in which the object lies.

5 **length** is the length of the bounding volume in which the object lies

6 **height** is the height of the bounding volume in which the object lies

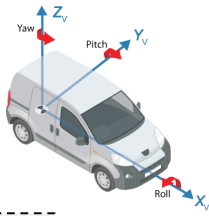


...Let us look at a few distributions of these measures across different objects on road

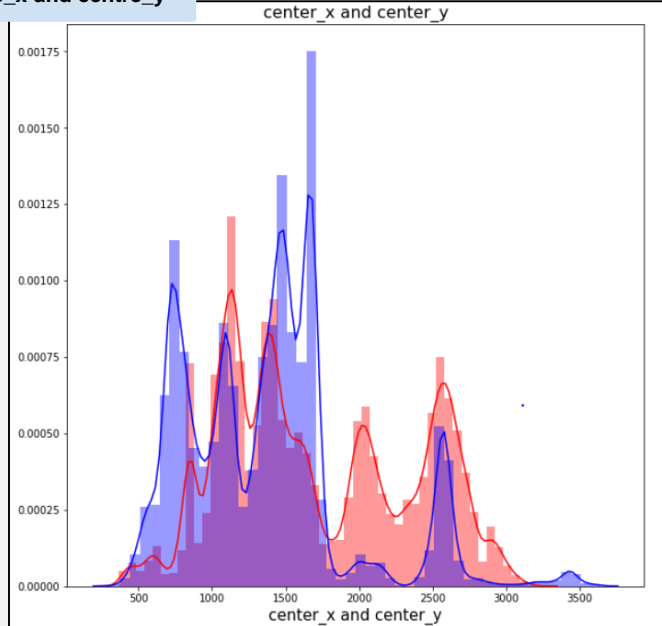
3

Exploratory Data Analysis

The distributions of center.x and centre.y are able to bring out the limitations of the LiDAR cameras

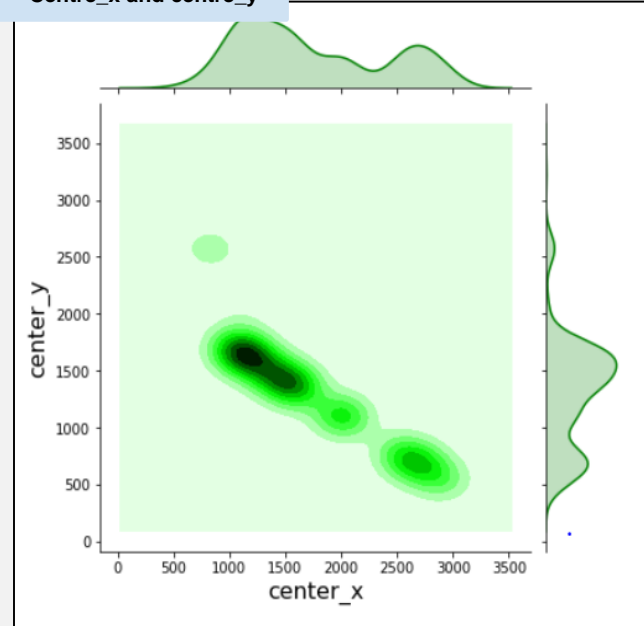


Centre_x and centre_y



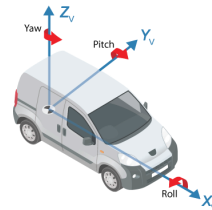
Probably because the car's camera can sense the objects on either left or right easily due to the small width of the road, when compared to the length and there is a higher chance of the camera's view being blocked

Centre_x and centre_y

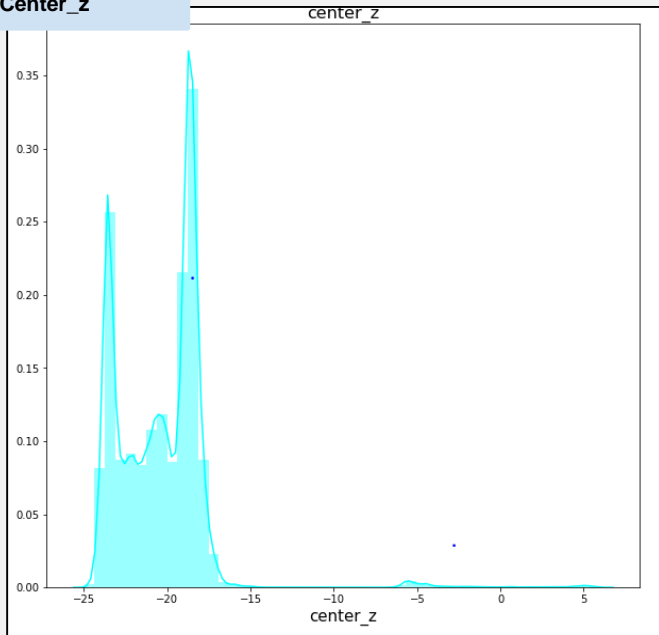


This could be because the camera could detect either objects which are too far ahead or objects which are too far to the side but not both

The `centre_z` distribution is extremely right skewed and is clustered around the -20 mark, as most objects are very close to the flat plane of the ground

