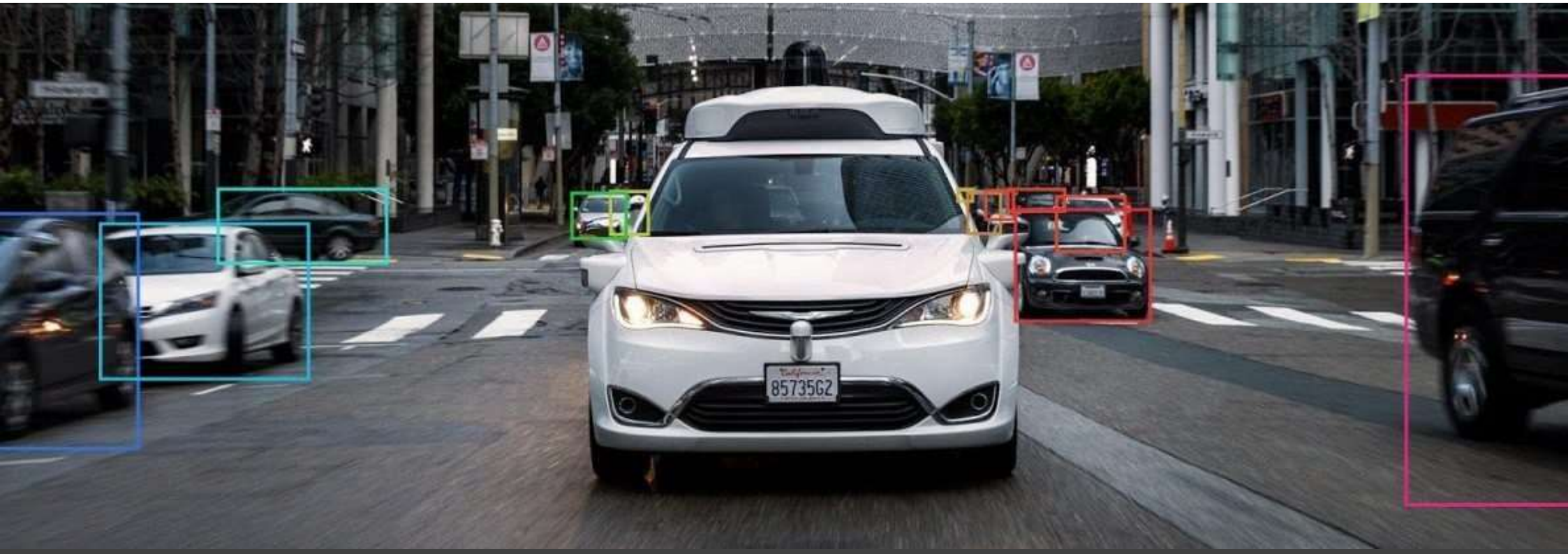


2D & 3D Object Detection on Waymo Dataset

Introduction

- 2D Object Detection:
YOLOv4

- 3D Object Detection:
LDLS

YOLOv4 Content

- Why YOLOv4 ?
- YOLOv4 Network Architecture
  - Backbone
  - Neck
- Why More Accurate ?
  - Example of Bag of freebies
  - Example of Bag of specials
- Experimental Result
  - Why overfitting ?

Why YOLOv4 ?

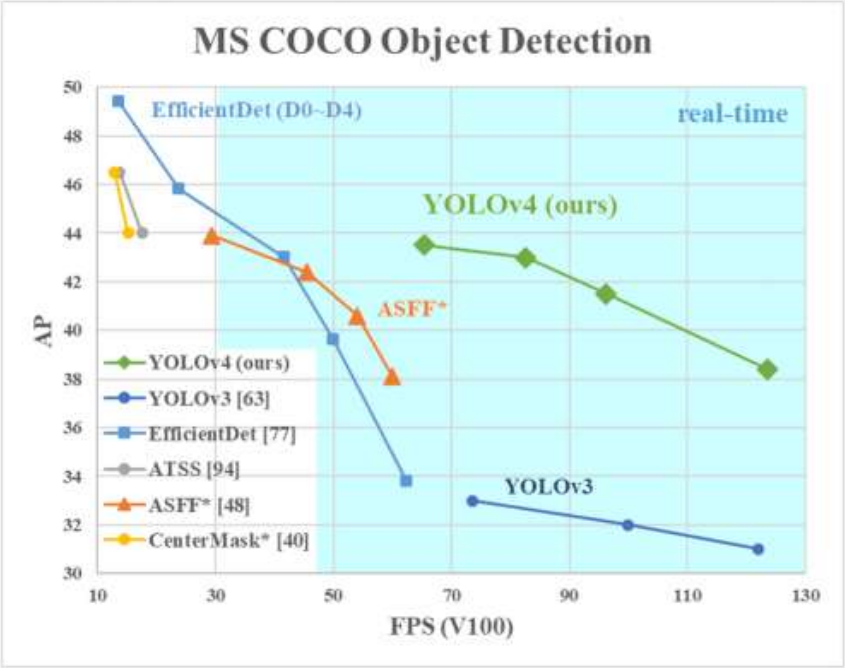
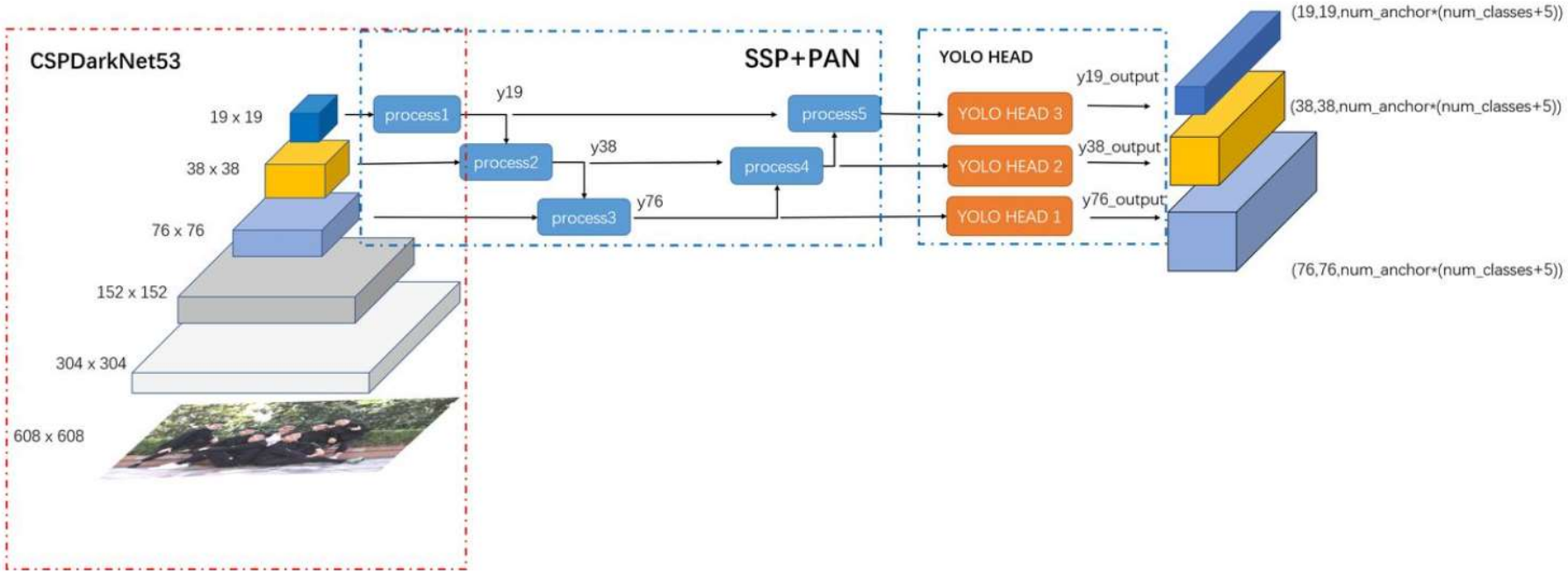


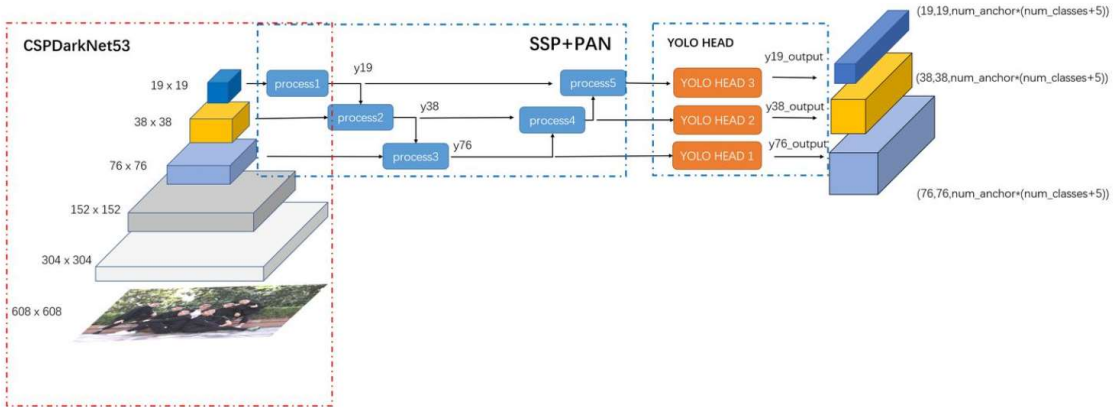
Figure 1: Comparison of the proposed YOLOv4 and other state-of-the-art object detectors. YOLOv4 runs twice faster than EfficientDet with comparable performance. Improves YOLOv3's AP and FPS by 10% and 12%, respectively.

