3D Object Detection for Self-Driving Cars

BURAK ÇOLAKOĞLU - 201611012
FURKAN HAN KEÇELİ - 201611037

ADVISOR: FARIS SERDAR TAŞEL
CO-ADVISOR: ROYA CHOUPANI

Contents

Problem Overview

Why Perception?

Figure 1: News report about self-driving car accident.

Place: March 18, 2018
Tempe, Arizona, U.S.

Companies & Manufacturers

Now, Lyft's AV unit is acquired by Woven Planet, a subsidiary of Toyota Motor Corporation.[1]

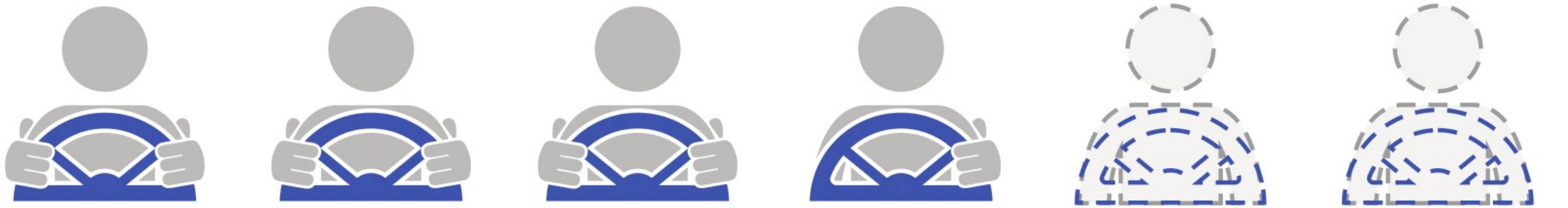
We're On a Mission of Mobility to Love, Safety to Live[2]

Formerly the Google self-driving car project—stands for a new way forward in mobility[3]

The future is for riders[4]

SAE AUTOMATION LEVELS

Full Automation



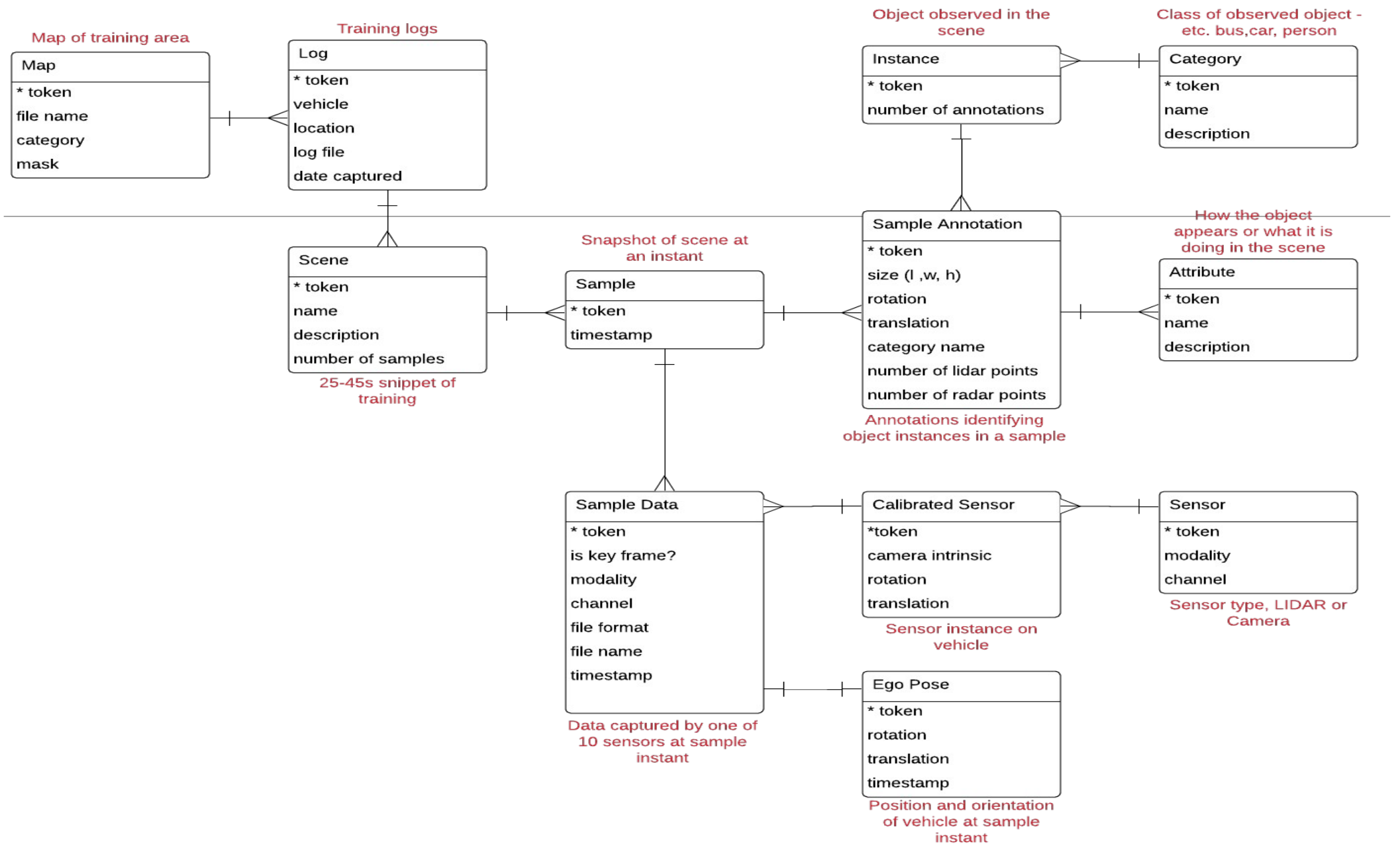
0	1	2	3	4	5
No Automation	Driver Assistance	Partial Automation	Conditional Automation	High Automation	Full Automation
Zero autonomy; the driver performs all driving tasks.	Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design.	Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.	Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.	The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.	The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.

Dataset

Ego vehicle information

Figure 2: Representing lyft's ego vehicle

- Data collected by 10 host cars
- RGB Data captured by 7 cameras
- LIDAR data captured by 3 sensors. One sensor at the top of the car and two sensors at the front.



Dataset Structure

Dataset Contents

JSON Files	Description
scene.json	25-45 seconds snippet of a car's journey.
sample.json	An annotated snapshot of a scene at a particular timestamp.
sample_data.json	Data collected from a particular sensor.
sample_annotation.json	An annotated instance of an object within our interest.
instance.json	Enumeration of all object instance we observed.
category.json	Taxonomy of object categories (e.g. vehicle, human).
attribute.json	Property of an instance that can change while the category remains the same.
visibility.json	(not used)
sensor.json	A specific sensor type.
calibrated_sensor.json	Definition of a particular sensor as calibrated on a particular vehicle.
ego_pose.json	Ego vehicle poses at a particular timestamp.
log.json	Log information from which the data was extracted.
map.json	Map data that is stored as binary semantic masks from a top-down view.

Dataset Contents(cont.)

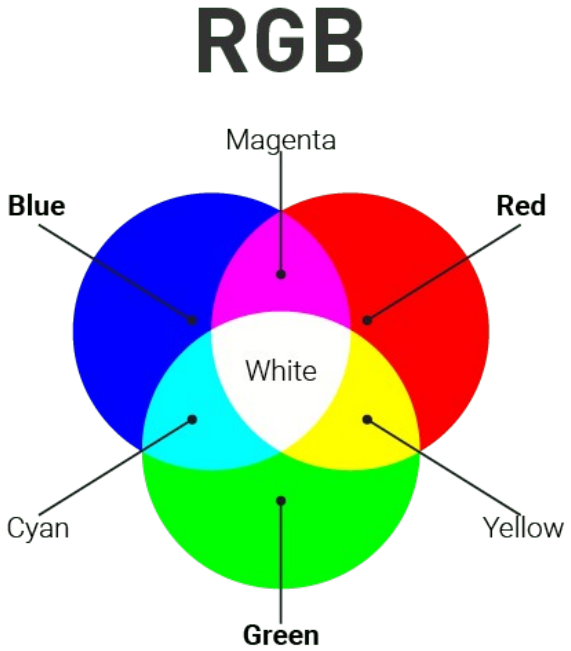


Figure 3: The RGB data(3 channels) superimose to form of the final image

Figure 4: LIDAR sensor is used to generating accurate 3D representation of surroundings by using laser beams.

192.276 - Test Data 158.757 - Train Data	30.744 - Test Data 27.468 - Train Data

Point clouds that recorded by LiDAR sensor are used to calculate; width, length, height and x, y and z coordinates of the frame.

Width is the volume in which the object lies.

Length is the volume in which the object lies.

Height is the volume in which the object lies.

Center_x and Center_y are representing x and y coordinates of the captured object's location on the xy plane.

Center_z represent xz coordinate of center of a desired object's position. It is corresponding the height of the object above xy plane.

Yaw do the direction in front of the ego car, and bounding box is indicating, while on the ground. It is the angle z coordinate.