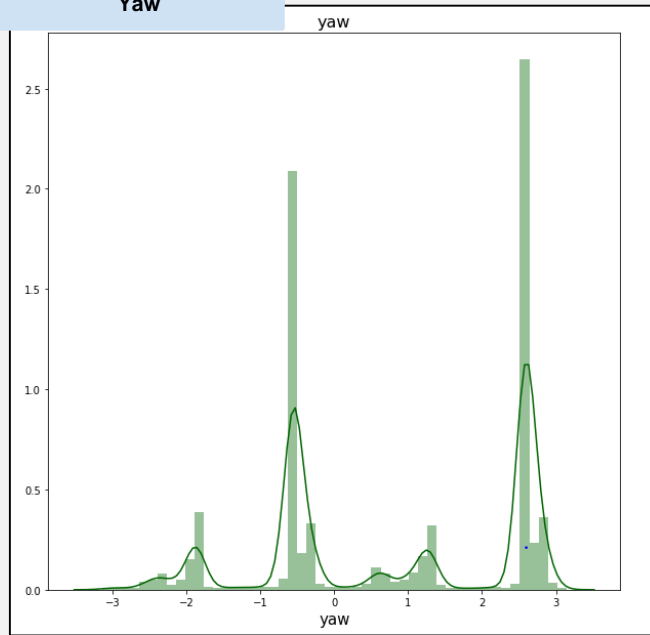


Center_z



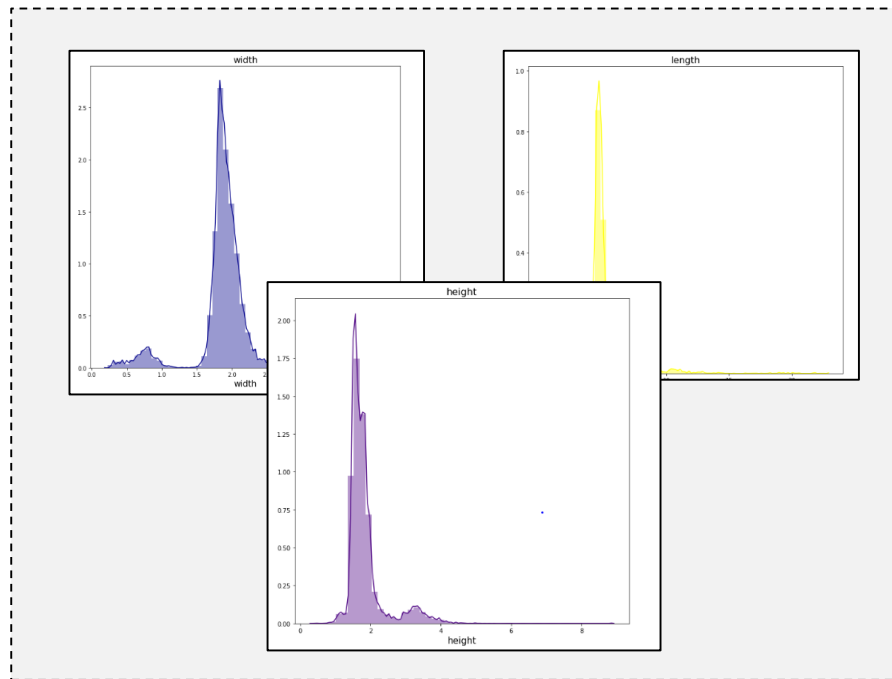
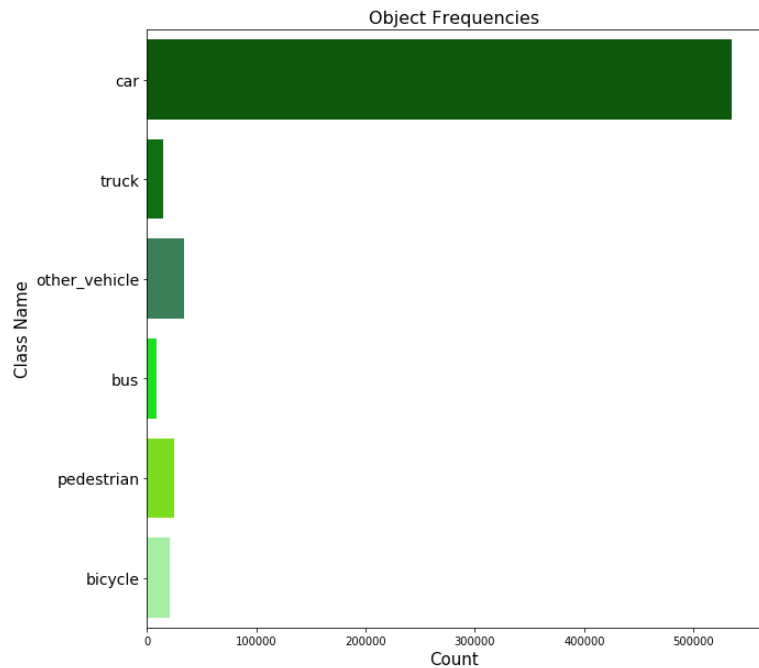
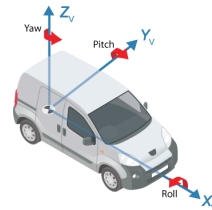
Also, most `z` coordinates are negative because the camera is attached at the top of the car. So, most of the times, the camera has to "look down" to see the objects. Therefore, the height or `z`-coordinate of the objects relative to the camera are generally negative.

Yaw



Yaw is the angle of the volume around the `z`-axis, making yaw the direction of the front of the vehicle/bounding box is pointing while on the ground

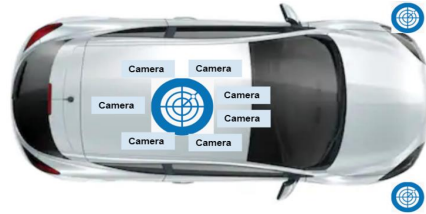
Since most of the objects are cars, the distributions of length, width and height of the objects are skewed to a particular range, with outliers representing either large sized trucks or tiny pedestrians and cyclists