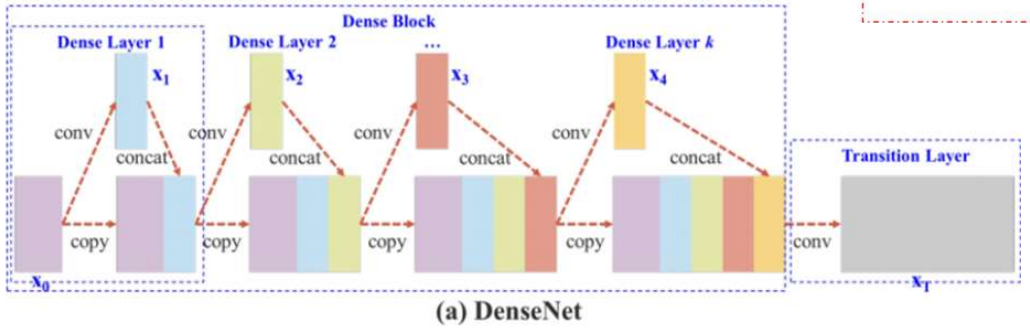
YOLOv4 Network Architecture



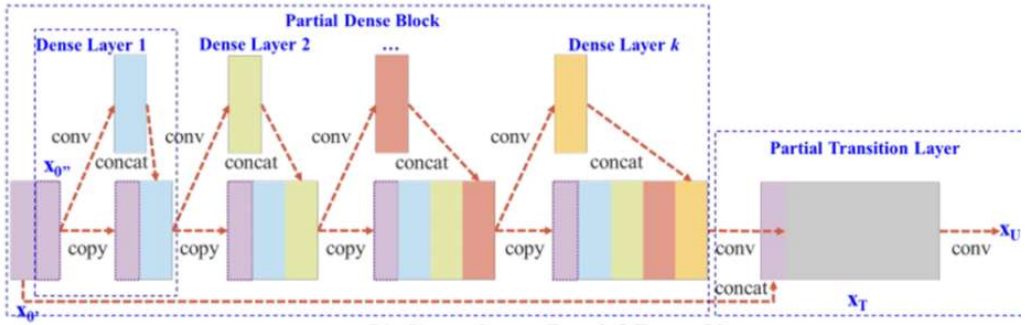
1. Bottom-up Path Augmentation
2. Adaptive Feature Pooling
3. Fully-Connected Fusion

Backbone





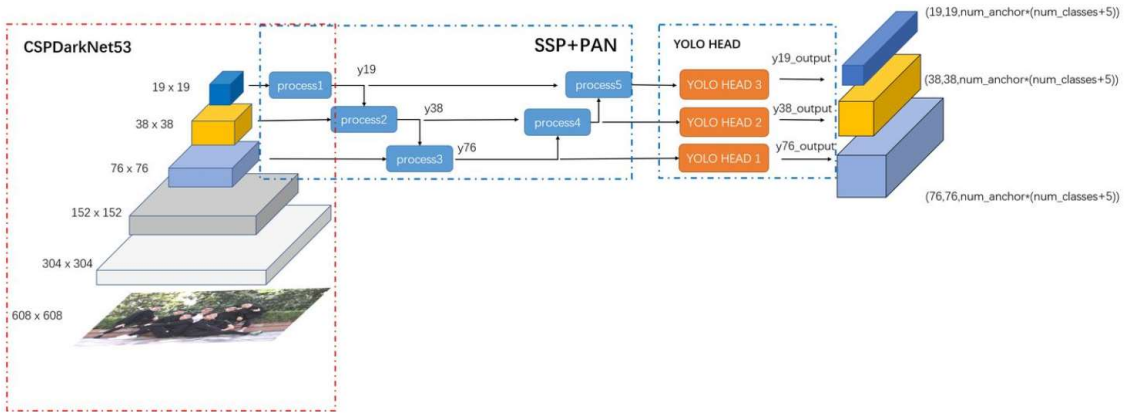
(a) DenseNet

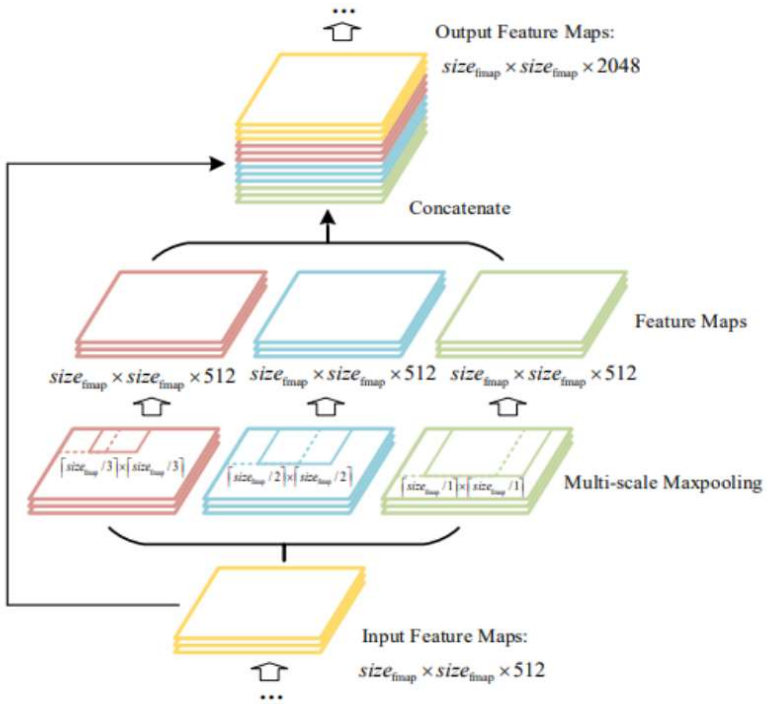


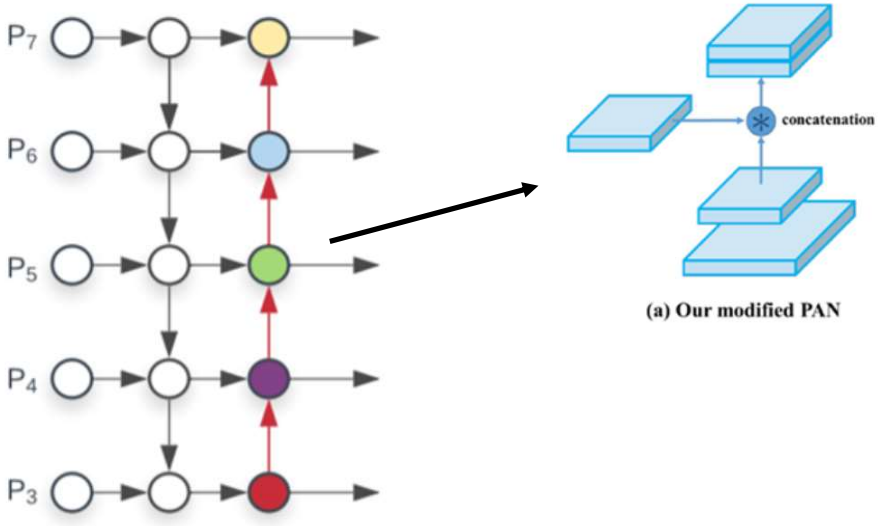
(b) Cross Stage Partial DenseNet

The method of CSP is utilizing the cross-stage feature fusion strategy and the truncating gradient flow to enhance the variability of the learned features within different layers.