Exploratory Data Analysis

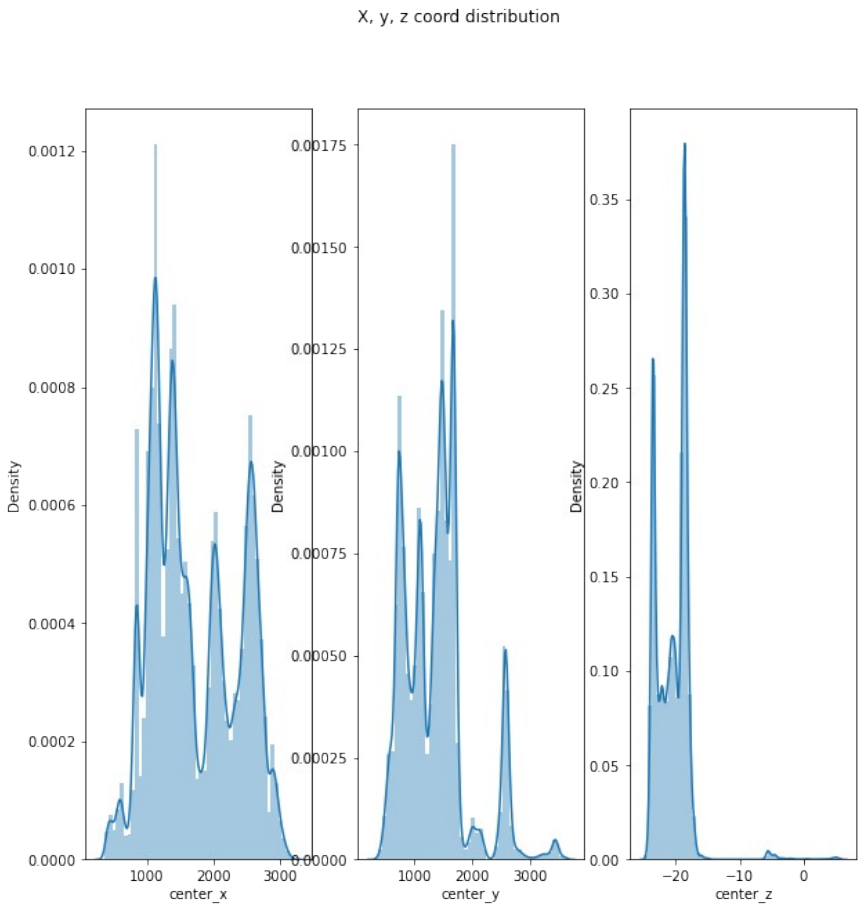


Figure 5: X, Y, Z coordinate distribution

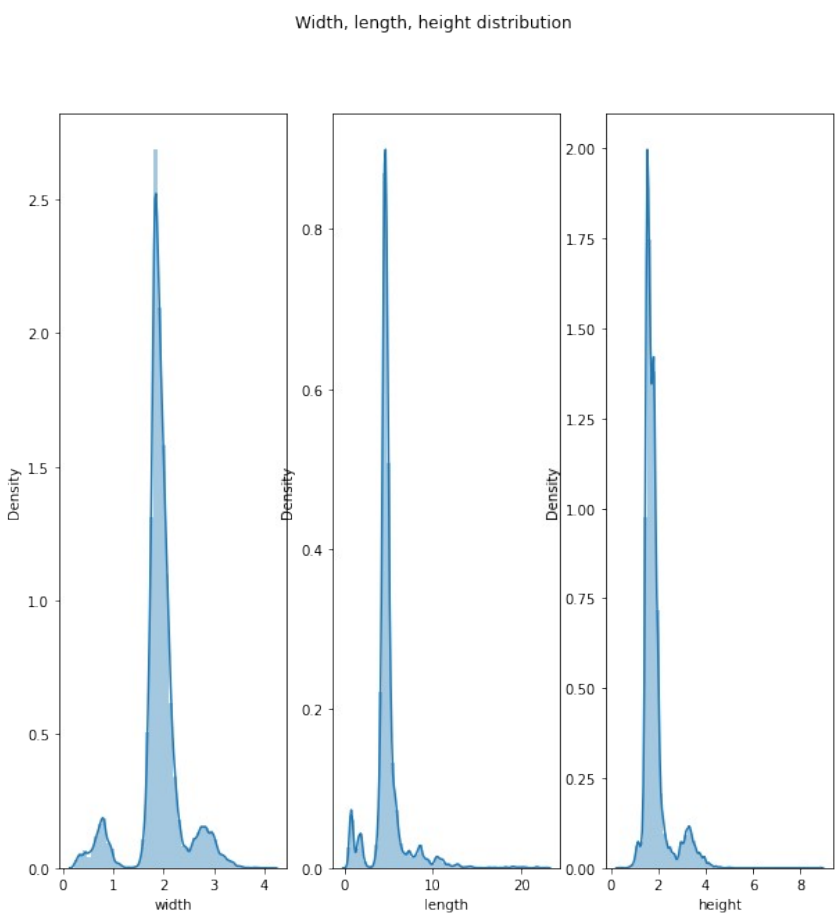


Figure 6: Width, lenght, height distribution

In the dataset, most of the objects are cars. So that, width, length and height distributions skewed to certain range. Cars objects dominating the other class with the annotated targets.

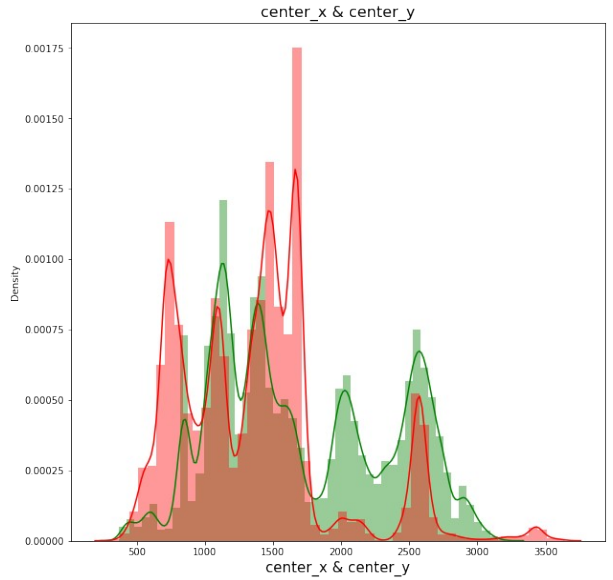


Figure 7: Ego vehicle's camera are sensing the objects left or right that because small width of the road, when compared to the length and there is a higher chance of the camera's view being blocked

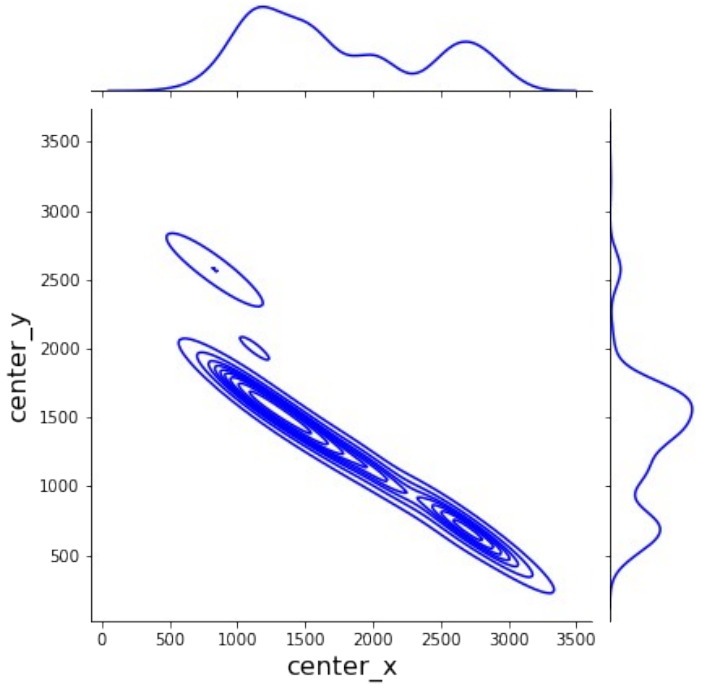


Figure 8: The reason could be the camera detect objects either too far ahead or side

center_z is right skewed and clustured near -20, and also coordinates are negative because, camera sensor placed at the top of the vehicle.

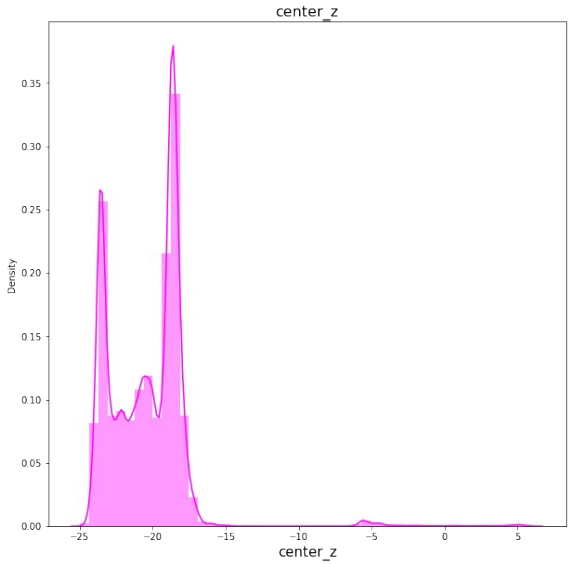


Figure 9: Most of the times, camera sensor look down to see objects due to the limitations of the camera system. So, z axis are generally negative.

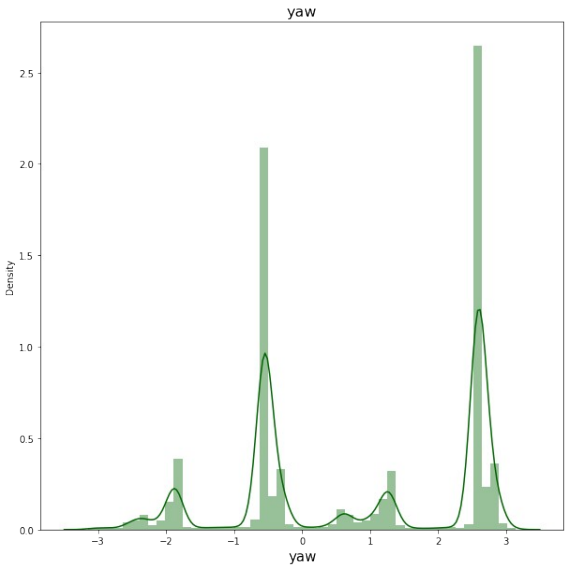
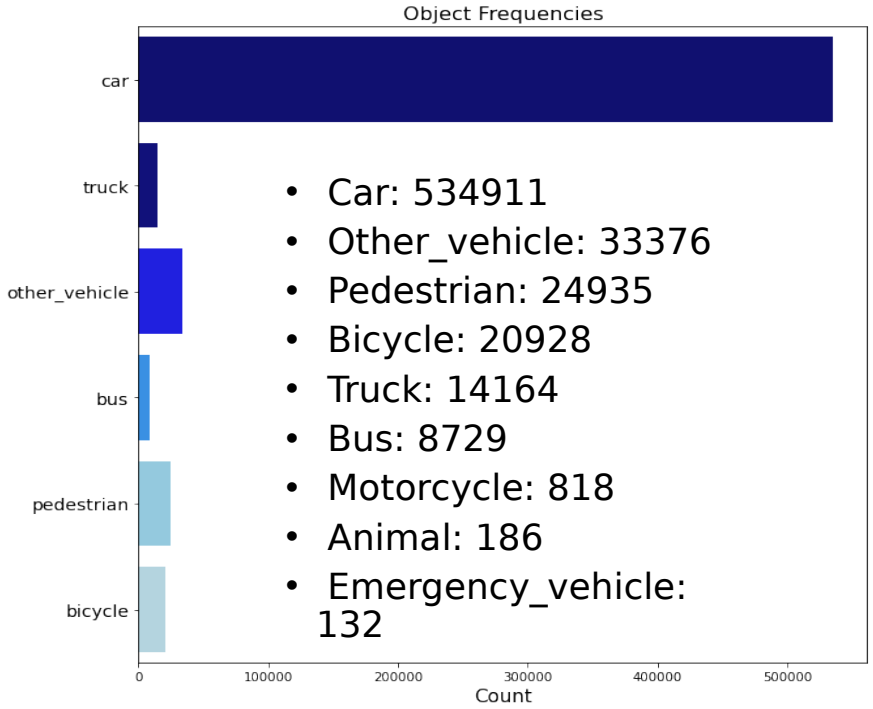


Figure 10: Yaw value is the volume of z axis. distribution of center_z has an extremely high positive. One of the peaks is around 0.5 and the other is around 2.5

Object Frequencies and Distributions



- Car: 534911
- Other_vehicle: 33376
- Pedestrian: 24935
- Bicycle: 20928
- Truck: 14164
- Bus: 8729
- Motorcycle: 818
- Animal: 186
- Emergency_vehicle: 132