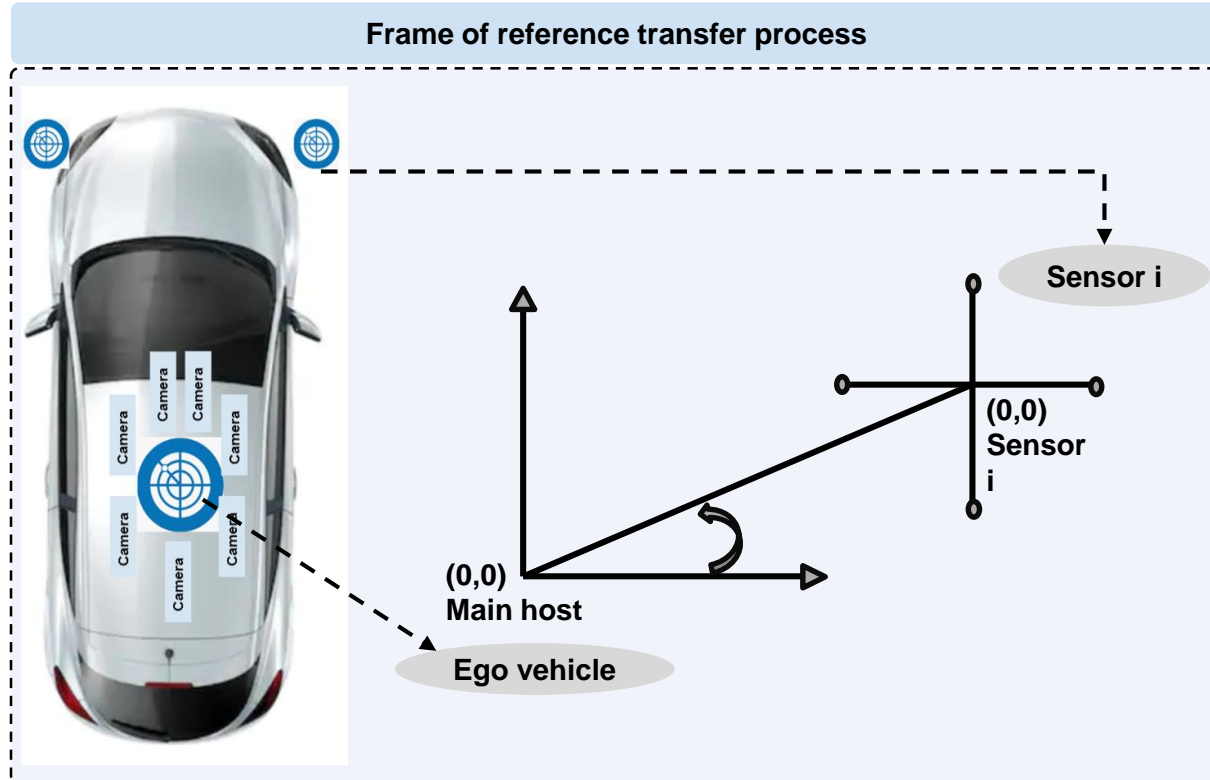
4

Data Transformation

Below are the images of a scene that were rendered

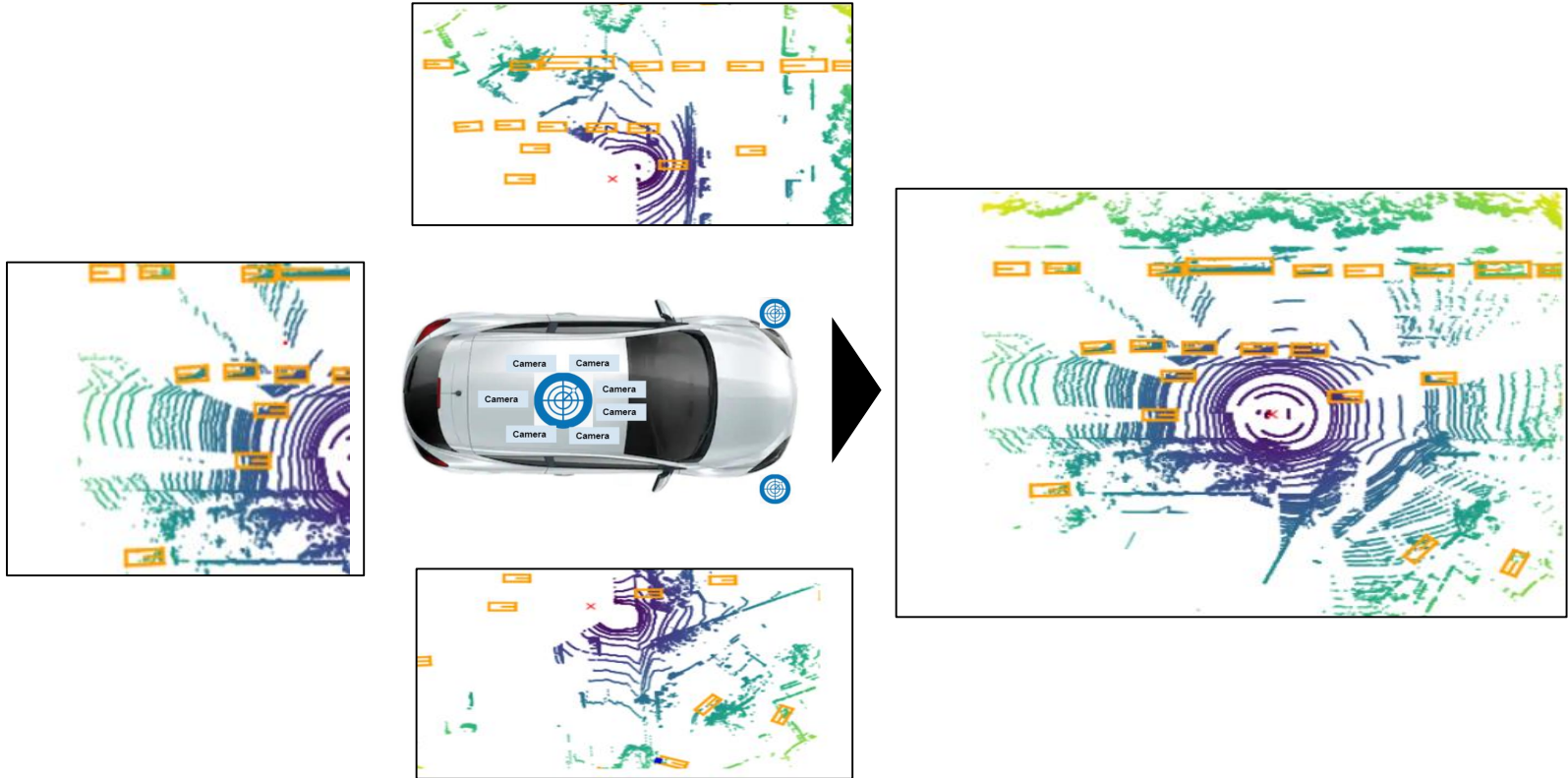


The output of the data transformation phase is a point cloud which defines the position of the objects in the ego vehicle's frame of reference



- ✓ The 3D point cloud from LIDAR provides accurate depth and reflection intensity
- ✓ This, when fused with the 2D image bounding box will result in a bounding box with higher accuracy
- ✓ These are fed as inputs to the modelling process

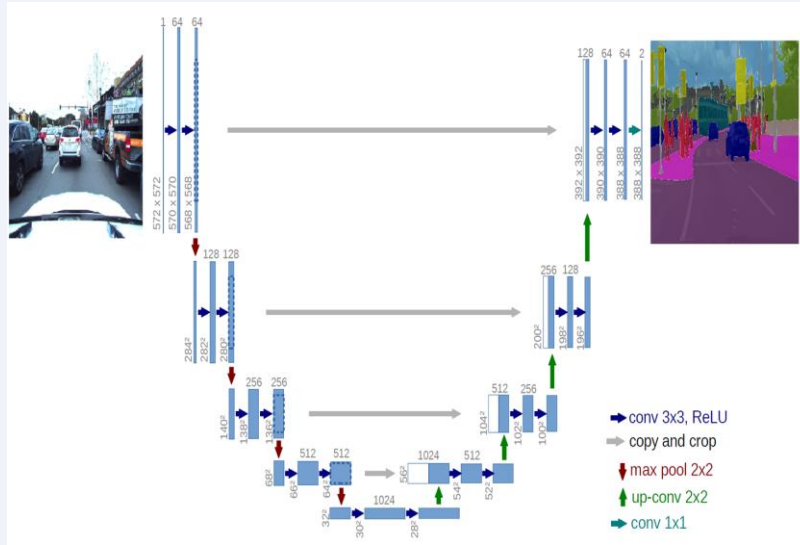
Below is an example of the **Point cloud transformation** done by the LiDAR