Neck


CSPDarkNet53
SSP+PAN
YOLO HEAD
19 x 19
38 x 38
76 x 76
152 x 152
304 x 304
608 x 608
process1
process2
process3
process4
process5
y19
y38
y76
YOLO HEAD 3
YOLO HEAD 2
YOLO HEAD 1
y19_output
y38_output
y76_output
(19,19,num_anchor*(num_classes+5))
(38,38,num_anchor*(num_classes+5))
(76,76,num_anchor*(num_classes+5))


Output Feature Maps:
$size_{fmap} \times size_{fmap} \times 2048$
Concatenate
Feature Maps
$size_{fmap} \times size_{fmap} \times 512$
$size_{fmap} \times size_{fmap} \times 512$
$size_{fmap} \times size_{fmap} \times 512$
Multi-scale Maxpooling
Input Feature Maps:
$size_{fmap} \times size_{fmap} \times 512$


P_7
P_6
P_5
P_4
P_3
concatenation
(a) Our modified PAN

Why More Accurate ?

	Backbone	Detector
Bag of Freebies (BoF)	• CutMix • Mosaic data augmentation • DropBlock • Class label smoothing	• CIoU-loss • Cross mini-Batch Normalization • DropBlock • Mosaic data augmentation • Self-Adversarial Training • Multiple anchors for a single ground truth • Cosine annealing scheduler • Optimal hyperparameters • Random training shapes
Bag of Specials (BoS)	• Mish activation • Cross-stage partial connections (CSP) • Multi-input weighted residual connections (MiWRC)	• Mish activation • SPP-block • SAM-block • PAN path-aggregation block • DIoU-NMS

Example of Bag of freebies

Example of Bag of specials

CIoU loss, simultaneously considers three metrics: overlapping area, the distance between center points, and the aspect ratio. It is an extension to DIoU loss.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v$$

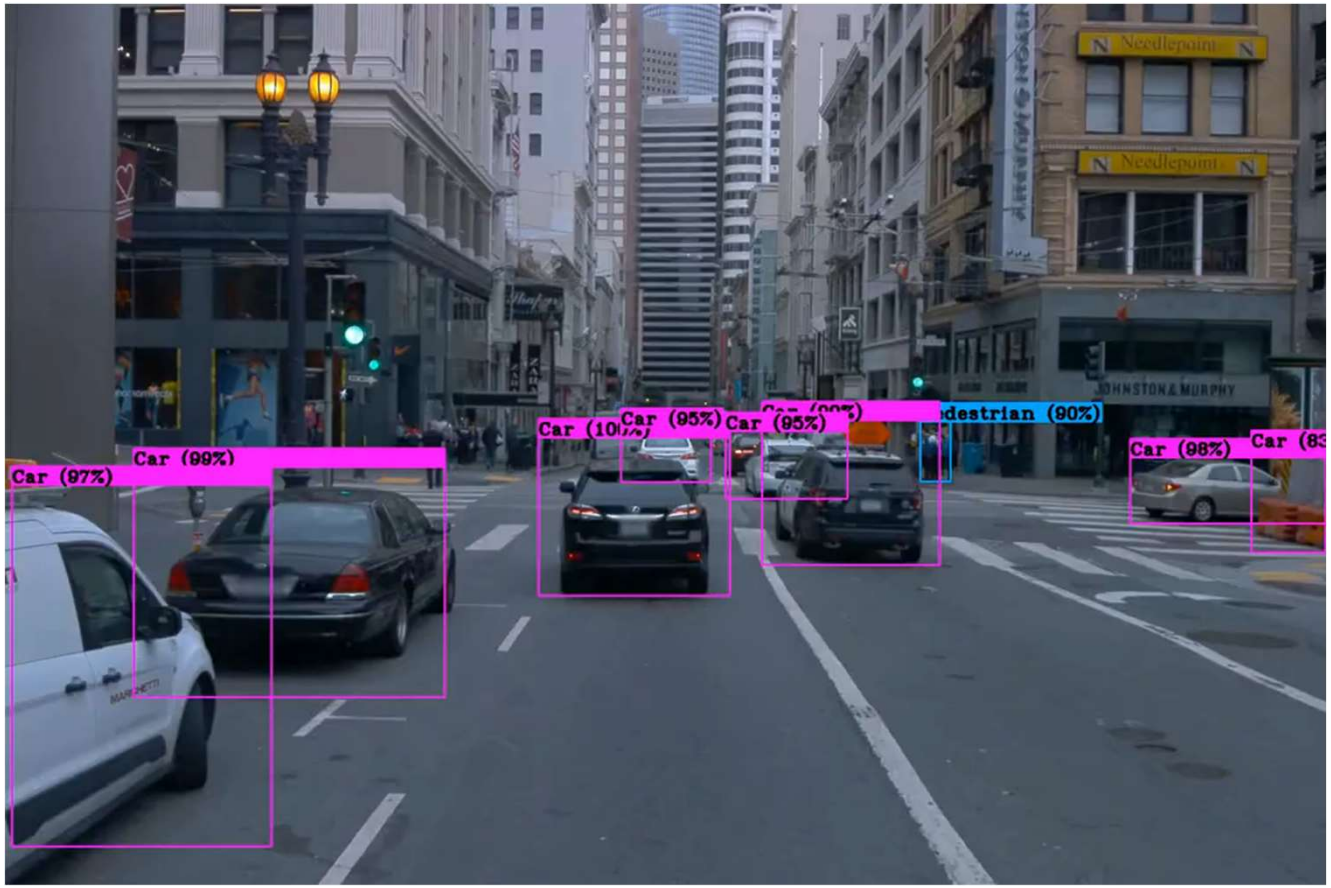
$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2.$$

$$\alpha = \frac{v}{(1 - IoU) + v},$$

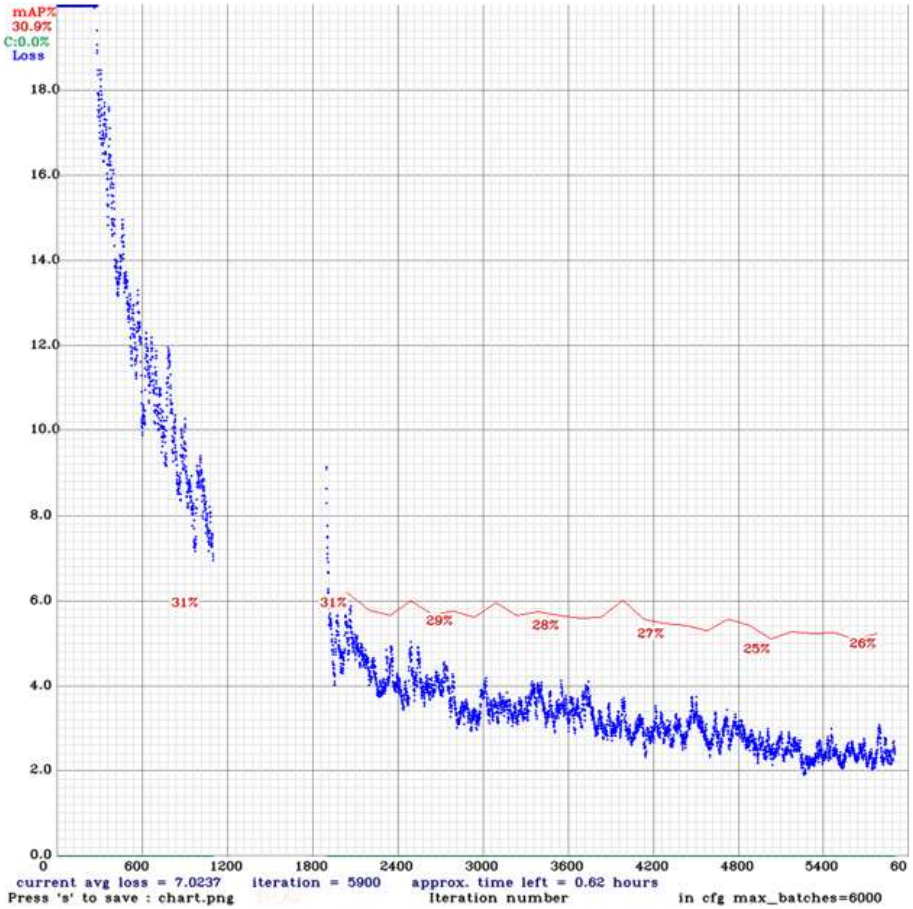
Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized. c is the diagonal length of the smallest enclosing box covering two boxes, and $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$ is the distance of central points of two boxes.