Figure 11: Object Frequencies of Dataset

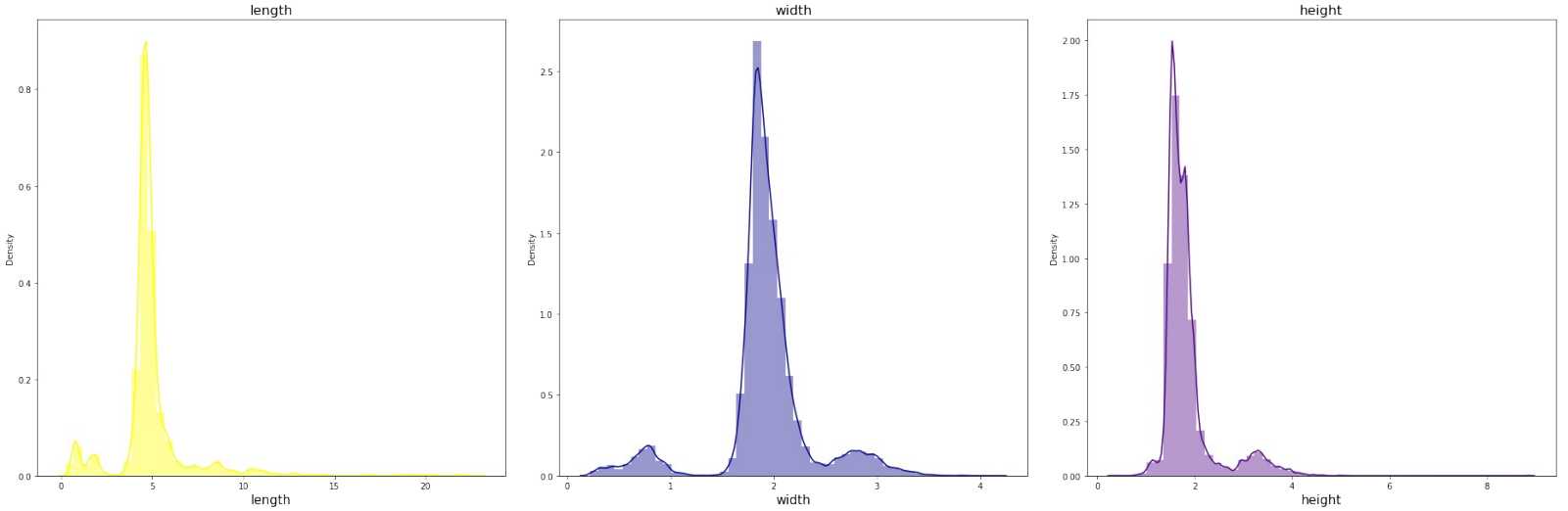
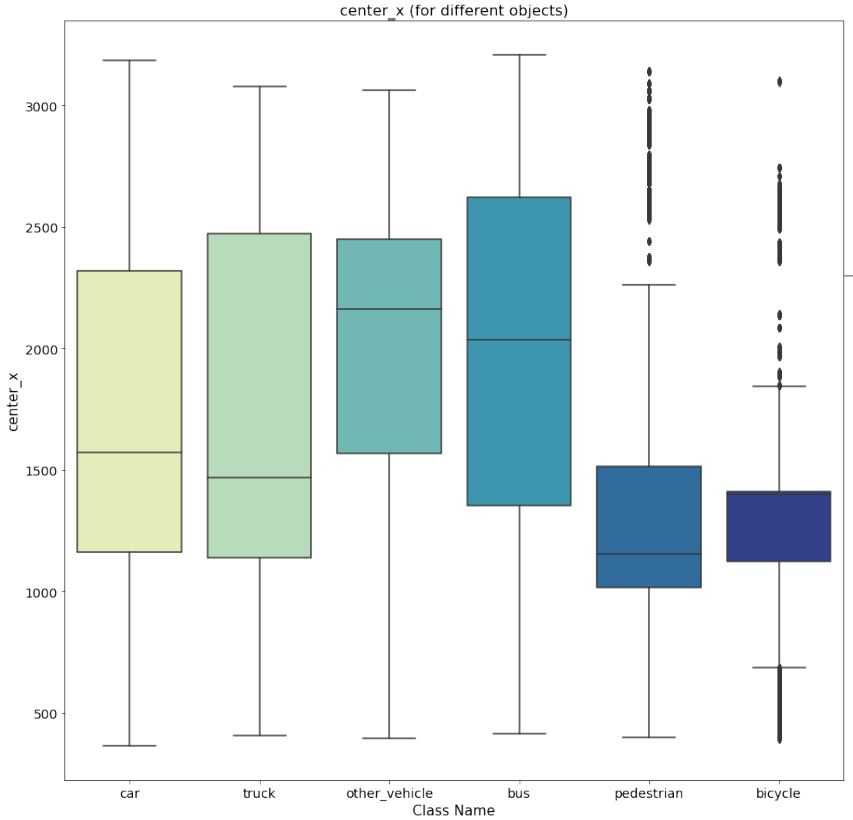
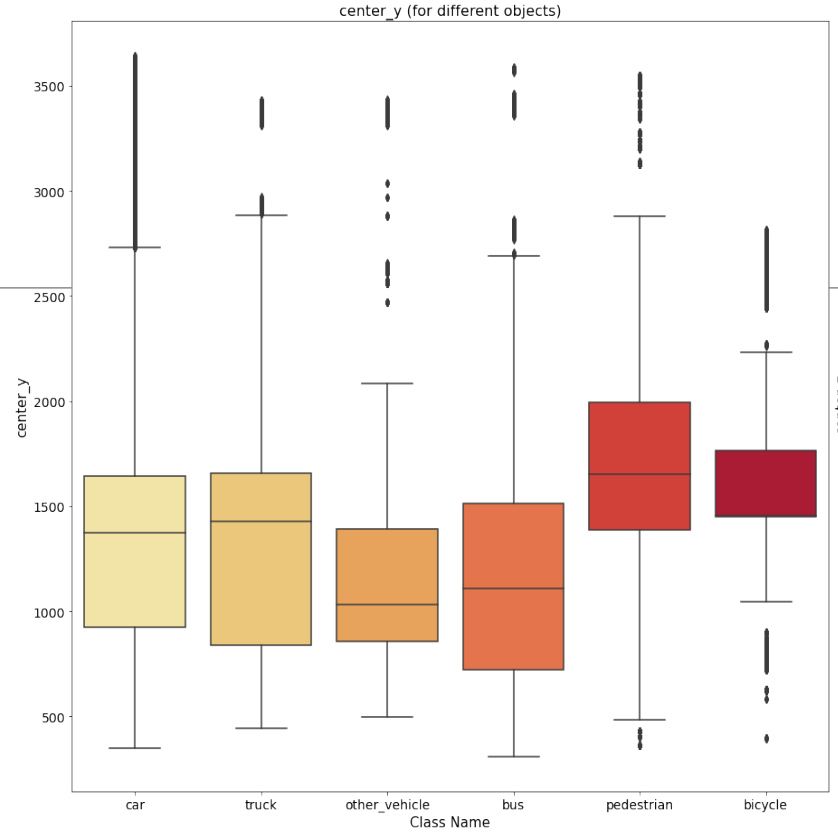


Figure 12/13/14: the distributions of length, width and height of the objects are skewed to a particular range.





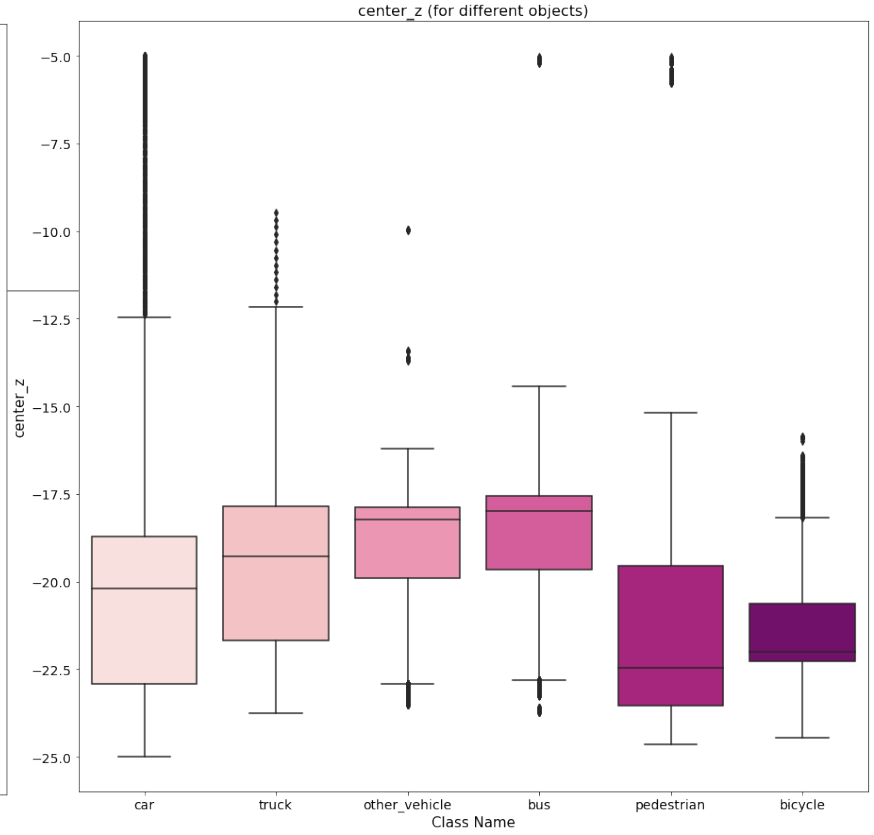


Figure 15/16/17: As we understand from the below box plots, larger objects detected more precisely because of their larger size compared to smaller objects. Also, they are far away from the host cameras.

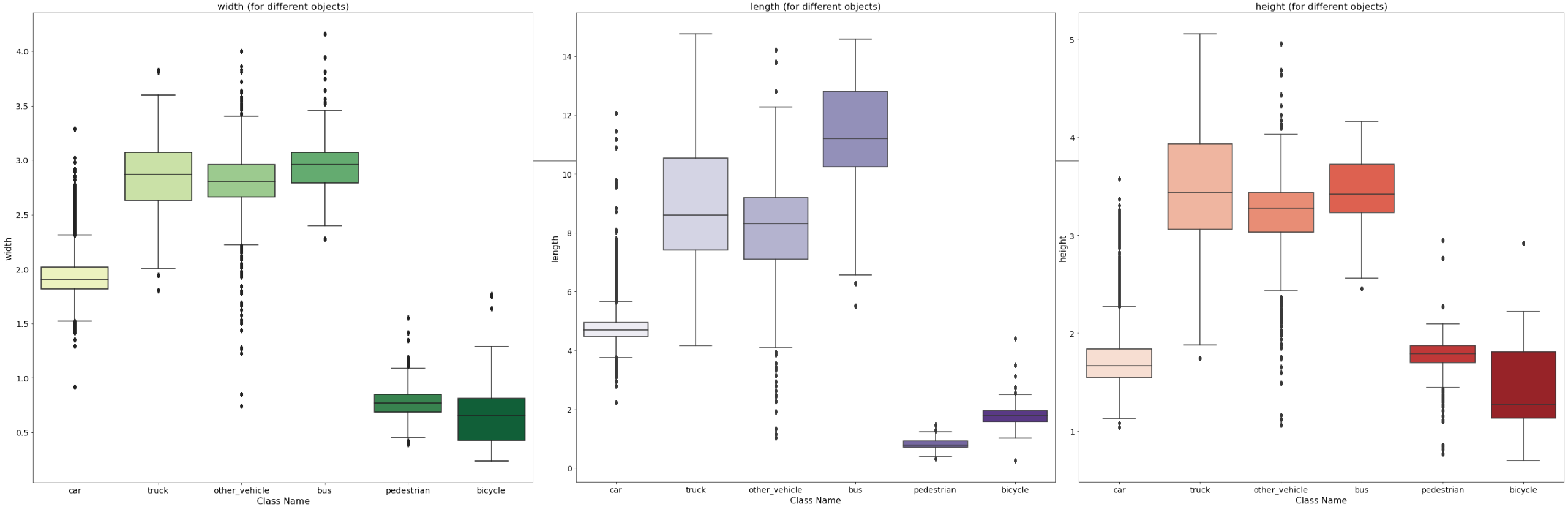


Figure 18/19/20: Width, length and height distributions for different objects are indicative of their sizes.