5

Model Building

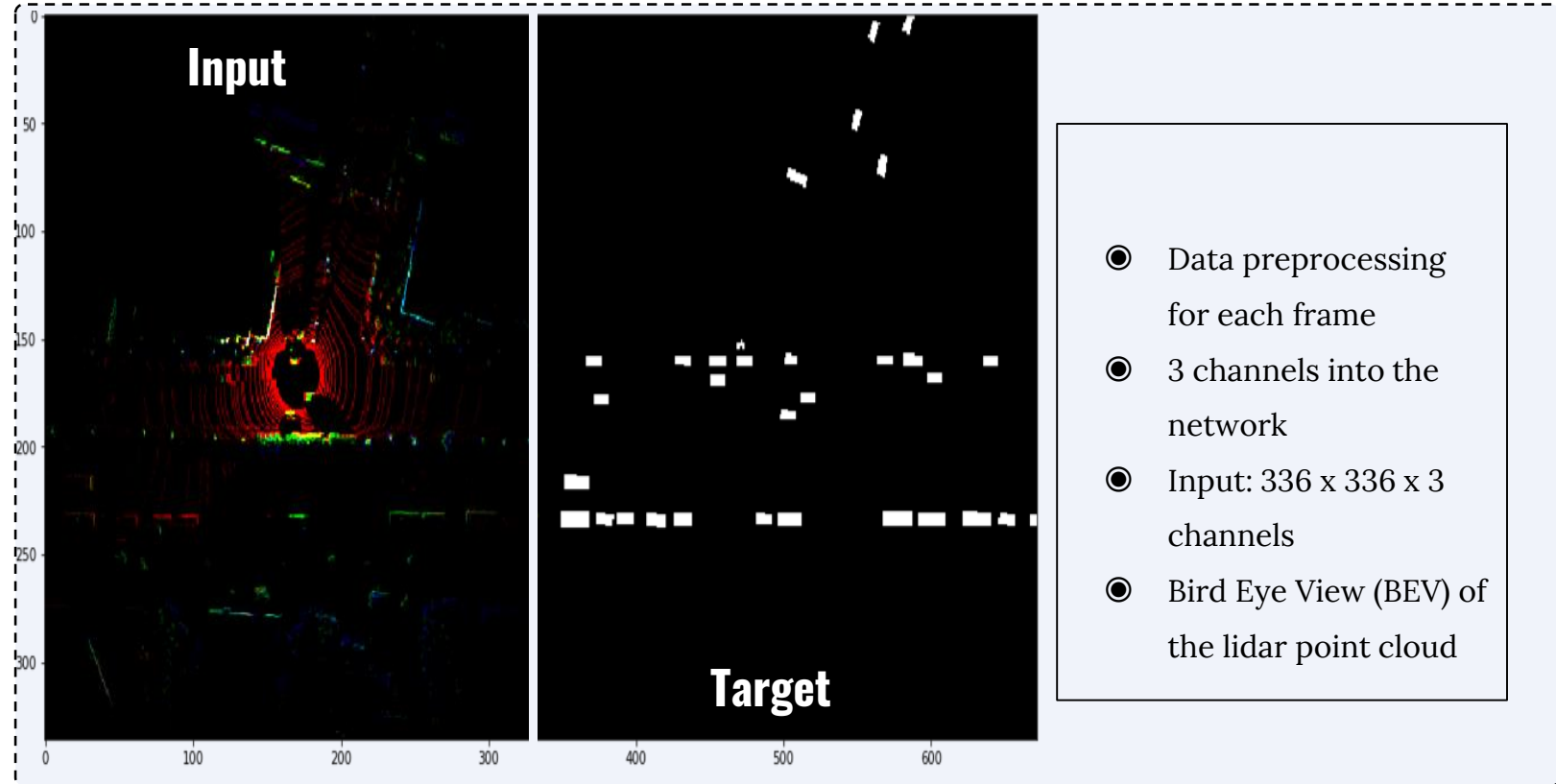
U-Net for semantic segmentation



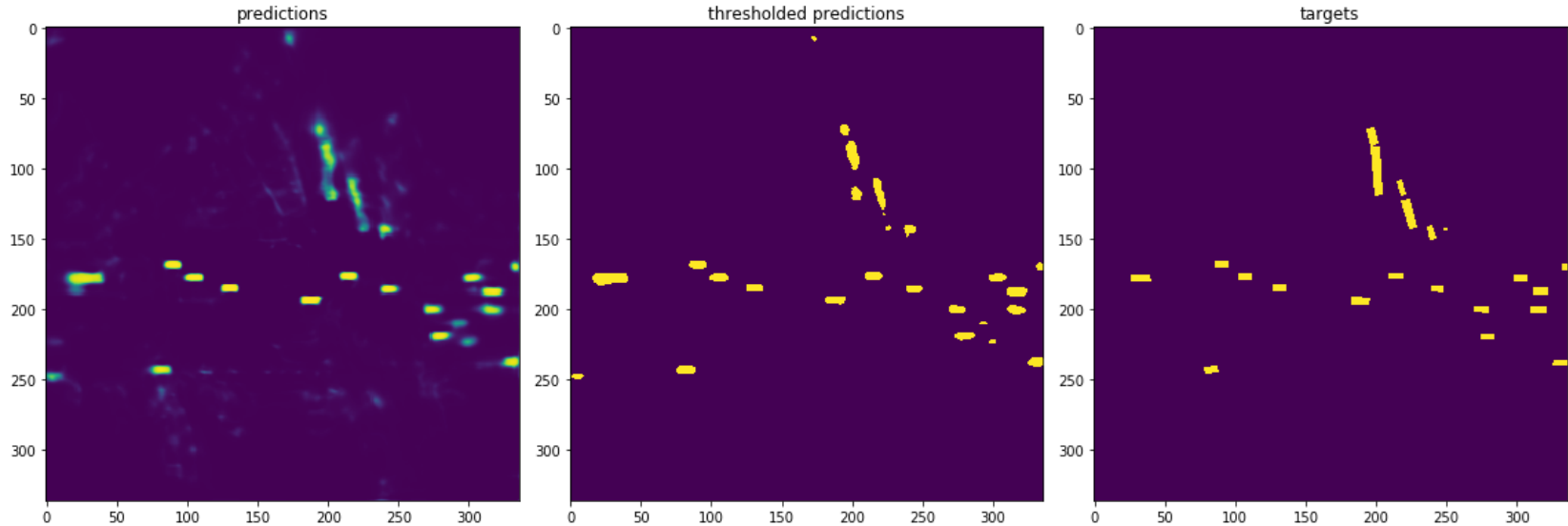
Architecture:

1. The contracting/downsampling path
 2. Bottleneck:
 3. The expanding/upsampling path
- U-net fully convolutional neural network
 - Predicts the objects for each pixel in BEV
 - Morphological transformations to fit boxes
 - Transformations to fit boxes in the world space

The U-Net model had the following inputs and targets



Thresholding and Morphological Transformation



Model Evaluation: Mean Average Precision (mAP)

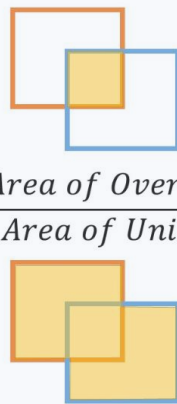
Intersection over Union (IoU):

Different thresholds (0.5 \rightarrow (0.95)
Step size = 0.05
3D Context: Z overlap in 1D

$$IoU(A, B) = \frac{A \cap B}{A \cup B} > tr$$

$$Intersection\ over\ Union\ (IoU) = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

— Prediction
— Ground-truth



Mean Average Precision (mAP) across different classes

$$\frac{1}{|thresholds|} \sum_t \frac{TP(t)}{TP(t) + FP(t)}$$

kaggle

Modifications that mattered! : mAP 0.034 to 0.039

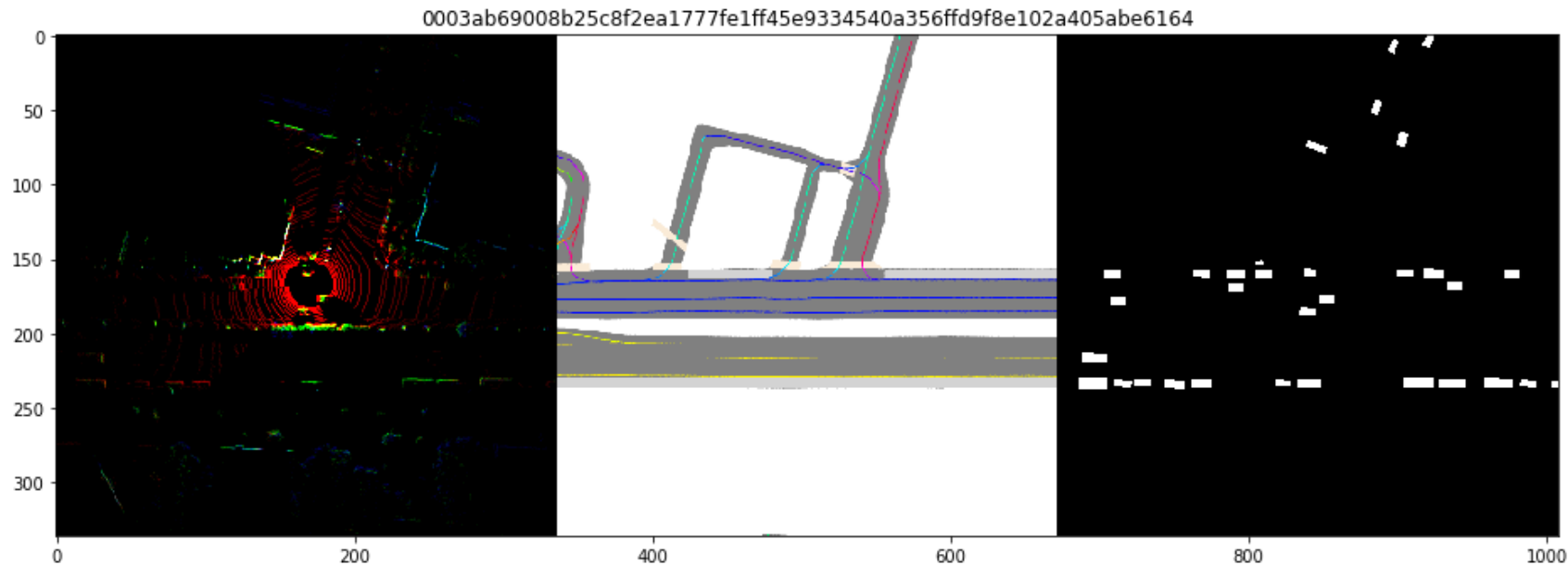
- Imbalanced classes - More cars than any other classes
- Height of all classes was considered the same, i.e. mean of all classes
- Considers only Lidar Data