Experimental Result

Experimental Result



Why Overfitting ?



We had about 2400 pictures for training and very many of them were similar since the data was collected as sequential pictures from a car driving down the street.

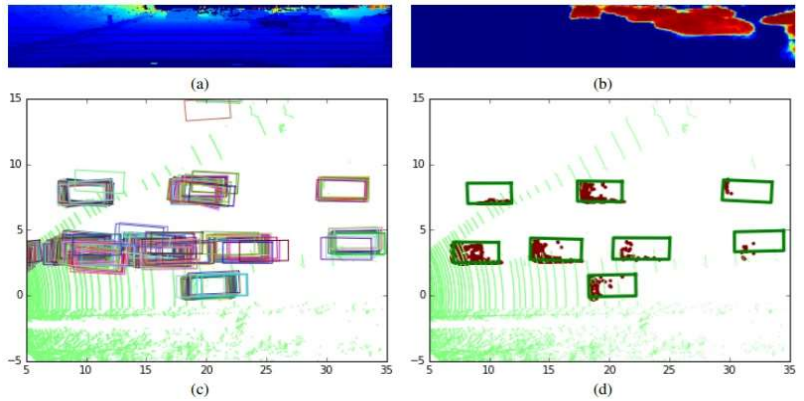
LDLS Content

- Related Work
- Sensor Fusion
- LDLS Framework
  - Pipeline
  - Procedure
- Result
- Comparison Method
- Evaluation & Conclusion

Related Work

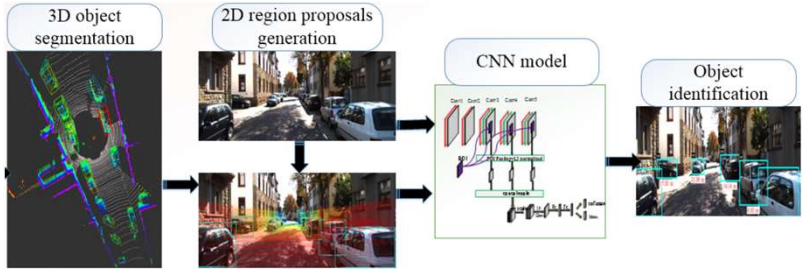
- Applying convolutional network on point cloud which is projected on 2D grid.

B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," *arXiv e-prints,* p. arXiv:1608.07916, 2016.



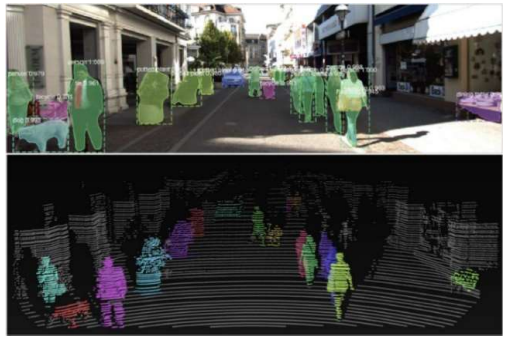
- 2D region proposals generation based on point cloud segmentation & projection.

X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LIDAR and Camera Data for Object Detection in Autonomous Vehicle Applications," *IEEE Sensors Journal,* vol. 20, no. 9, pp. 4901-4913, 2020.

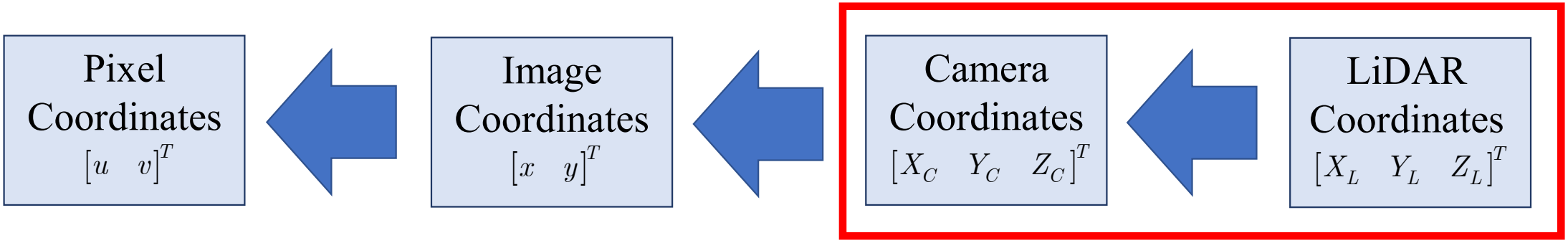


- 3D object detection based on point cloud projection & Mask-RCNN.

B. H. Wang, W. Chao, Y. Wang, B. Hariharan, K. Q. Weinberger, and M. Campbell, "LDLS: 3-D Object Segmentation Through Label Diffusion From 2-D Images," *IEEE Robotics and Automation Letters,* vol. 4, no. 3, pp. 2902-2909, 2019.

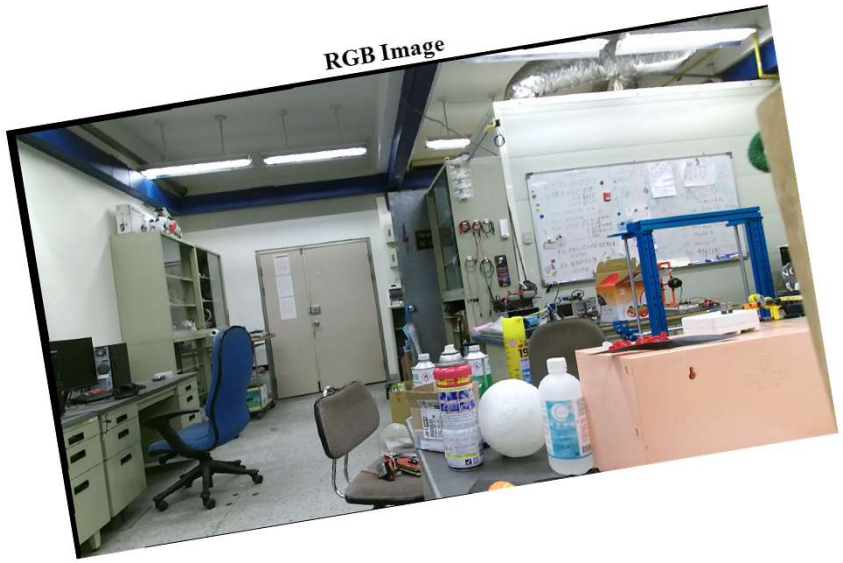


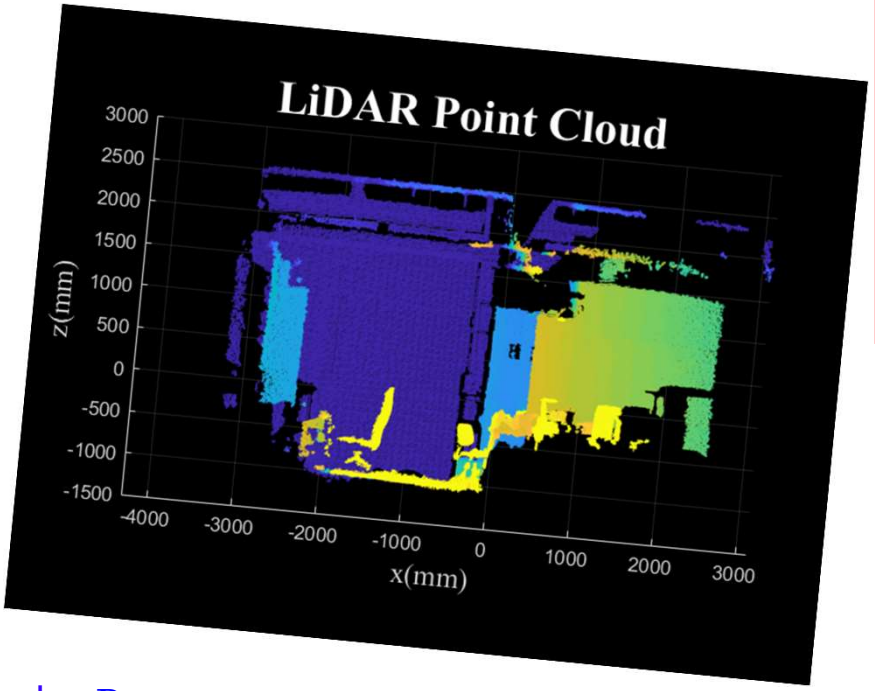
From LiDAR to Camera

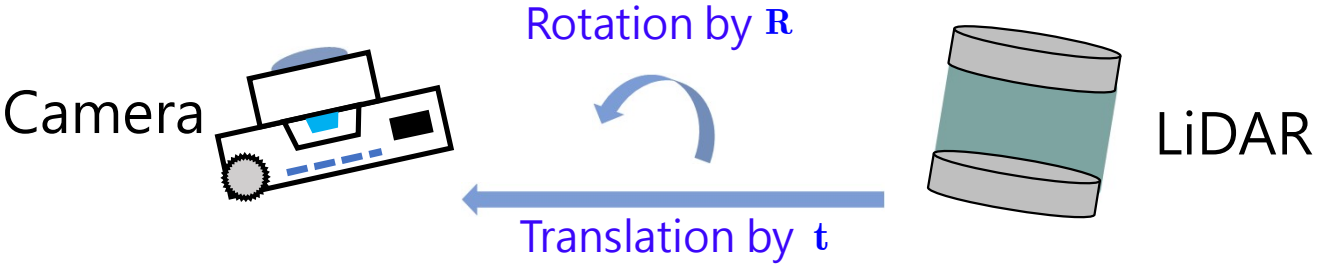


$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{(3\times1)} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}_{(3\times3)} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}_{(3\times4)} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}_{(4\times1)}$$