Data Transformation

Point clouds defines the position of the 3D objects in the ego vehicle's frame of reference

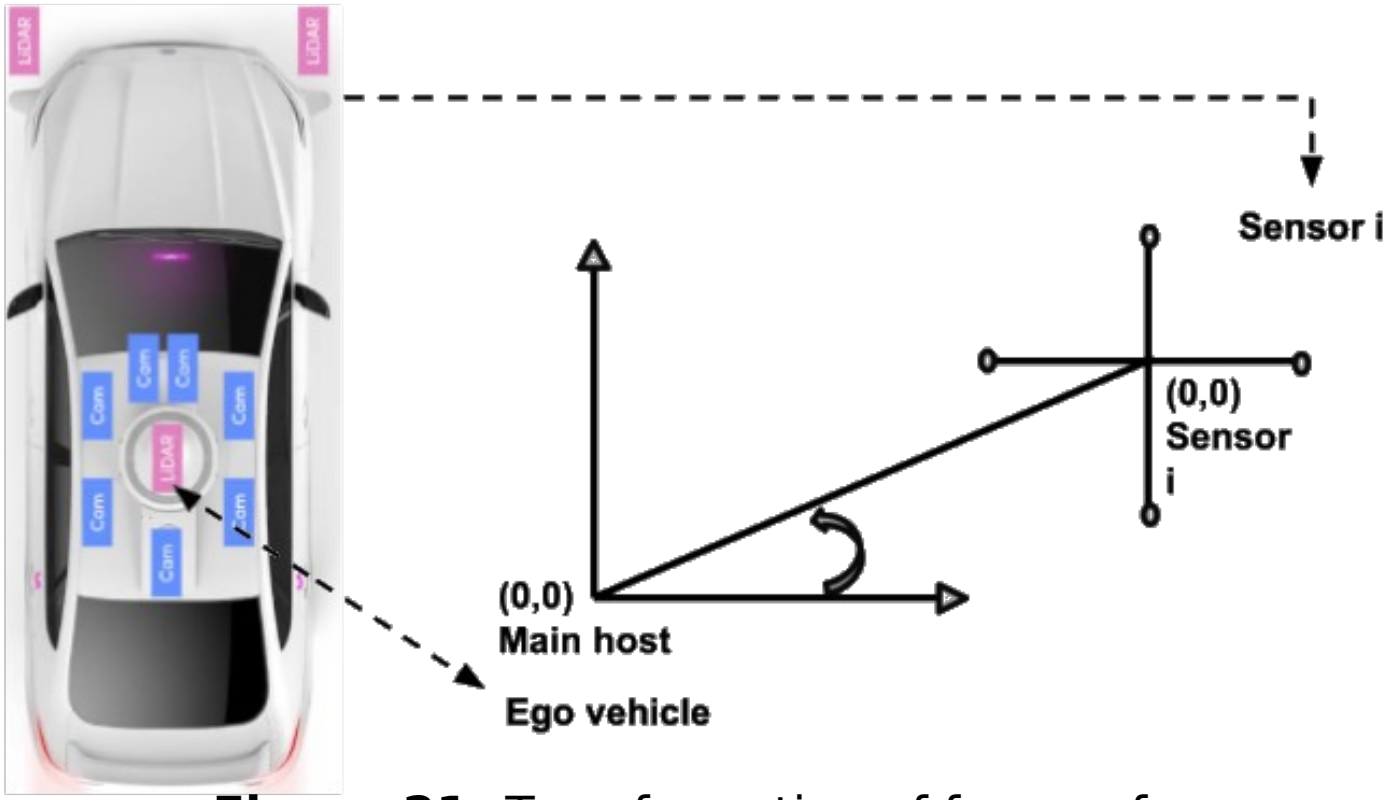


Figure 21: Transformation of frame of reference.

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figure 22: Data transformation matrix

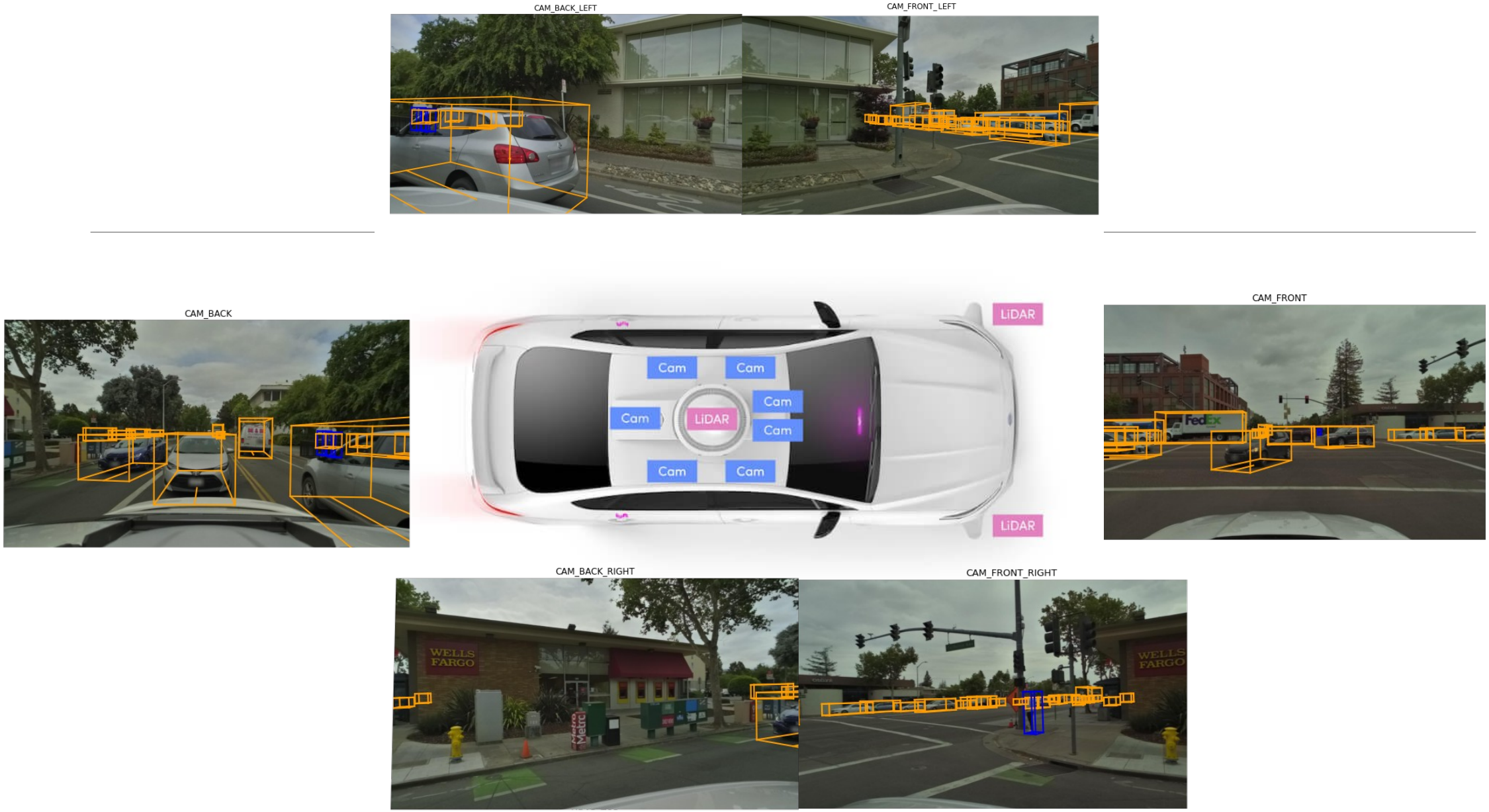


Figure 23: Representation of rendered scene.

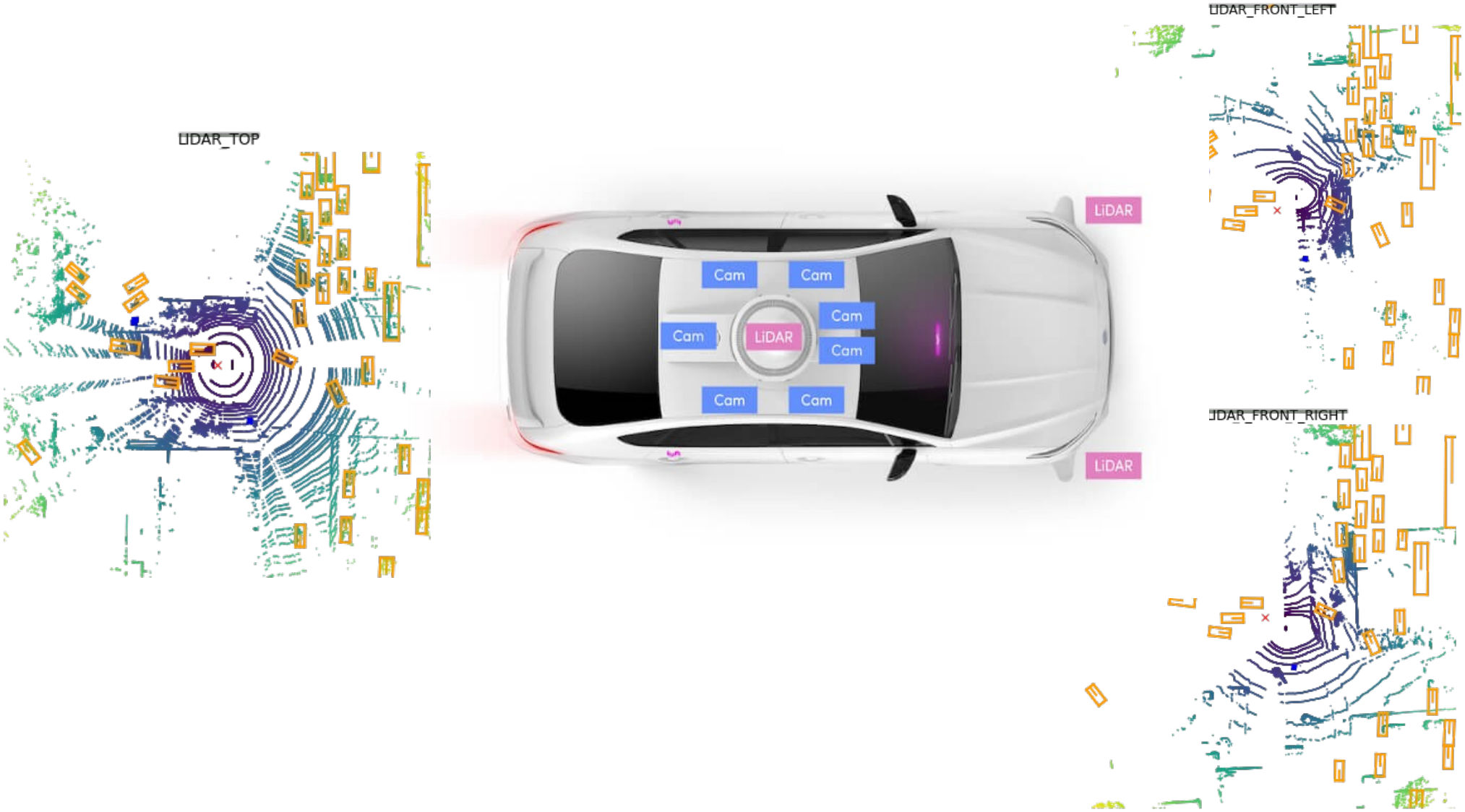


Figure 24: Representation of the point cloud transformation of the scene which done by LiDAR.

Below images example of the, how the raw image, and projected LiDAR and Camera into one image looks like.