- Reduce class weights while training using cross entropy / softmax loss
- Height of the class was modified to be the conditional mean of individual classes
- Changed parameters in the UNET Architecture (optimizer, sampling, batch size)

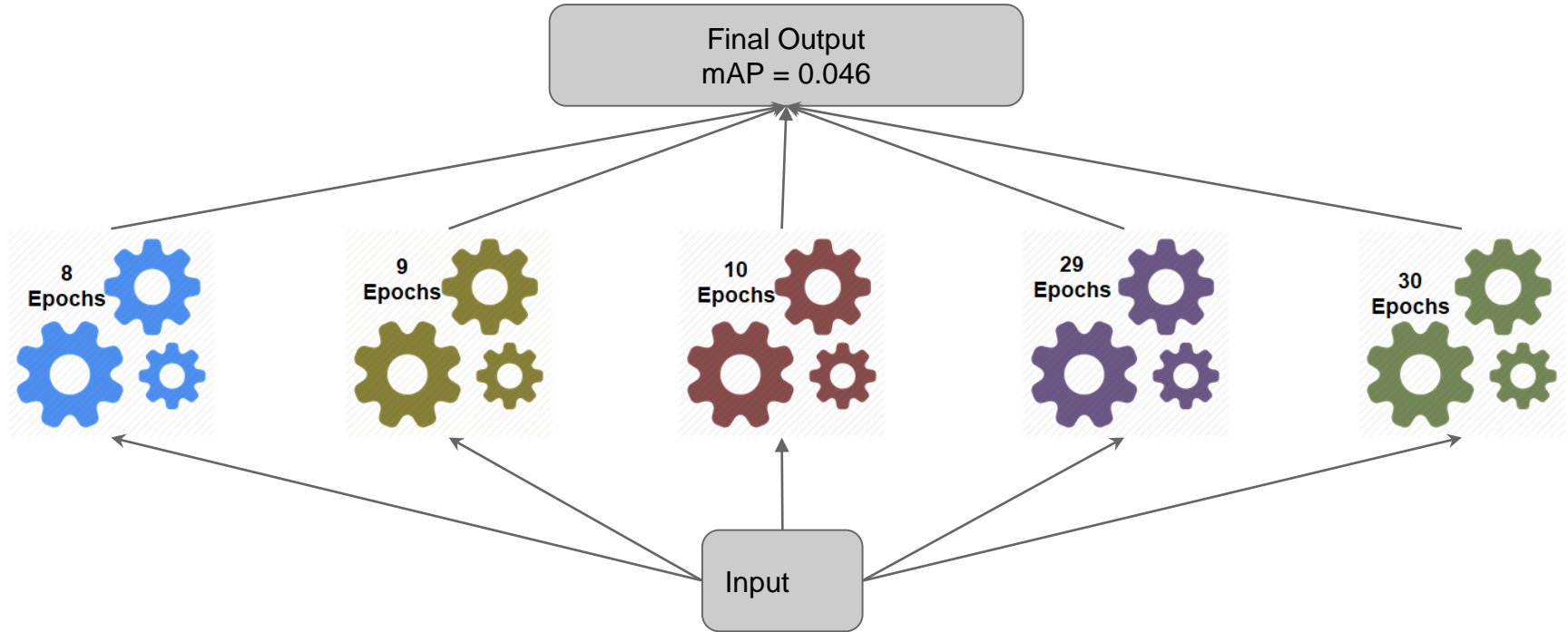


Bird Eye View (BEV) + Map Masks: mAP 0.040



The original 3 input channels (RGB) increased to 6 input channels

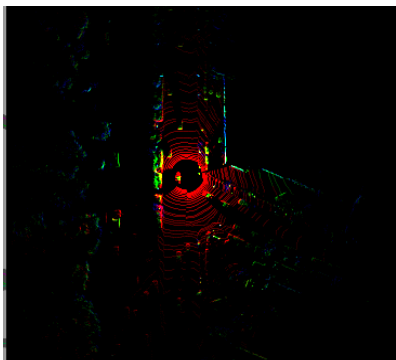
Ensembles from different epochs



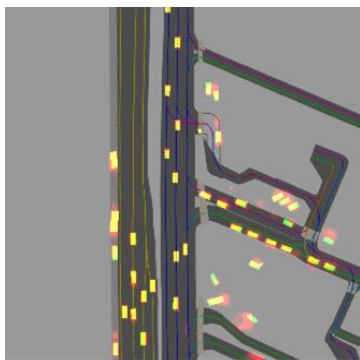
6

Evaluation and Results

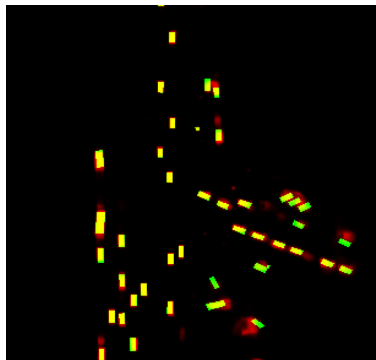
Inference and Predictions



Input (BEV)



Input (map mask)



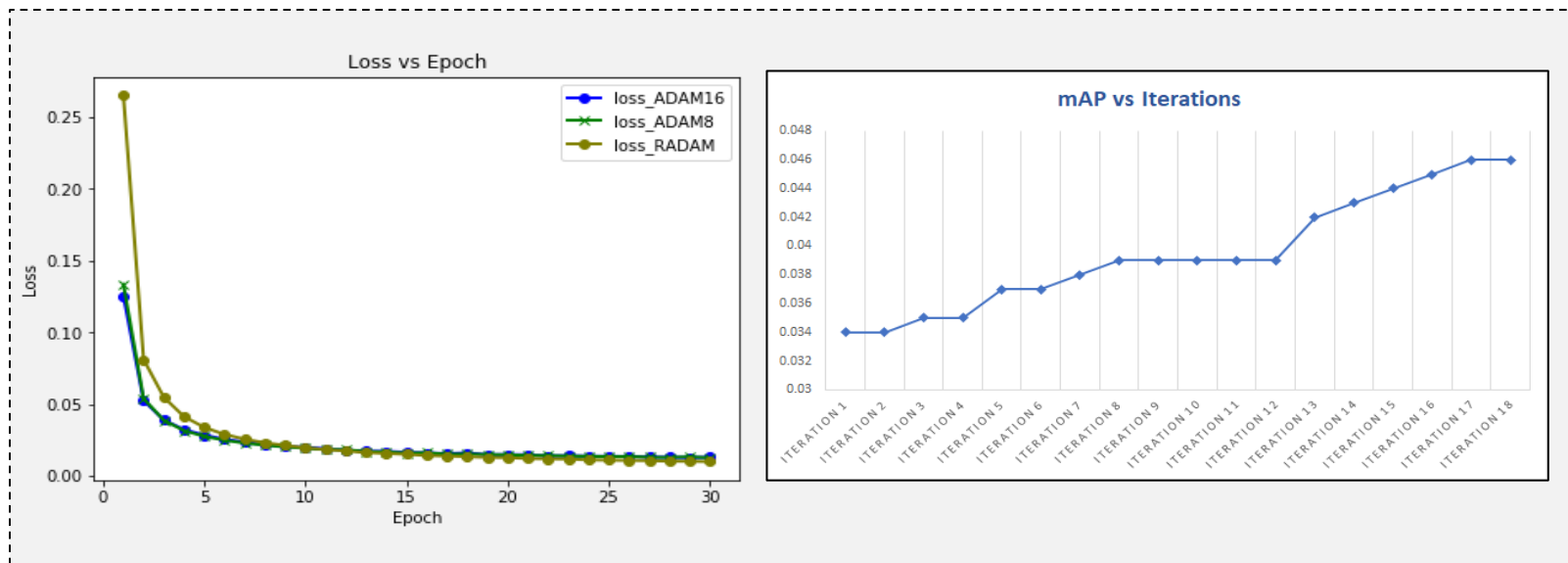
Predictions



Predictions

Green: False negative (target)
Yellow: True Positive
Red: False Positive (Prediction)