From LiDAR to Camera







LiDAR Coordinates
$[X_L \quad Y_L \quad Z_L]^T$

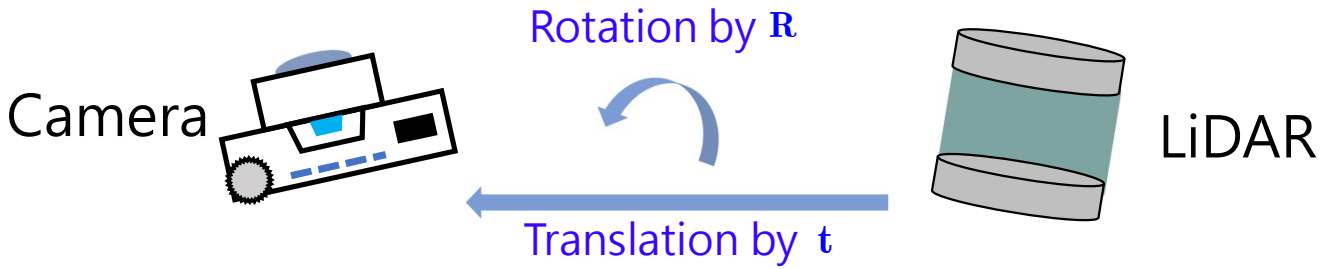
Camera Coordinates
$[X_C \quad Y_C \quad Z_C]^T$

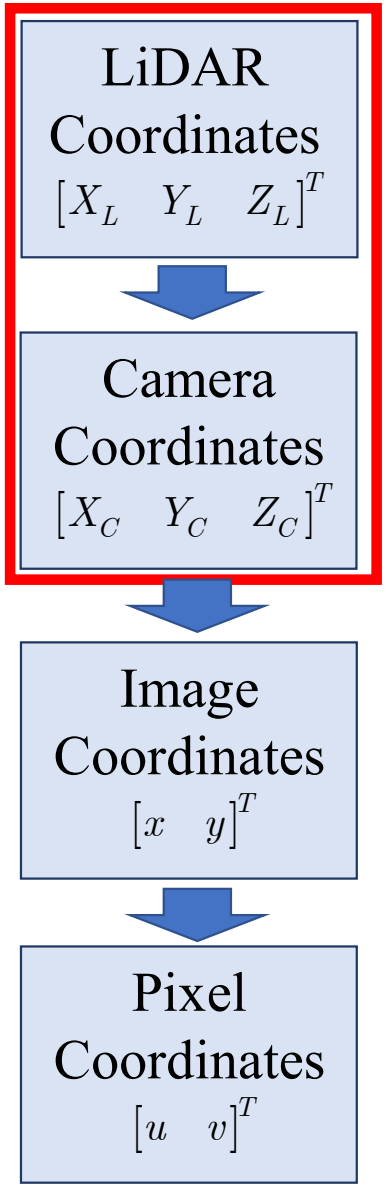
Image Coordinates
$[x \quad y]^T$

Pixel Coordinates
$[u \quad v]^T$

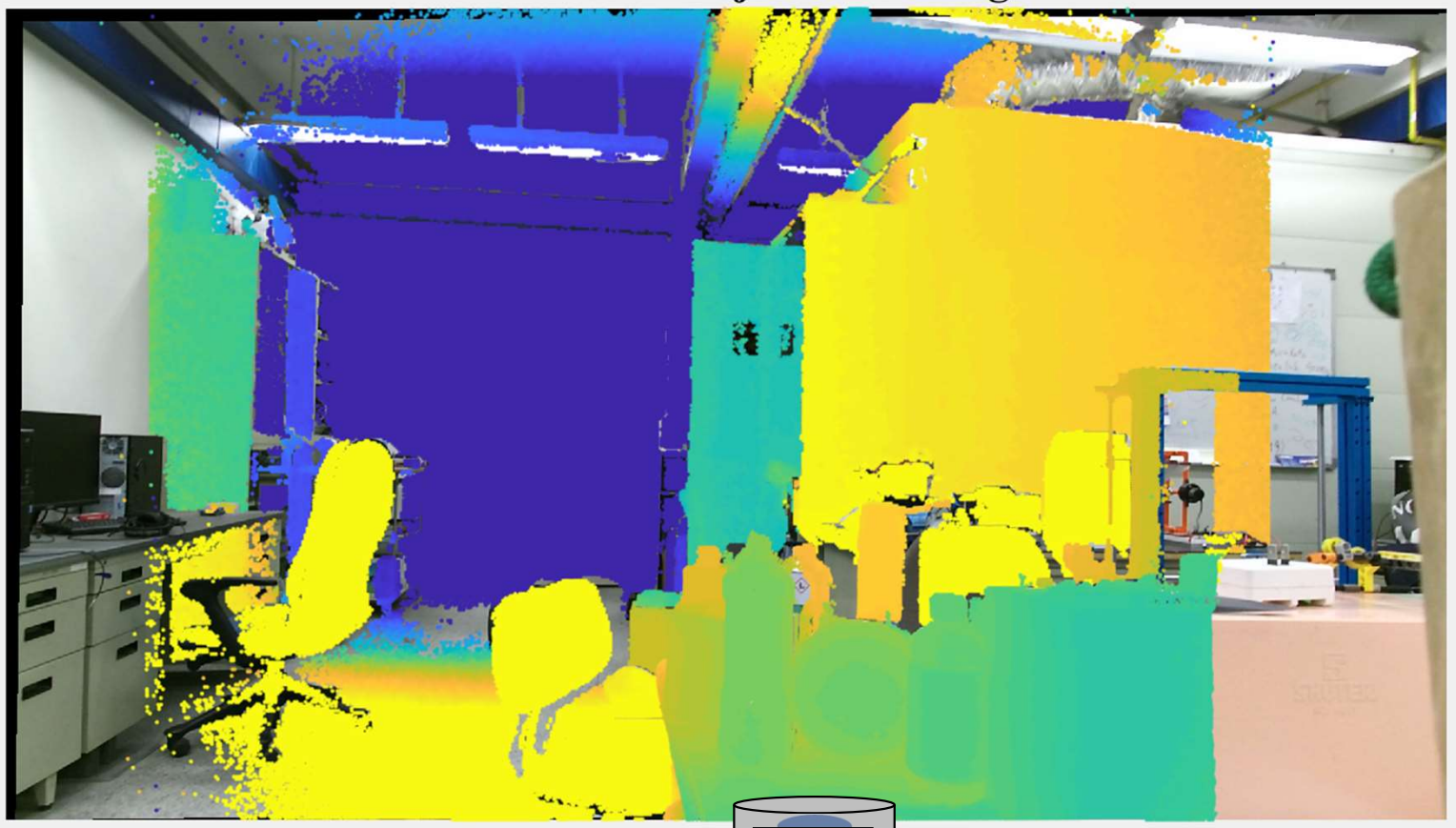
From LiDAR to Camera

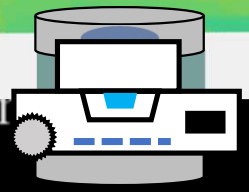
$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}_{(3\times1)} = \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}_{(3\times4)}}_{\text{Extrinsic Parameters}} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}_{(4\times1)}$$