Figure 25:
Raw

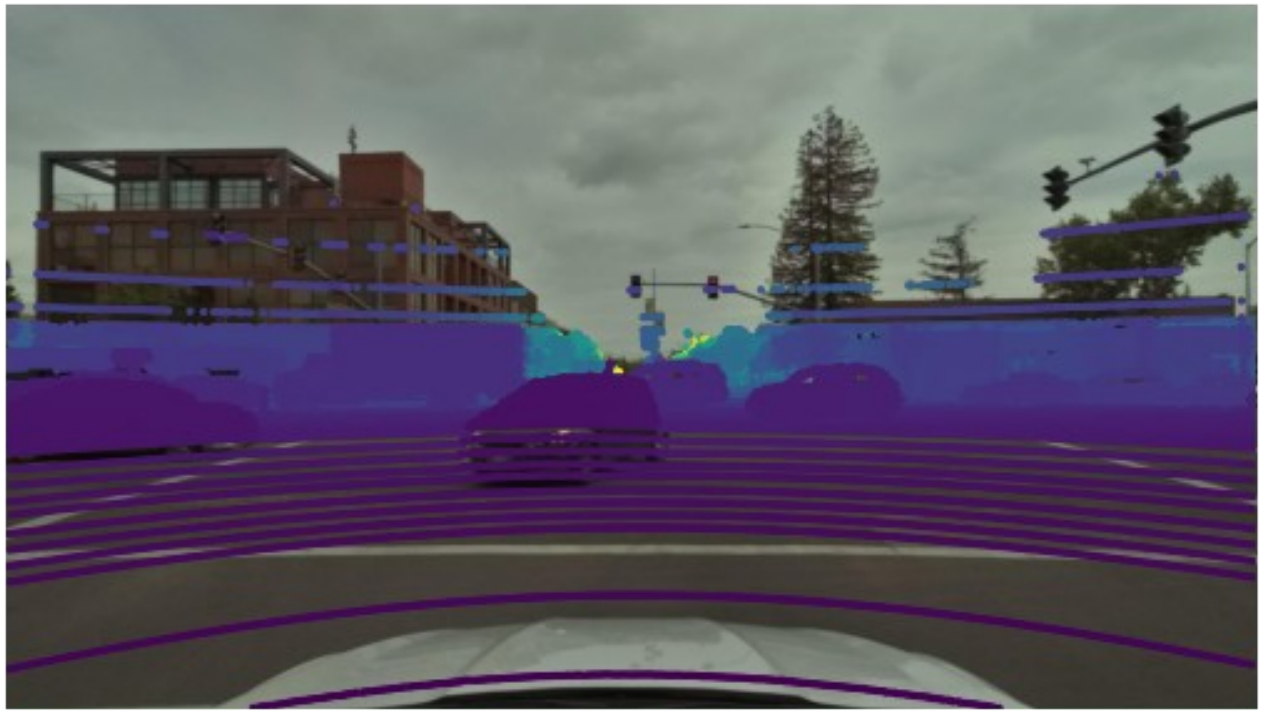


Figure 26:
Projected

Model Building

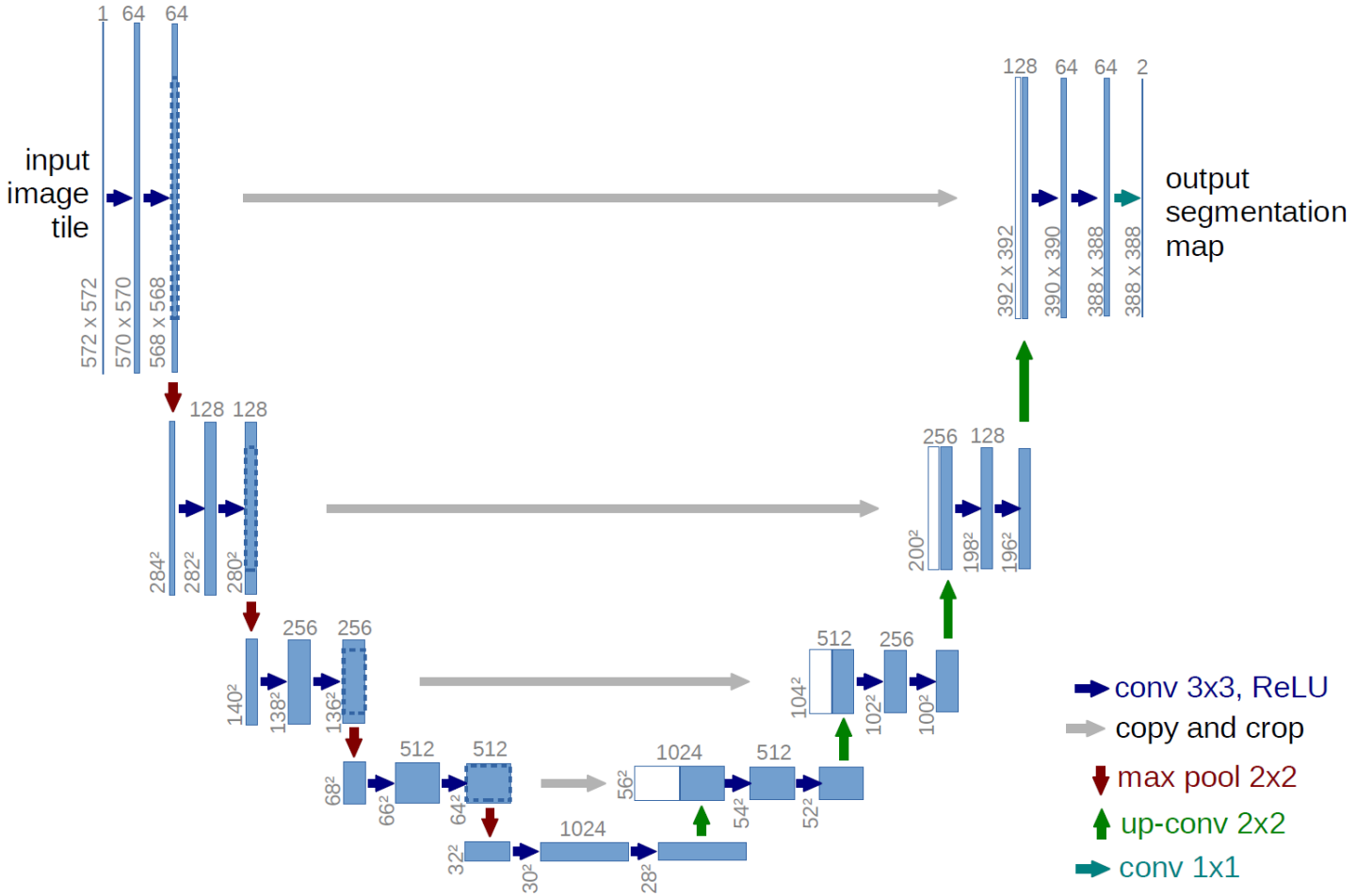


Figure 27: U-Net Architecture

1- U-Net architecture

1. The contracting/downsampling path
2. Bottleneck
3. The expanding/upsampling path

- U-Net Fully Conv Neural network
- Predicts the objects for each pixel in BEV
- Morphological transformations to fit

Boxes

- Transformations to fit boxes in the

world space

Figures below are representing U-Net input/output model.

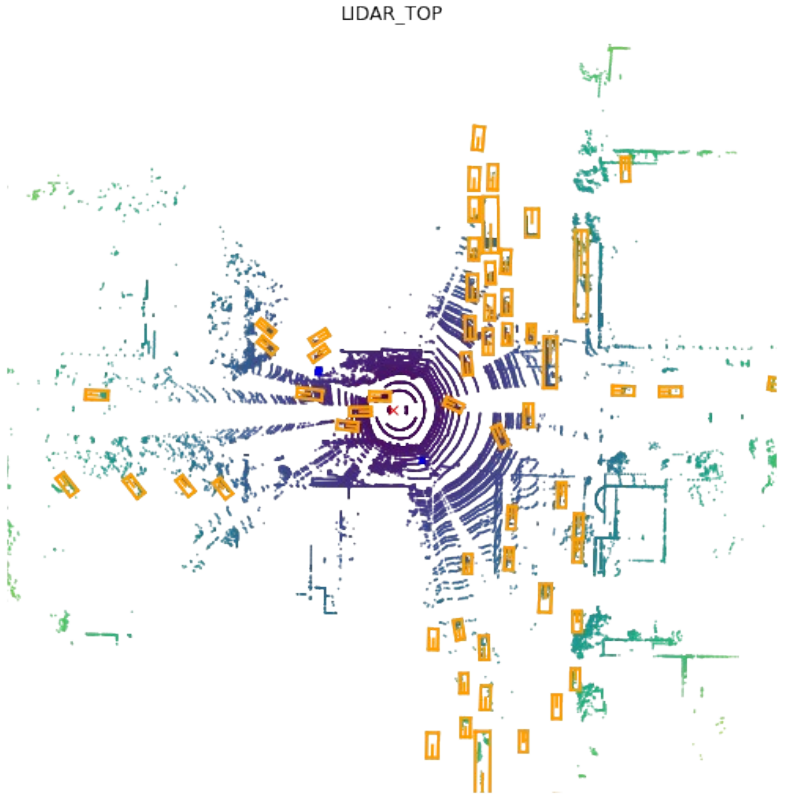


Figure 28: Bird Eye View(BEV) with bounding box.

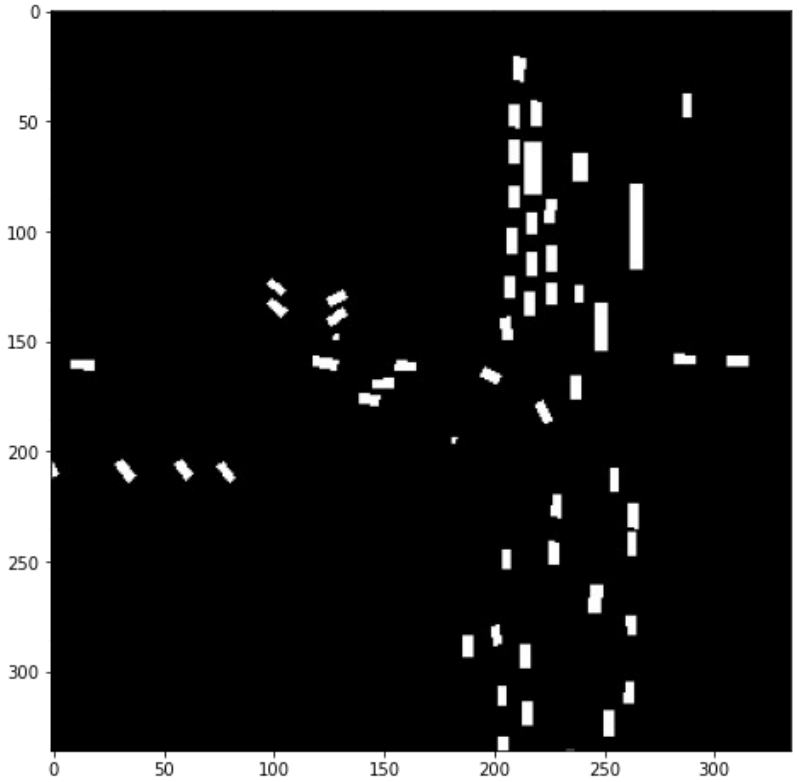
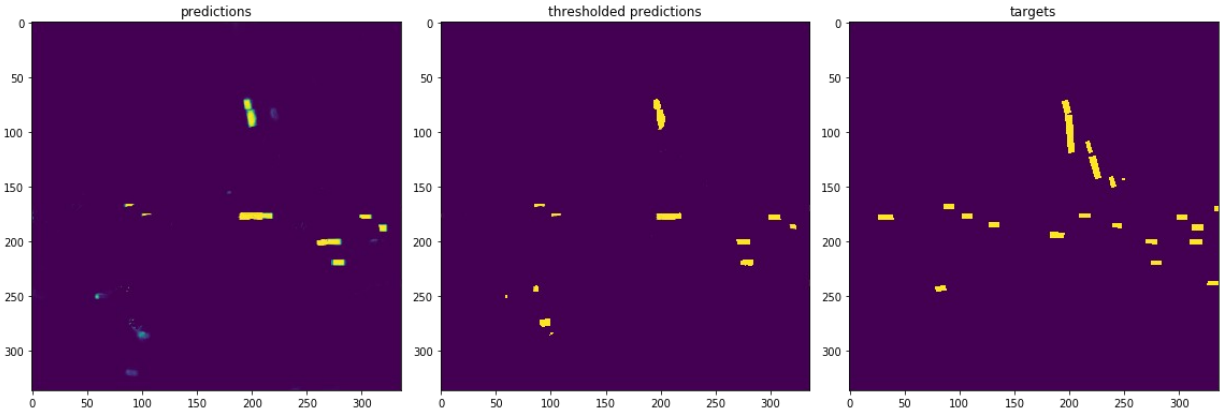


Figure 29: Ground Truth(GT) for segmentation.