Consolidated results of 18 different model iterations



7

Kaggle Competition








Kaggle Leaderboard

Your most recent submission

Your most recent submission

Your most recent submission

Your most recent submission

87	Val An		0.046	21	19d
88	LouisClouatre		0.046	31	6d
89	David		0.046	10	7h
90	Sijo VM		0.046	13	3m
Your Best Entry ↑ You advanced 11 places on the leaderboard! Your submission scored 0.046, which is an improvement of your previous score of 0.045. Great job!  Tweet this!					
91	Antonio Marin		0.045	24	7d
92	bobbqe		0.045	72	8h

Final Leaderboard

- Top 20% among 547 Teams
- 0.045 on the private leaderboard
- 18 submissions
- First place: mAP 0.216

Sample Prediction string:

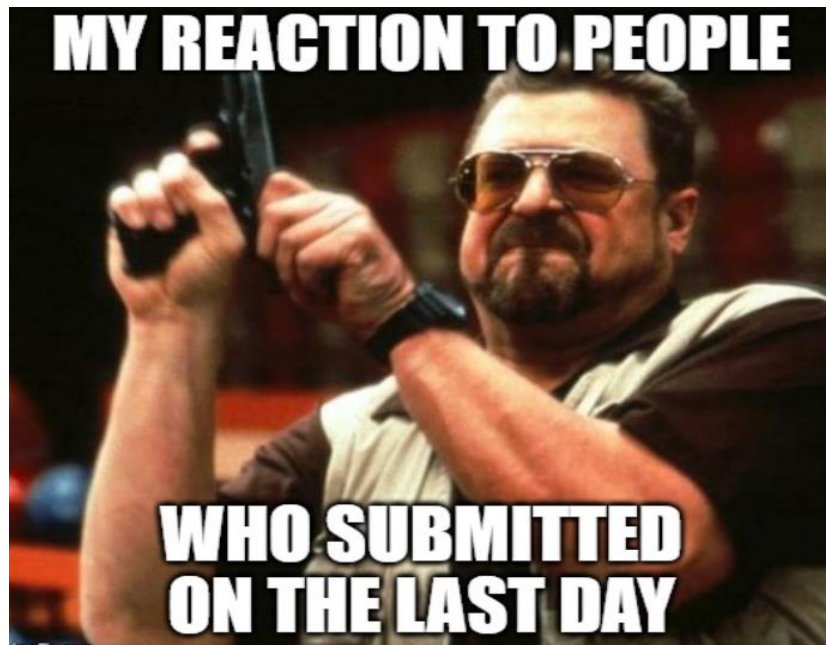
97ce3ab08ccbc0baae0267cbf8d4da947e1f11ae1dbcb80c3f440

8784cd9170c,1.0 2742.15 673.16 -18.65 1.834 4.609

1.648 2.619 car

Confidence score| X, Y Z | Width, Length, Height,

yaw, Class



Lyft 3D Object Detection for Autonomous Vehicles

Can you advance the state of the art in 3D object detection?

Featured · 16 days ago · image data, object detection



108/547

Top 20%

Predictions Video



https://www.youtube.com/watch?v=5LN6mFjK6go&feature=emb_title

8

Challenges and Learnings

Challenges and Learnings

- Could not do cross validation due to memory constraints, even using Google Cloud
- Could not increase the batch size beyond 32 due to the same issue
- 30 Epochs took around 16 - 18 hours to train batch size 16 , around 30 - 40 minutes to ensemble 3-4 models → around 20 hours for one submission

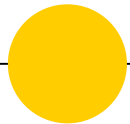
- Model still fails to predict the small objects from the BEV eg. pedestrians
- The model assumes flat surface: all the objects are at the same height as that of the ego vehicle
- User only one Lidar Sweep - use images and other lidar sweeps to make accurate predictions
- With better computational power, the model can be tuned efficiently
- Could try models with different architectures



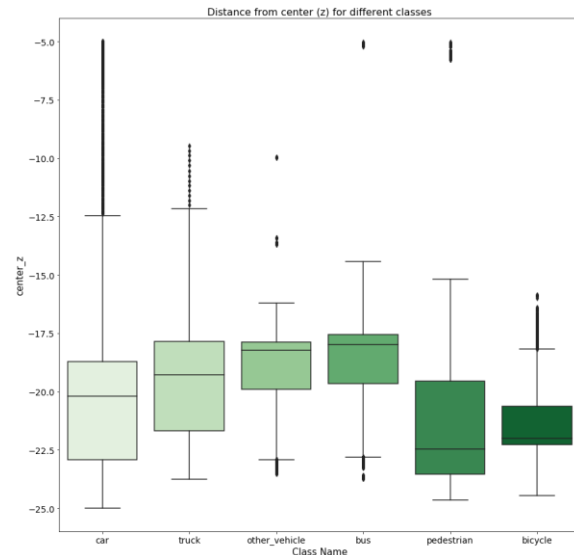
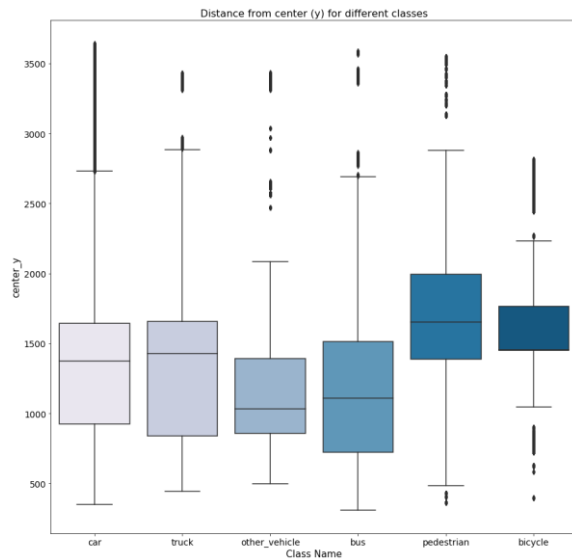
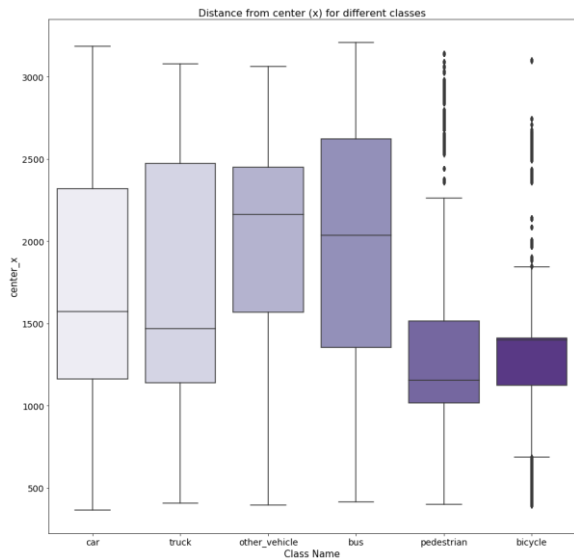
Thanks!