Point Cloud Projected to Image





LiDAR Coordinates
$[X_L \quad Y_L \quad Z_L]^T$

Camera Coordinates
$[X_C \quad Y_C \quad Z_C]^T$

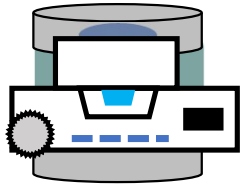
Image Coordinates
$[x \quad y]^T$

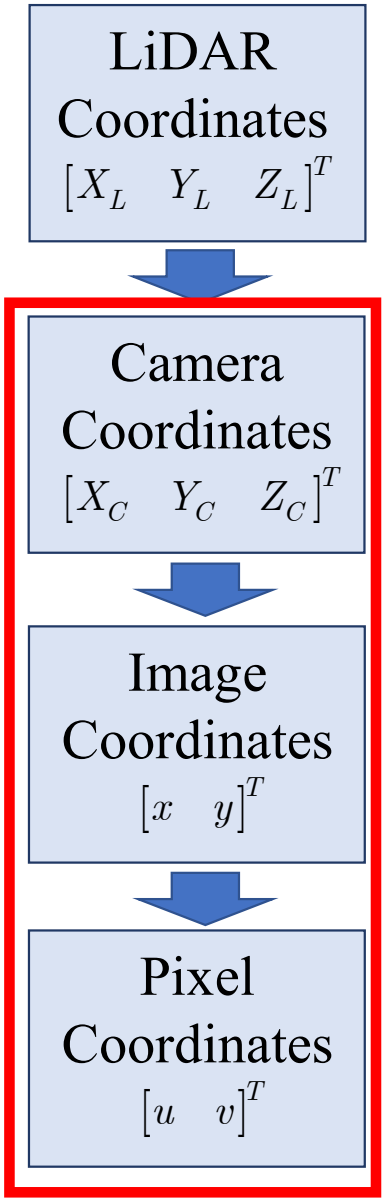
Pixel Coordinates
$[u \quad v]^T$

From LiDAR to Camera

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{(3\times1)} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}_{(3\times3)} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$

Intrinsic Parameters





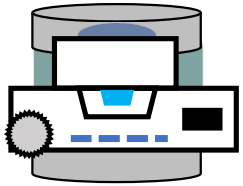
From LiDAR to Camera

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{(3\times1)} = \begin{bmatrix} f_u & & & \\ 0 & & & \\ 0 & 0 & 1 & \end{bmatrix}_{(3\times3)} \begin{bmatrix} & & 13 & t_1 \\ & & 23 & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}_{(3\times4)} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}_{(4\times1)}$$

Intrinsic Parameters

Extrinsic Parameters

Given !



LiDAR Coordinates $[X_L \quad Y_L \quad Z_L]^T$

↓

Camera Coordinates $[X_C \quad Y_C \quad Z_C]^T$

↓

Image Coordinates $[x \quad y]^T$

↓

Pixel Coordinates $[u \quad v]^T$

From LiDAR to Camera

Can we use 2D object detection result to identify 3D objects?

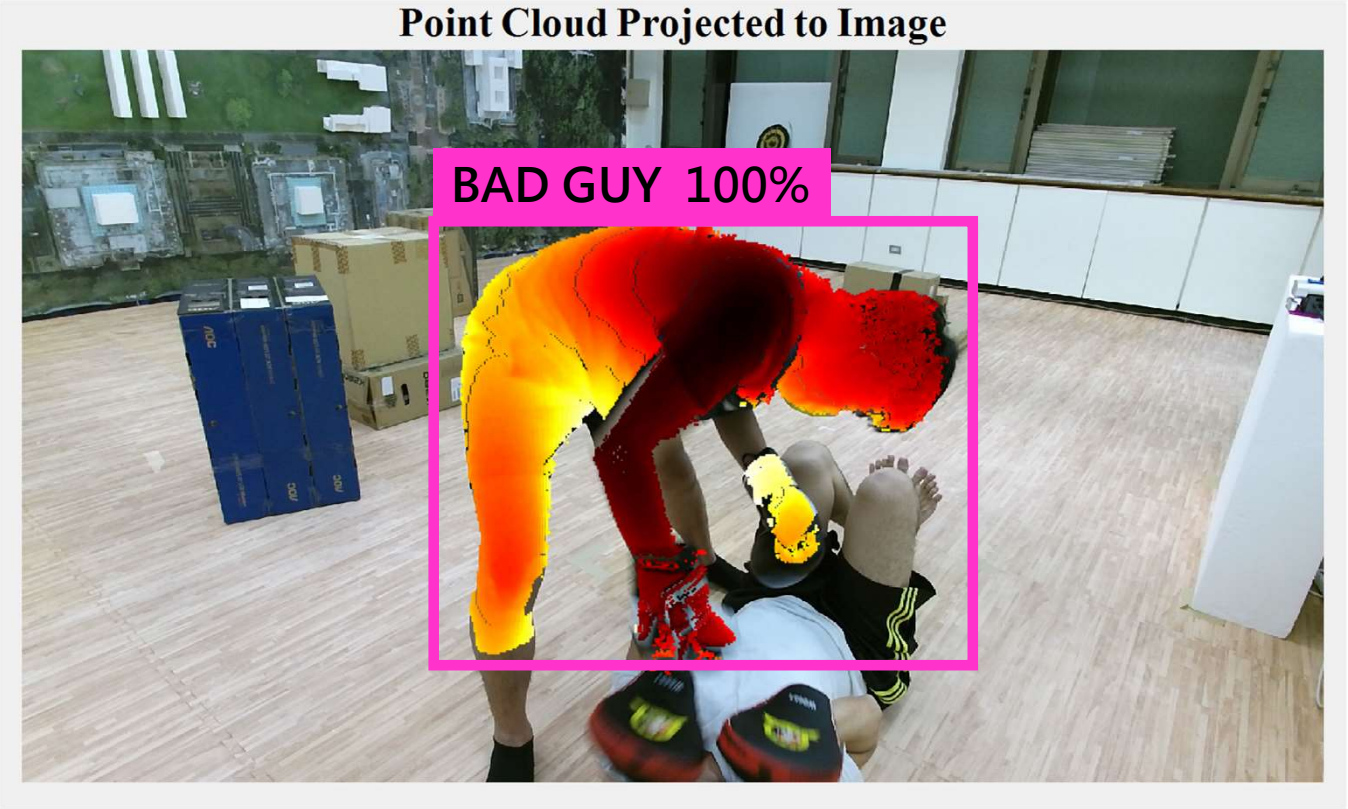
Do 2D object detection and get bounding box.

From LiDAR to Camera

Can we use 2D object detection result to identify 3D objects?

Do 2D object detection and get bounding box.

Project 3D point cloud to 2D image and register the points inside bounding box.



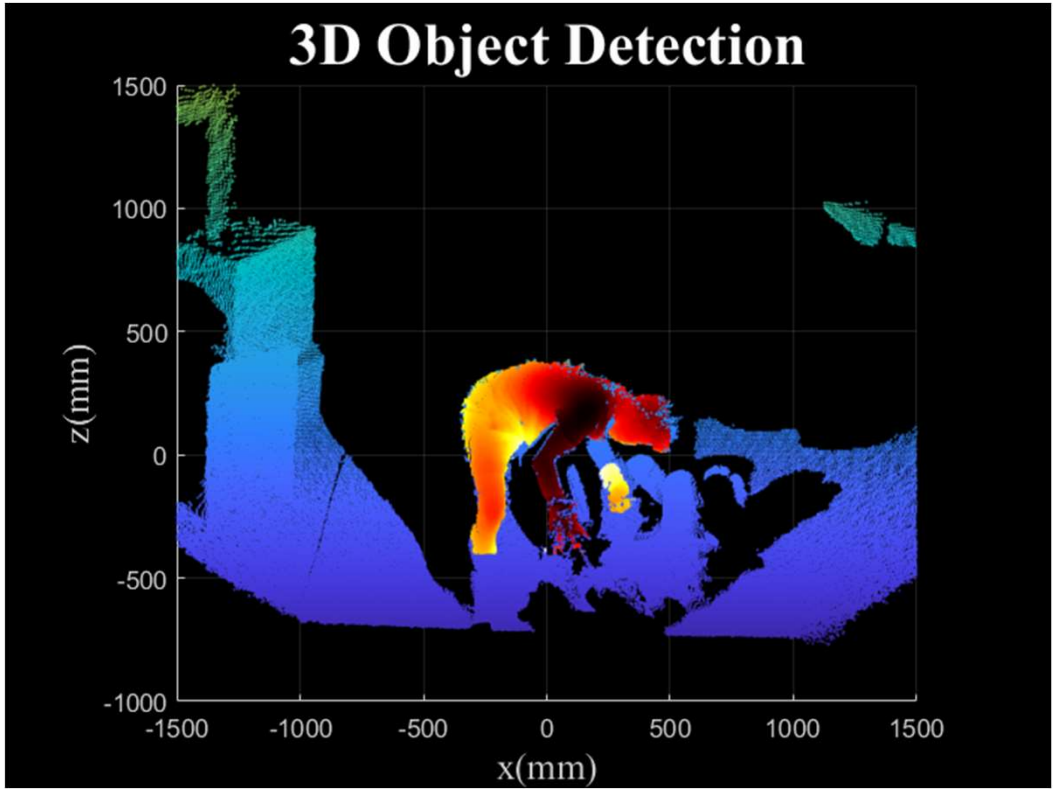
From LiDAR to Camera

Can we use 2D object detection result to identify 3D objects?