- Preprocessing for each frame
- 3 channels into the network
- Input: 336 x 336 x 3 channels
- BEV of the point cloud



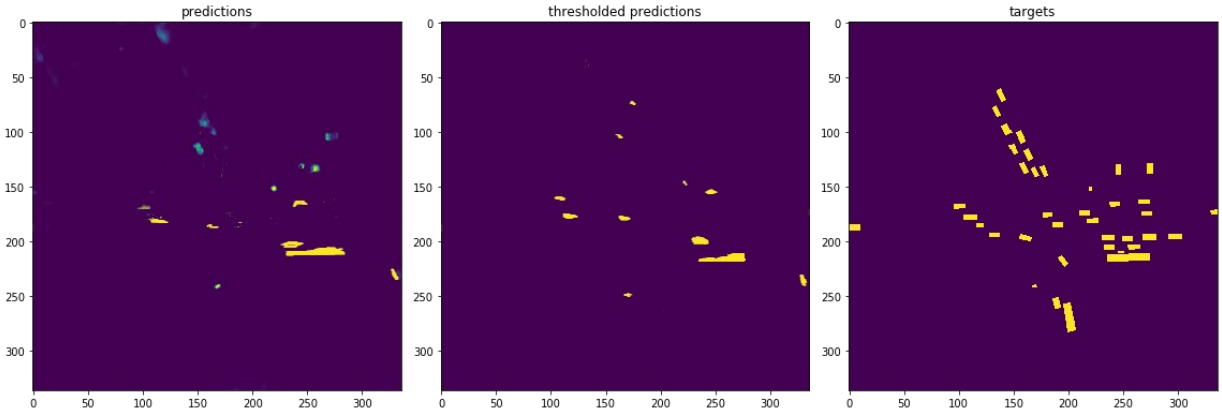


Figure 30/31: U-Net output predictions. Predictions with threshold. Ground truth.

Morphological Transformation and Thresholding

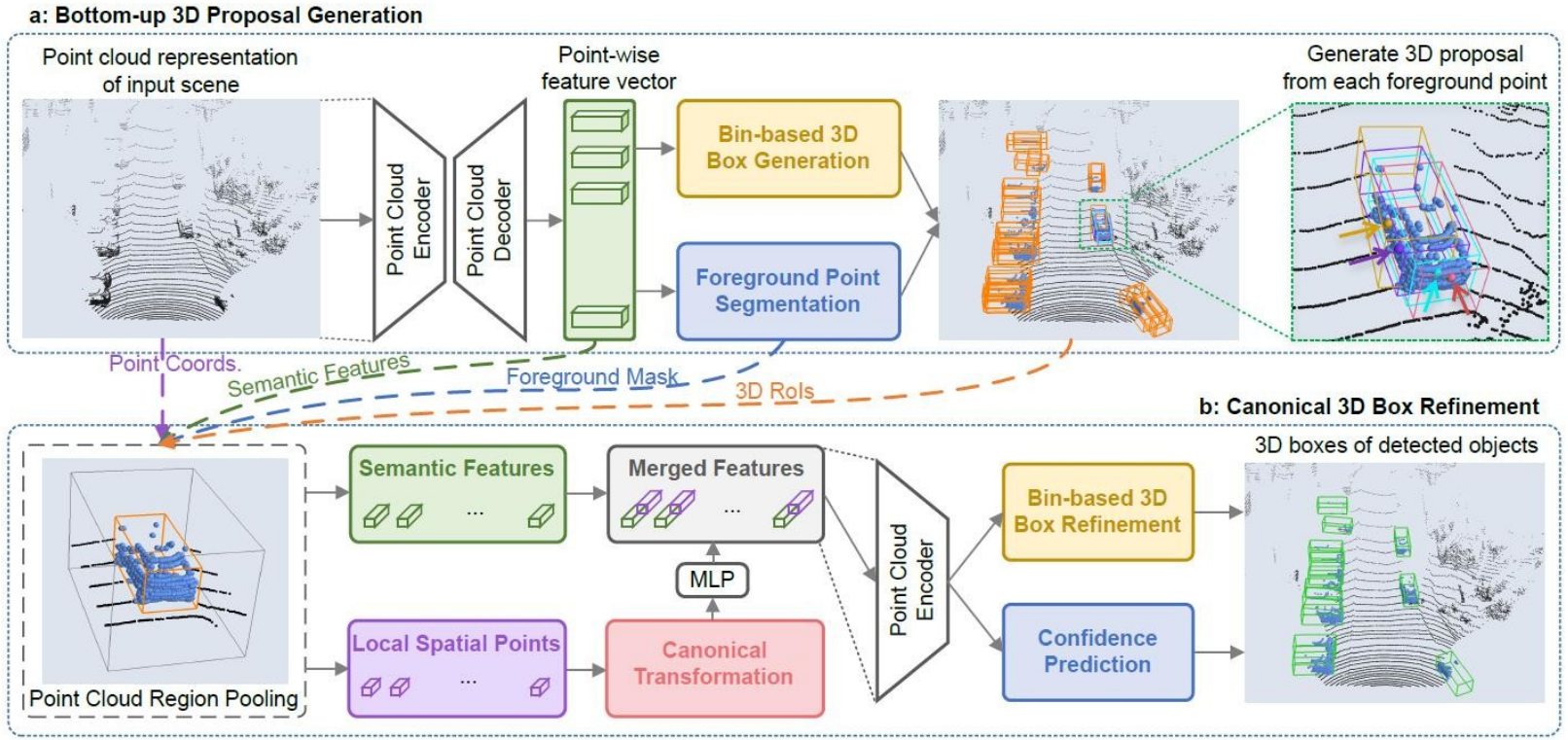


Figure 32: The PointRCNN architecture for 3D object detection from point cloud. The whole network consists of two parts: (a) for generating 3D proposals from raw point cloud in a bottom up manner. (b) for refining the 3D proposals in canonical coordinate.

2- PointRCNN architecture

- Bottom-up 3D proposal generation via point cloud segmentation
- Point cloud region pooling
- Canonical 3D bounding box refinement

Figures below are representing PointRCNN input/output model.

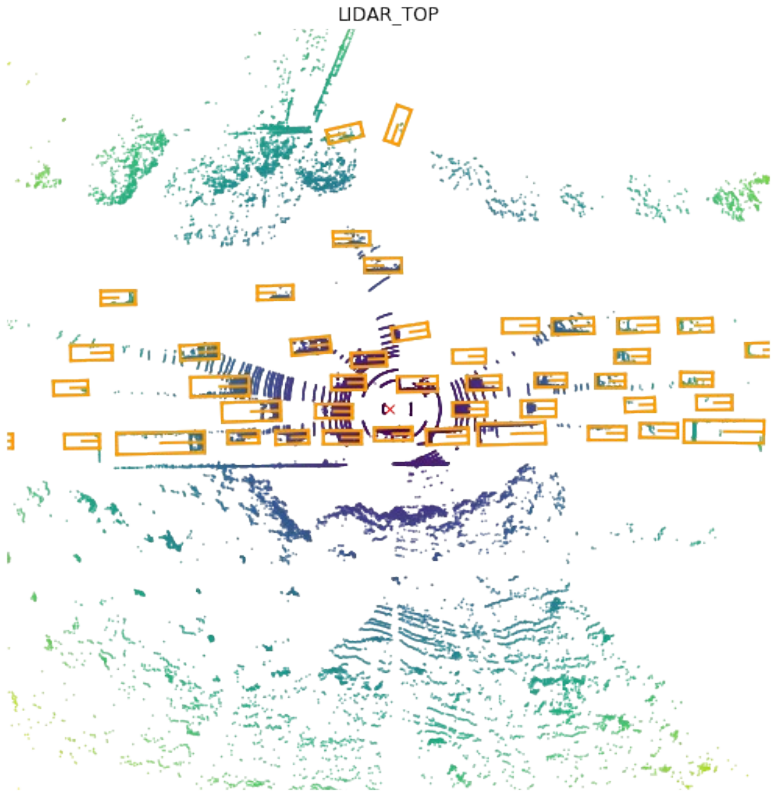


Figure 33: Bounding boxes representing ground truth annotated objects of the scene.

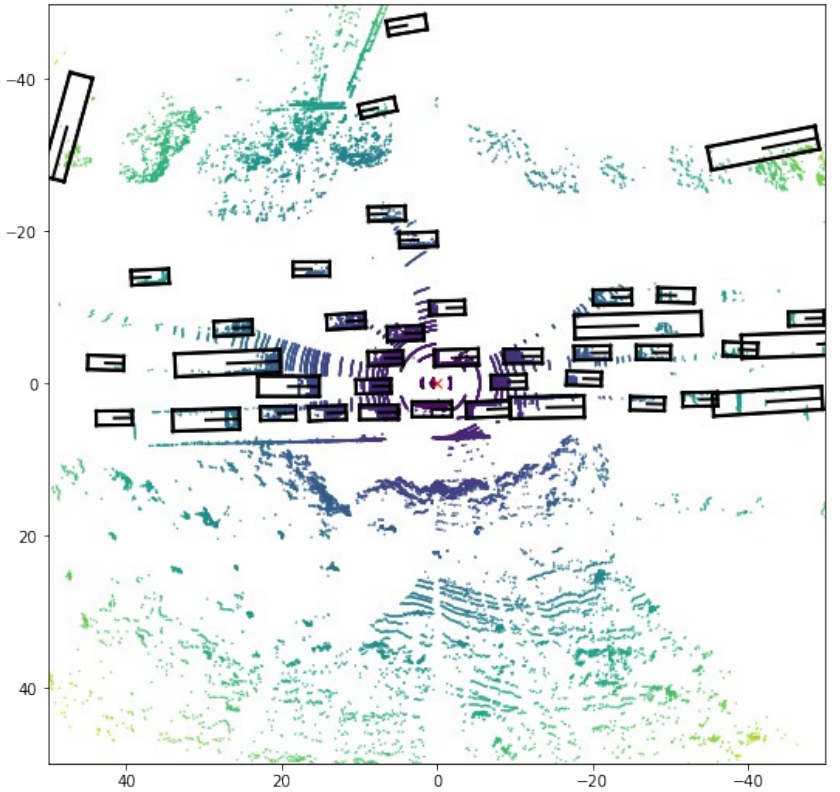


Figure 34: Bounding boxes representing predicted objects.