Any **questions** ?

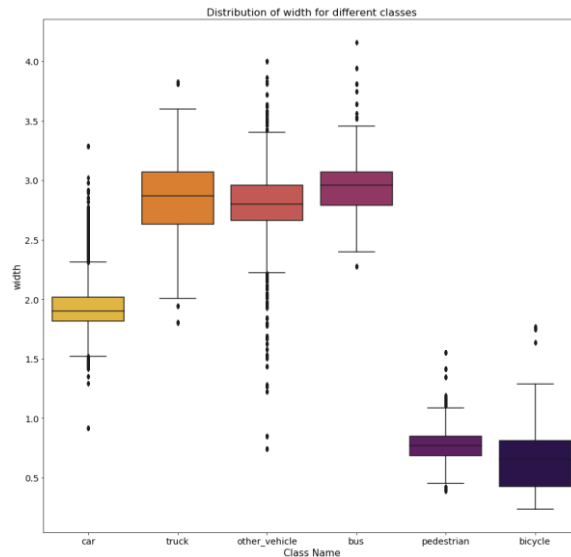
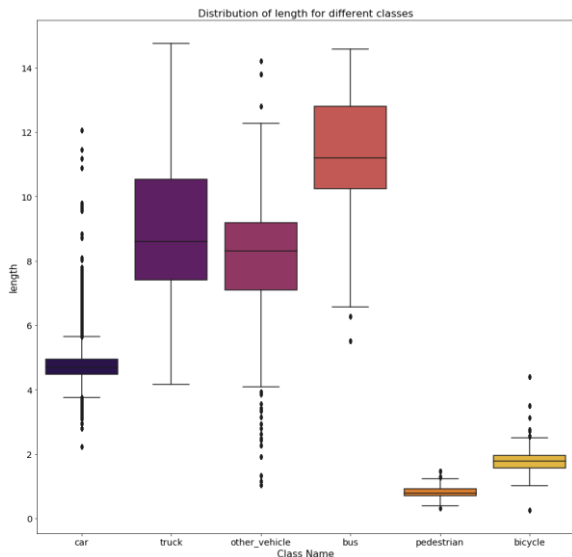
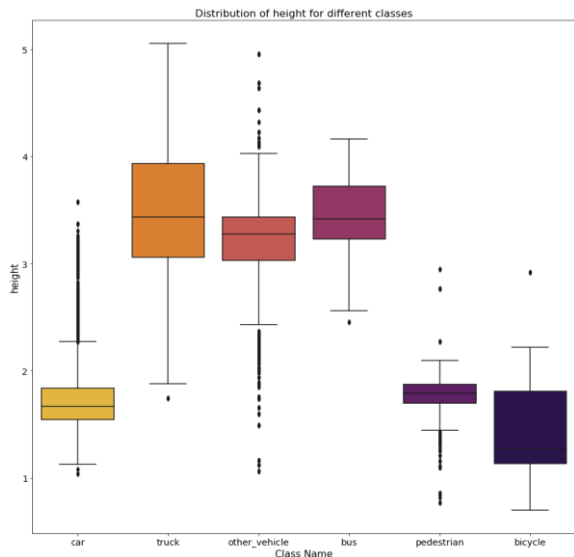
Appendix



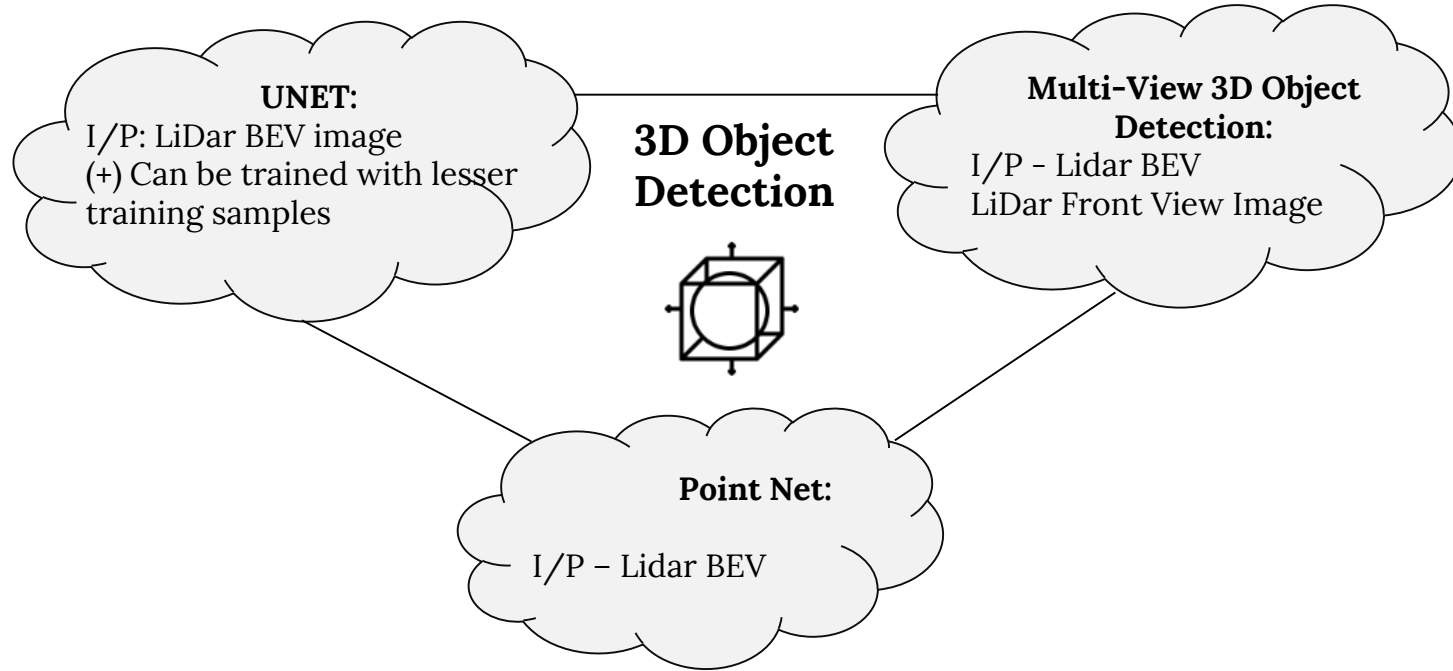
We also observed that the larger objects get detected more because of their large size when compared to the smaller ones, and are also likely to be far away from the hosts



The length, width and height distributions of different classes of objects are indicative of their sizes

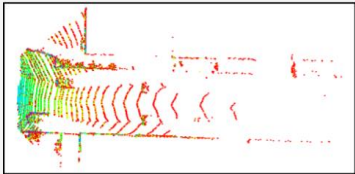
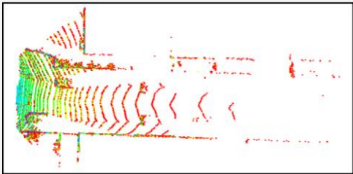
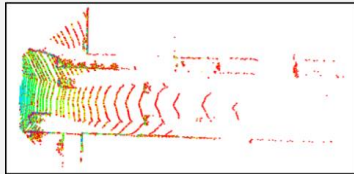