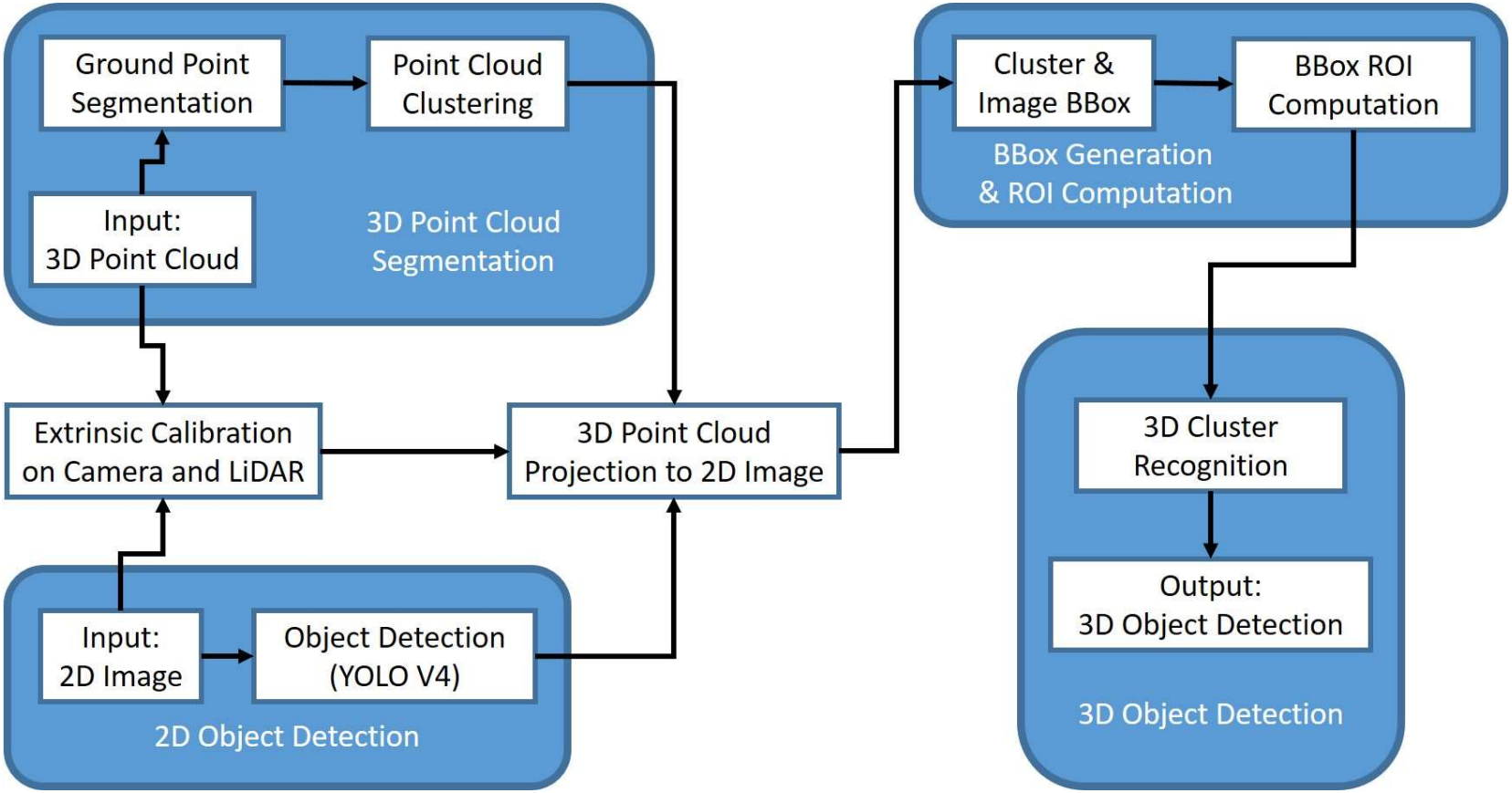
Do 2D object detection and get bounding box.

Project 3D point cloud to 2D image and register the points inside bounding box.

3D object detection complete !



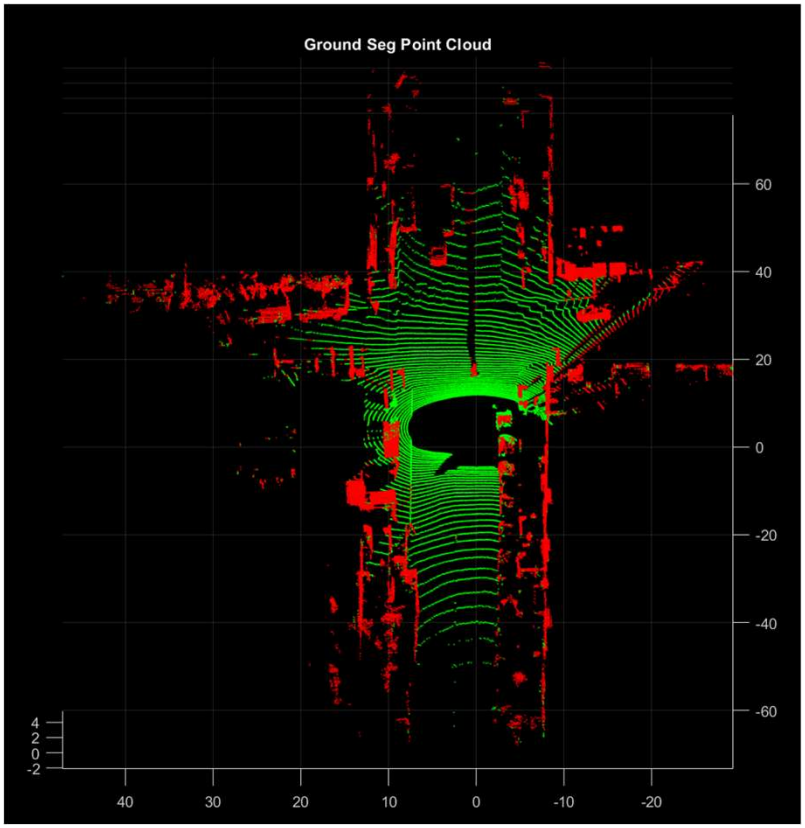
Pipeline


Ground Point Segmentation
Point Cloud Clustering
Input: 3D Point Cloud
3D Point Cloud Segmentation
Extrinsic Calibration on Camera and LiDAR
3D Point Cloud Projection to 2D Image
Input: 2D Image
Object Detection (YOLO V4)
2D Object Detection
Cluster & Image BBox
BBox ROI Computation
BBox Generation & ROI Computation
3D Cluster Recognition
Output: 3D Object Detection
3D Object Detection

Ground Point Segmentation

- For point cloud preprocessing, ground point segmentation is a vital step for further execution.

- For LiDAR data with order, we applied optimized scan line based ground point segmentation to distinguish ground points(green) & nonground points(red).