Evaluation and Experimental Results

Model Evaluation: Mean Average Precision (mAP) and Intersection over Union(IoU)

Different thresholds:(0.5->(0.95))
Learning rate = 0.05

$$accuracy = \frac{1}{|\delta|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

mAP across different classes

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

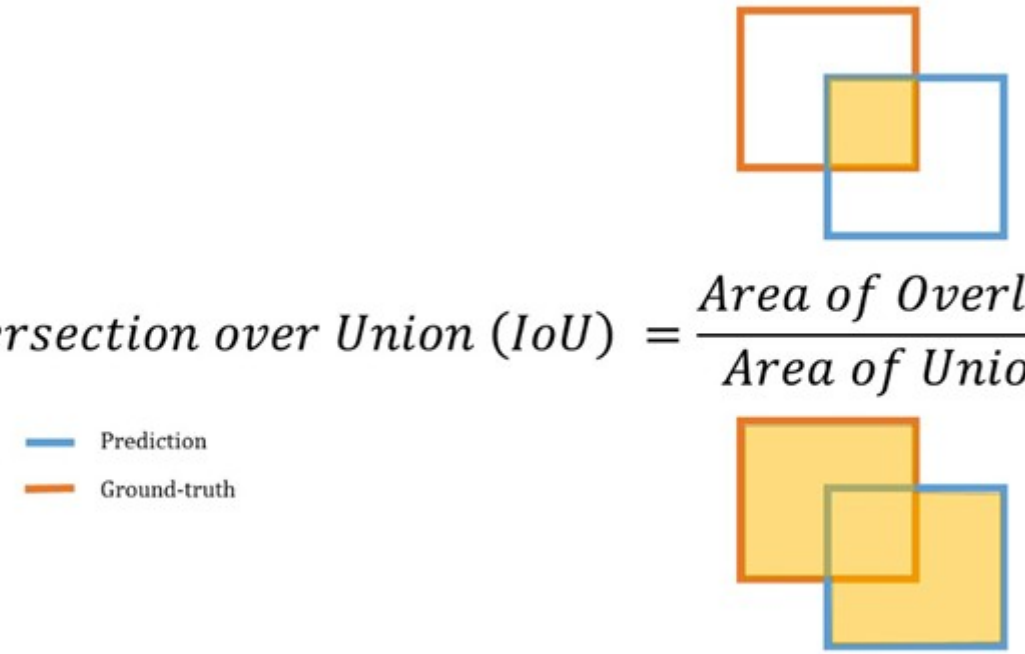


Figure 35: Approach to calculate Intersection over union

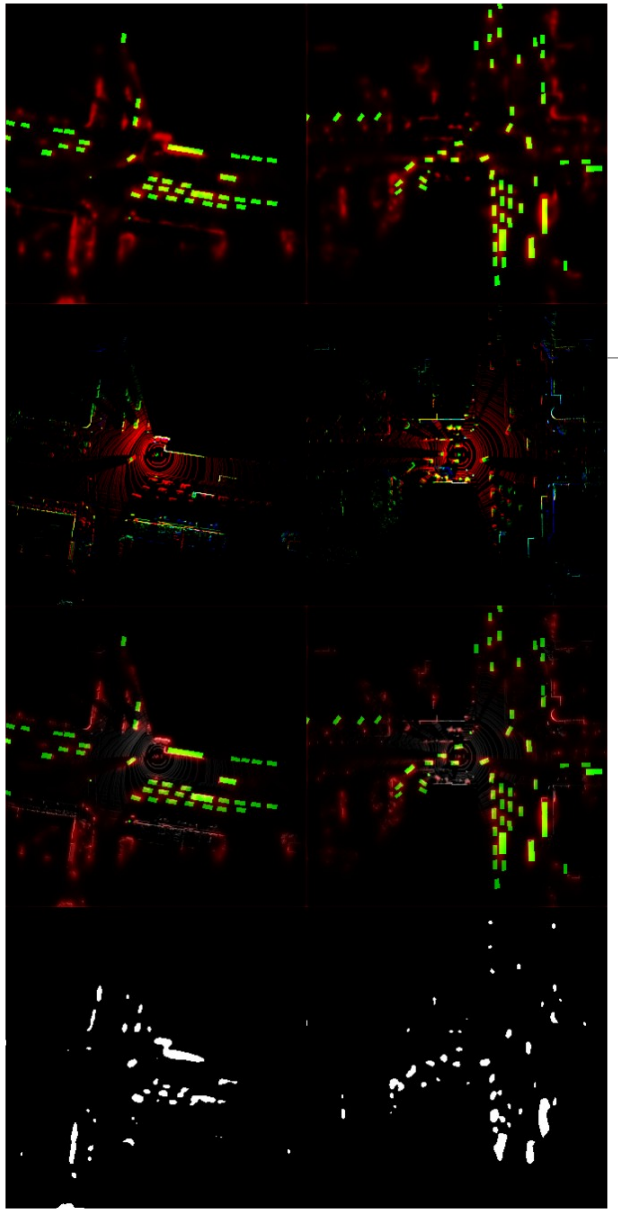


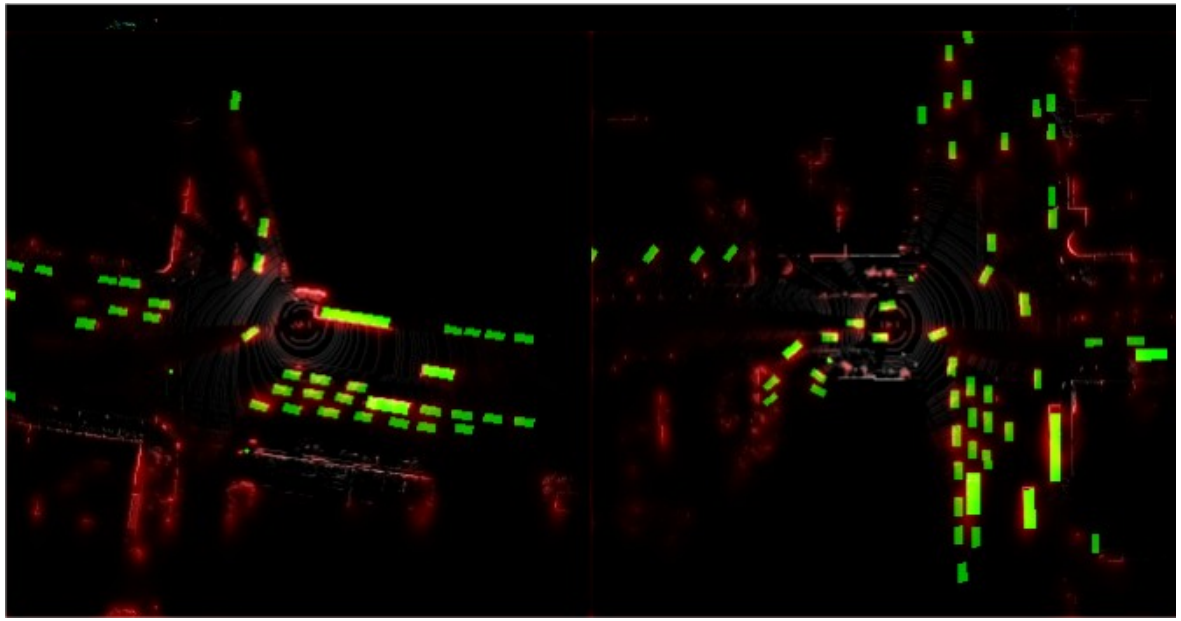
Figure 36: 15th epoch's output

1- U-Net Evaluation

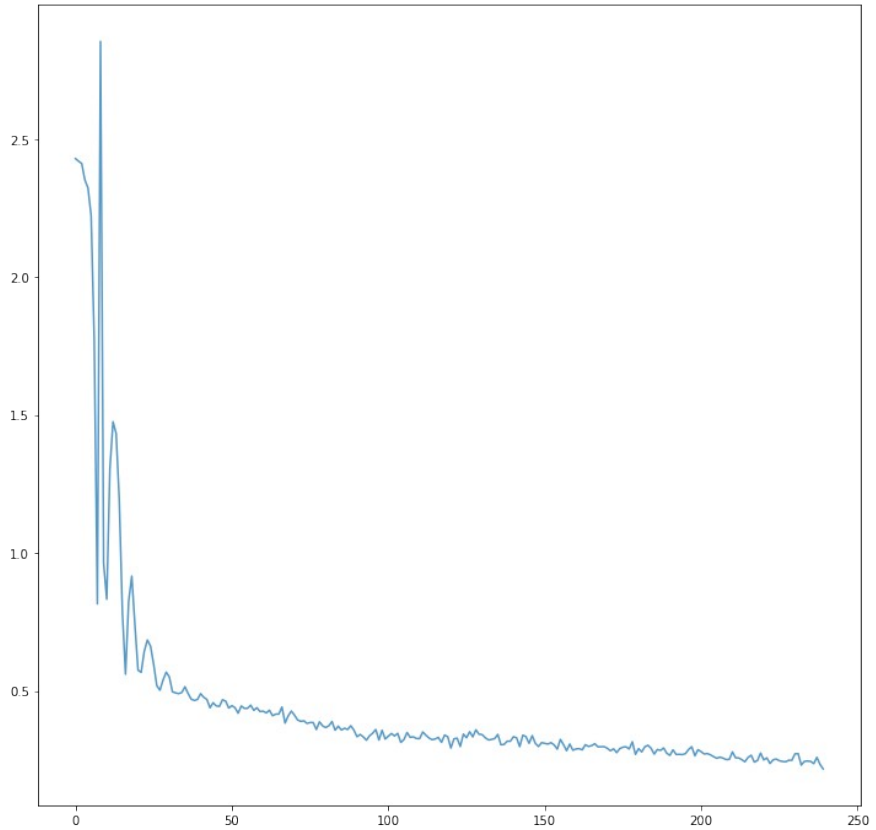
IoU threshold = 0.5
Class_names = ['animal', 'bicycle', 'bus', 'car', 'motorcycle', 'other_vehicle', 'pedestrian', 'truck']
Average per class mean average precision(mAP) = 0.003045863536953 2
('animal', 0.0)
('bicycle', 0.0)
('bus', 0.002812631567788 14)
('car', 0.09122945038512 7)
('motorcycle', 0.00232171318517606)
('other_vehicle', 0.00442513591886761)
('pedestrian', 0.00032145378282948 2)
('truck', 0.00345225256732867)
Mean loss: 0.38486697

- **Input:** BEV 336 x 336 x 3 (3 filters)
- **Activation function:** Adam Optimizer
- **Learning rate:** 0.001
- **Batch size:** 8 and 16
- **Number of epochs:** 15 epochs

1- U-Net Evaluation(cont.)



Black: True Negative
Green: False Negative
Yellow: True Positive
Red: False Positive



Loss: 0.31997657

2- PointRCNN Evaluation

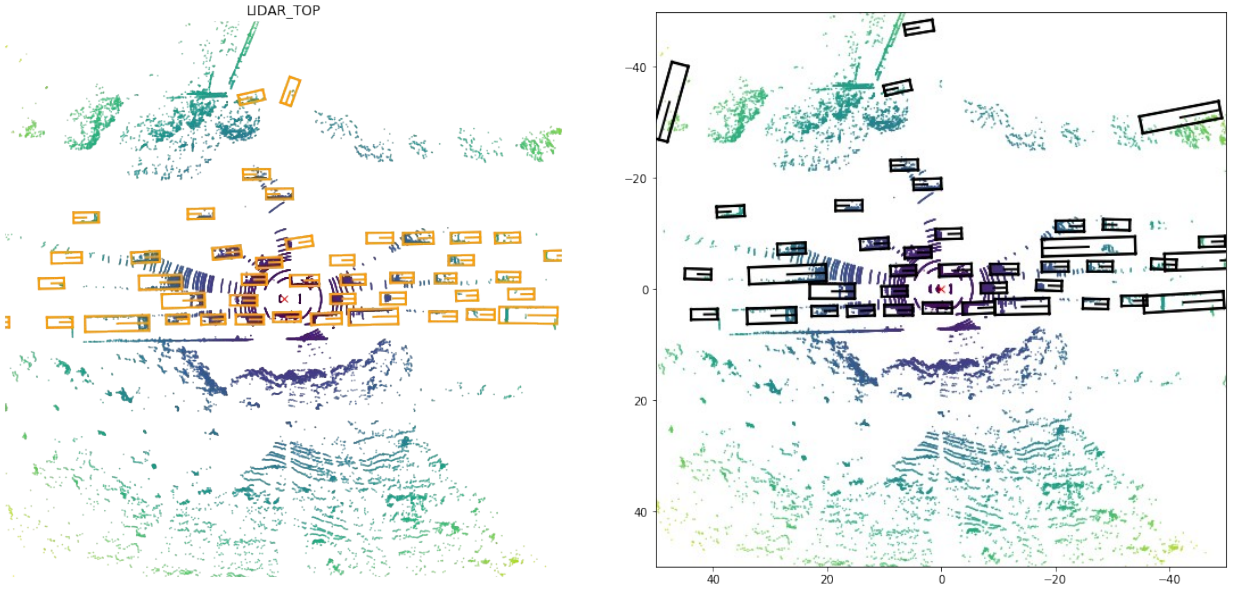


Figure 37/38: GT and Predicted representations

IoU threshold = 0.5

Class_names = ['animal', 'bicycle', 'bus', 'car', 'motorcycle', 'other_vehicle', 'pedestrian', 'truck']

Average per class mean average precision (mAP) = 0.0467794515086847 5

('animal', 0.056354749281918 2)

('bicycle', 0.17918228143918 2)

('bus', 0.44281841841201874)

('car', 0.7129183918219 5813)

('emergency_vehicle', 0.51918274281741 3812)

('motorcycle', 0.5628174291822 18312)

('other_vehicle', 0.612818248174281 73)

('pedestrian', 0.09182713817217486)

('truck', 0.5418373172281 7461)

2- PointRCNN Evaluation(cont.)