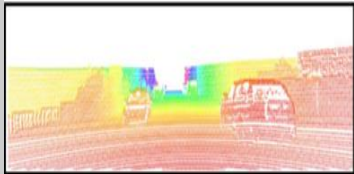


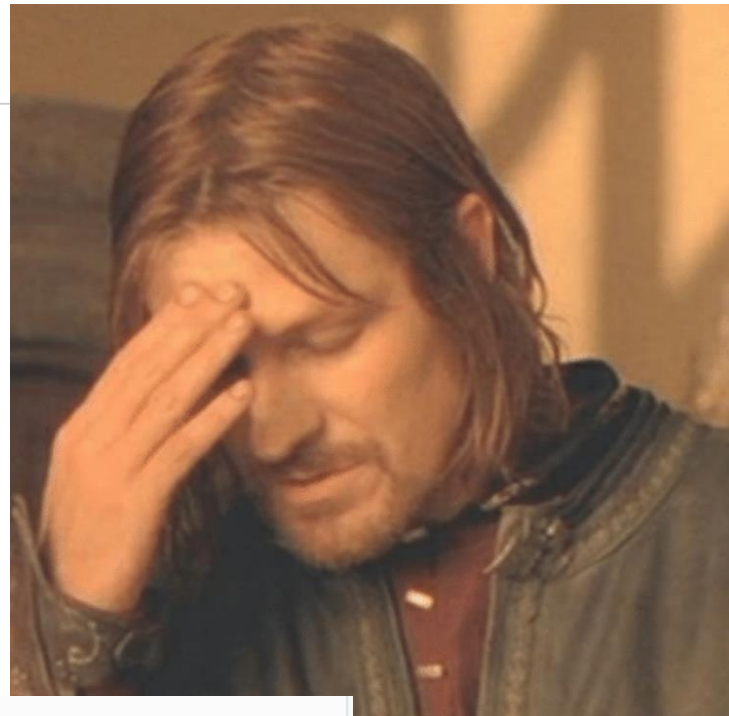
Different models used for 3D object detection take in different types of inputs





Different models

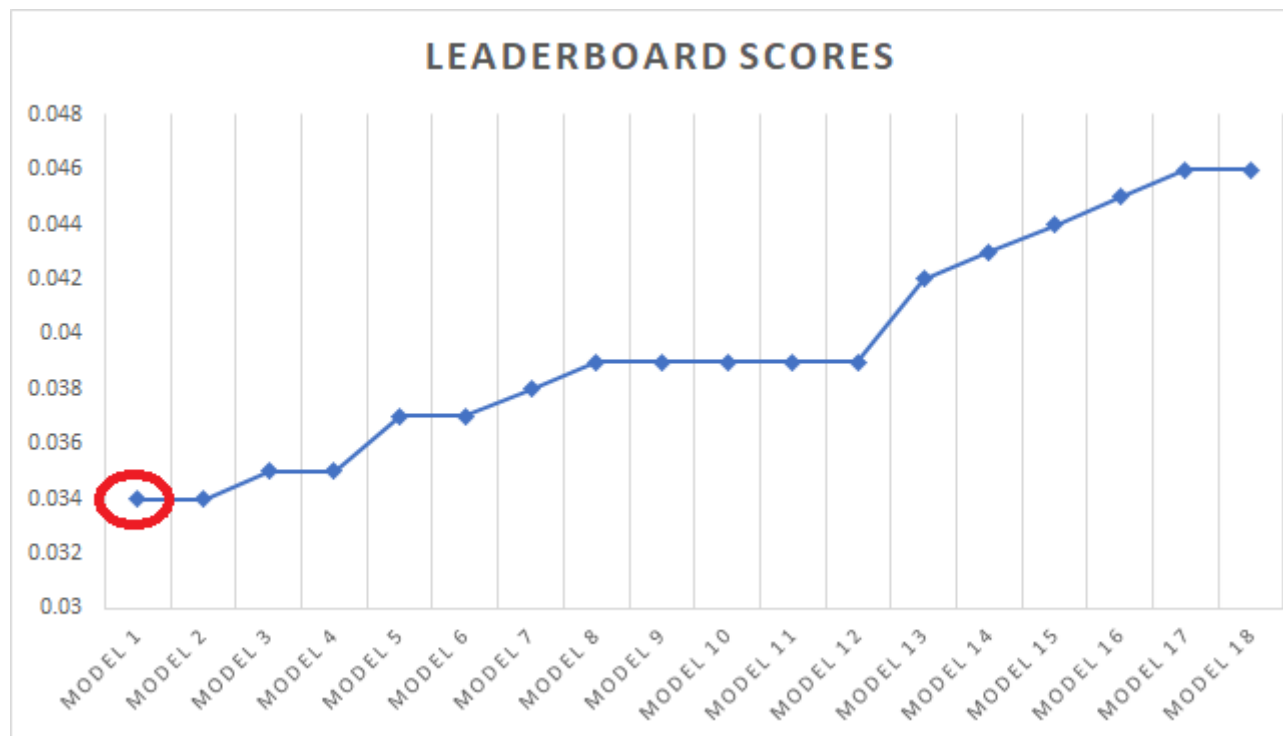
	PointNet	UNET	Multi-View 3D Object Detection
BEV LiDar Point Cloud			
RGB Image			
Front View Lidar Point Cloud			



Your most recent submission

Name	Submitted	Wait time	Execution time	Score
lyft3d_pred.csv	3 minutes ago	61 seconds	102 seconds	0.039

Complete



Level 5 - put arrows

Business context & overview

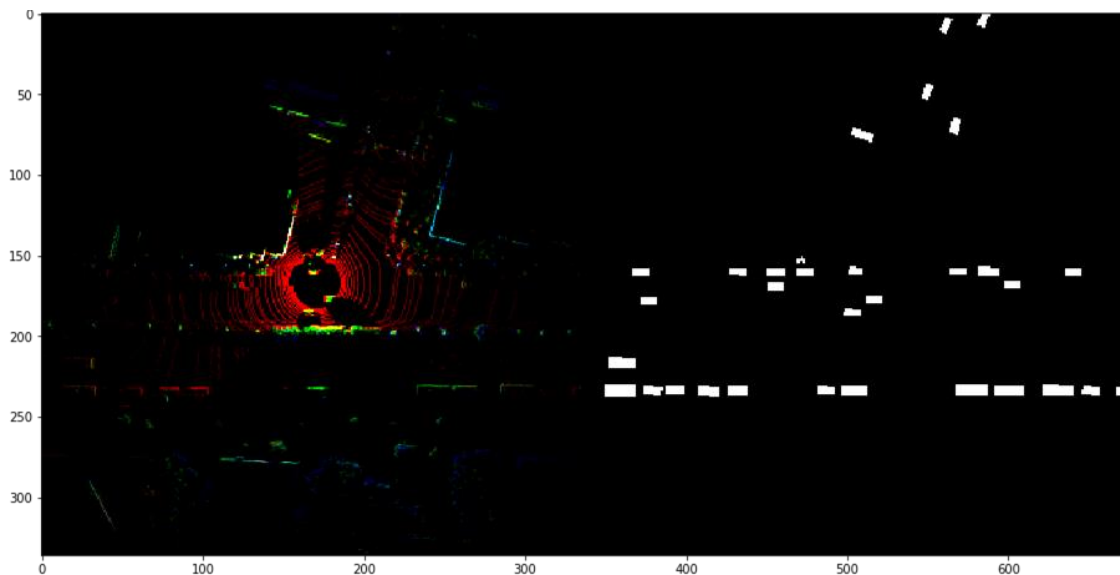
Slide 6 - Add actual table

After slide 12 - add a snapshot of the data transformation and output

Add different types of models that can be used and say that data transformation is the link (MV3D, Point Net, UNET)



Input and Target



- Input: BEV 336 x 336 x 3
(3 filters)
- Adam Optimizer
- Learning rate: 0.001
- Batch size: 8 and 16
- Weigh the loss for the cars lower to account for (some of) the big class imbalance.
- Trained for 15 epochs