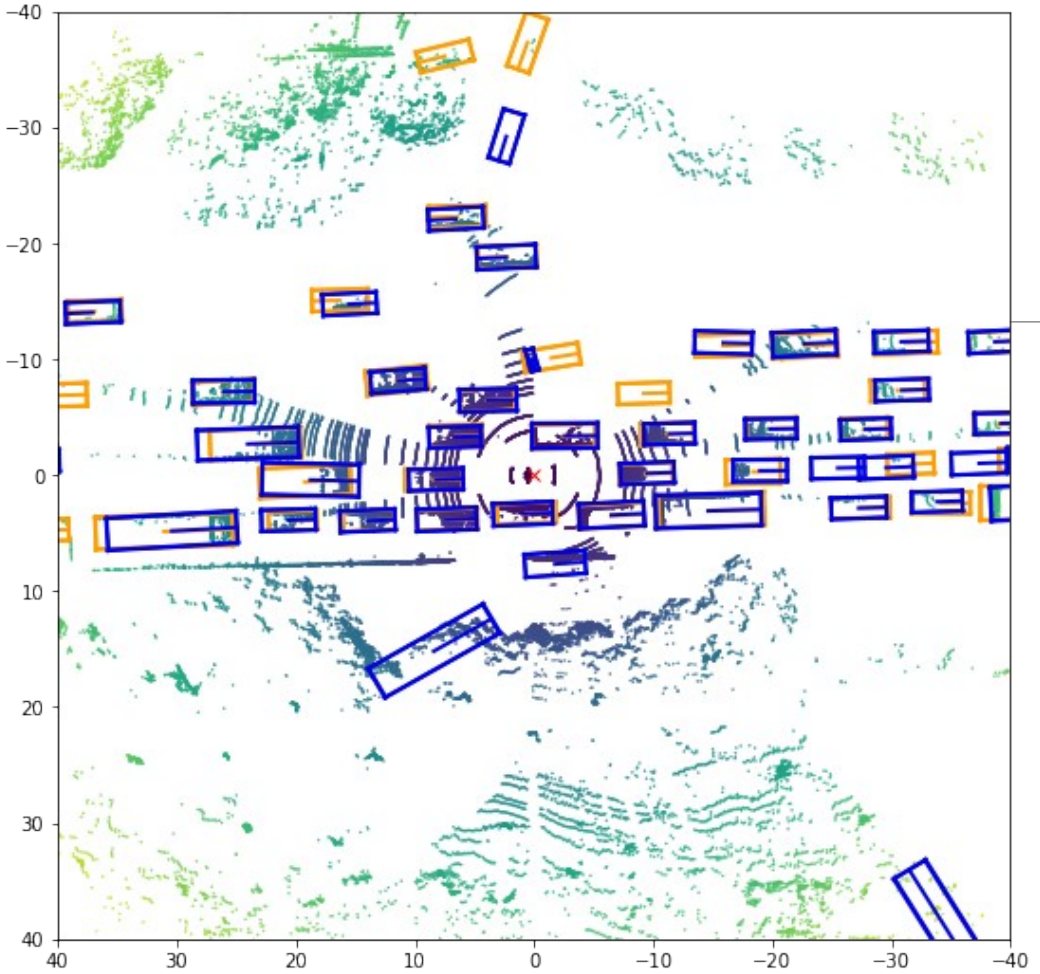


Figure 39: Oranges are ground truth(GT) objects, blues are predicted objects.

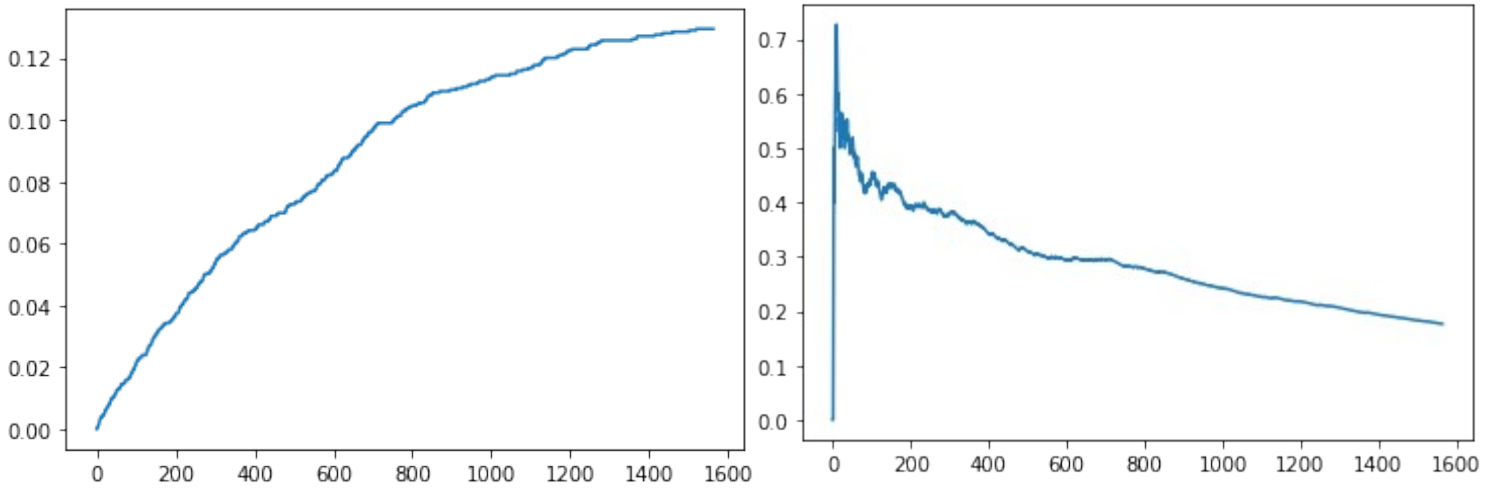


Figure 38/39: Recall and Precision graphics

Challenges and Learnings

Challenges and Learnings

- Lack of the hardware equipment of the laboratory(GPUs, VRAM, etc.)
- Couldn't increase batch size beyond 32 because of the above issues.
- Couldn't worked with all of the data.
- 15 Epochs took around 3 hours to train batch size 8, just with 882 image data, and 379 LiDAR data.
- U-Net model fails to predict, because of the working with limited data and lack of the hardware.
- Couldn't perform cross validation due to memory constraints.
- With the better hardware, the model could be tuned efficiently.
- Could be try different architectures.

Conclusion

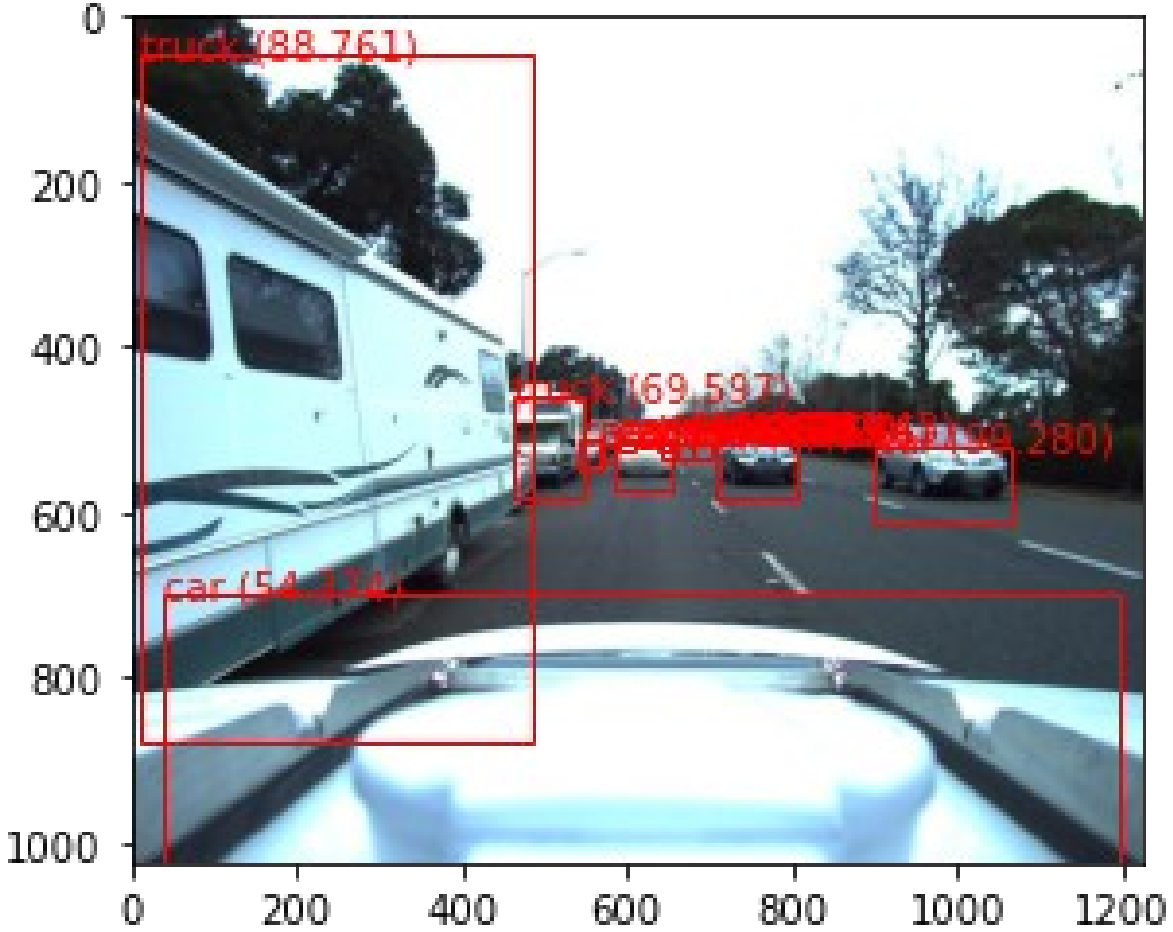
- This research project aimed to provide optimize algorithms by using real-life data which has been collected by ego vehicles.
- With this project, we planned to build 3D CNN model with latest research papers.
- In the future, we hope to further democratize would be access to self-driving technology for hundreds of millions of passengers.

Acknowledgement

Special thanks to our Advisors **Faris Serdar Taşel** and **Roya Choupani** for their insightful comments and suggestions.

We are also grateful to **Efe Çiftçi** for his great supports.

Appendix



truck 88.76119256019592
car 54.373836517333984
truck 69.59725022315979
car 99.28027391433716
car 47.321513295173645
car 41.67772829532623
car 48.240575194358826
car 91.34989380836487
car 59.06717777252197
car 47.62899875640869
car 80.41974306106567
car 86.63581609725952
car 95.26423811912537
car 98.44319820404053

References

[1] Level 5 - Lyft - self-driving Lyft

https://self-driving.lyft.com/level5/

[2] Woven Planet Holdings

https://www.woven-planet.global/

[3] Waymo

https://waymo.com/

[4] Zoox: The future is for riders

https://zoox.com/

[5] U-Net: Convolutional Networks for Biomedical Image Segmentation

https://arxiv.org/abs/1505.04597

[6] PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud

https://arxiv.org/abs/1812.04244

Thanks for Listening.