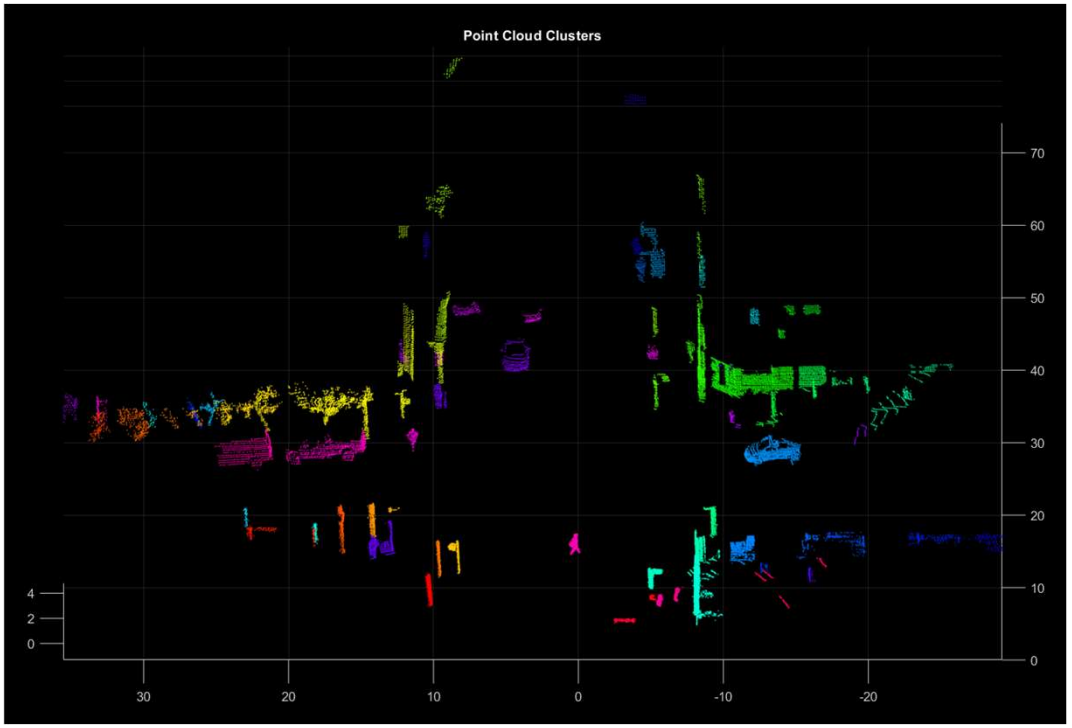


Y. Zhou *et al.*, "A Fast and Accurate Segmentation Method for Ordered LiDAR Point Cloud of Large-Scale Scenes," *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 11, pp. 1981-1985, 2014.

organization31415@gmail.com

Point Cloud Clustering

- We select point cloud field by its horizontal resolution based on camera vision orientation.
- Then we clustered point cloud based on Euclidean distance threshold and filtered out small clusters which don’ t have enough points.
- Next, we dyed remaining clusters into different colors for visualization.

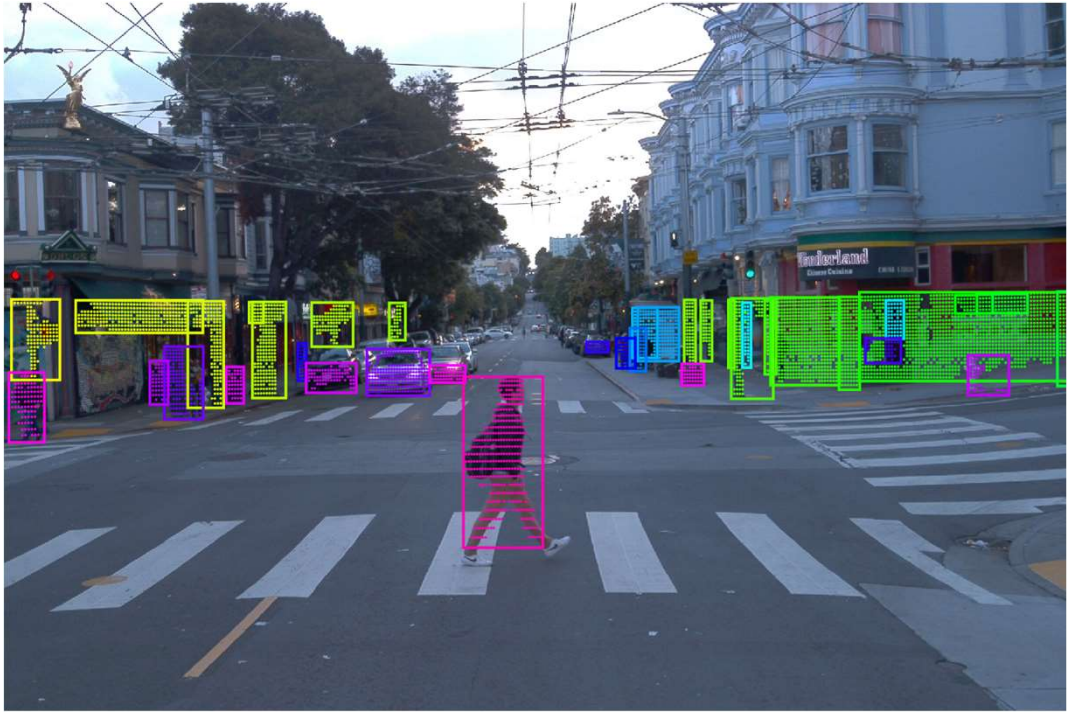


organization31415@gmail.com

Point Cloud Projection

- When point cloud clustering is completed, we project clusters into 2D image based on extrinsic calibration of LiDAR & Camera.

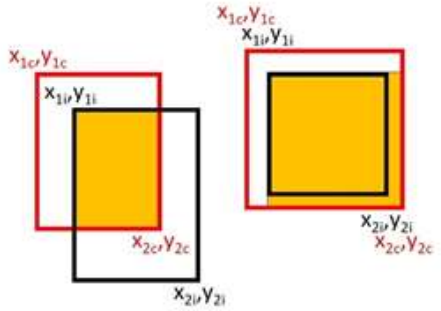
- After point cloud projection, we define cluster bounding box by max. & min. coordinate value of each cluster.



BBox ROI Computation

- By the output of YOLOv4 Object Detection, we can obtain 2D bound box on image.

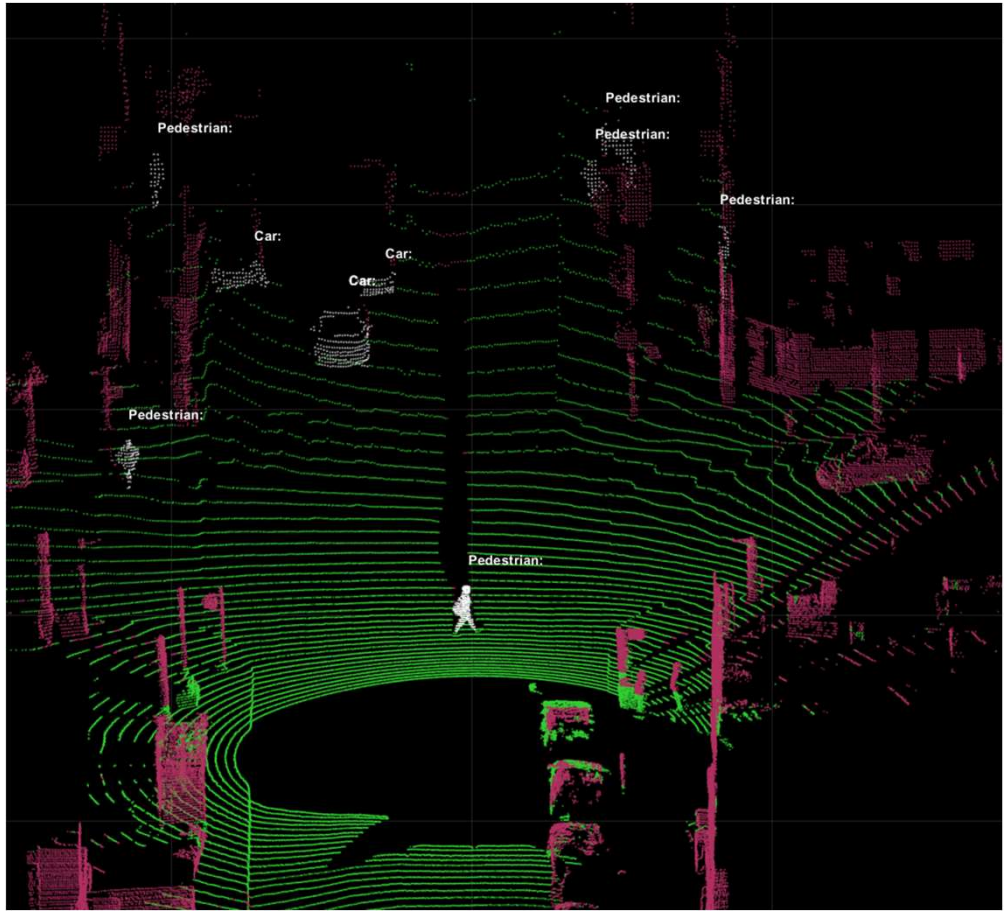
- With the relationship between cluster & image bounding box, we compute Region of Interest (ROI) and choose candidate cluster with highest score.



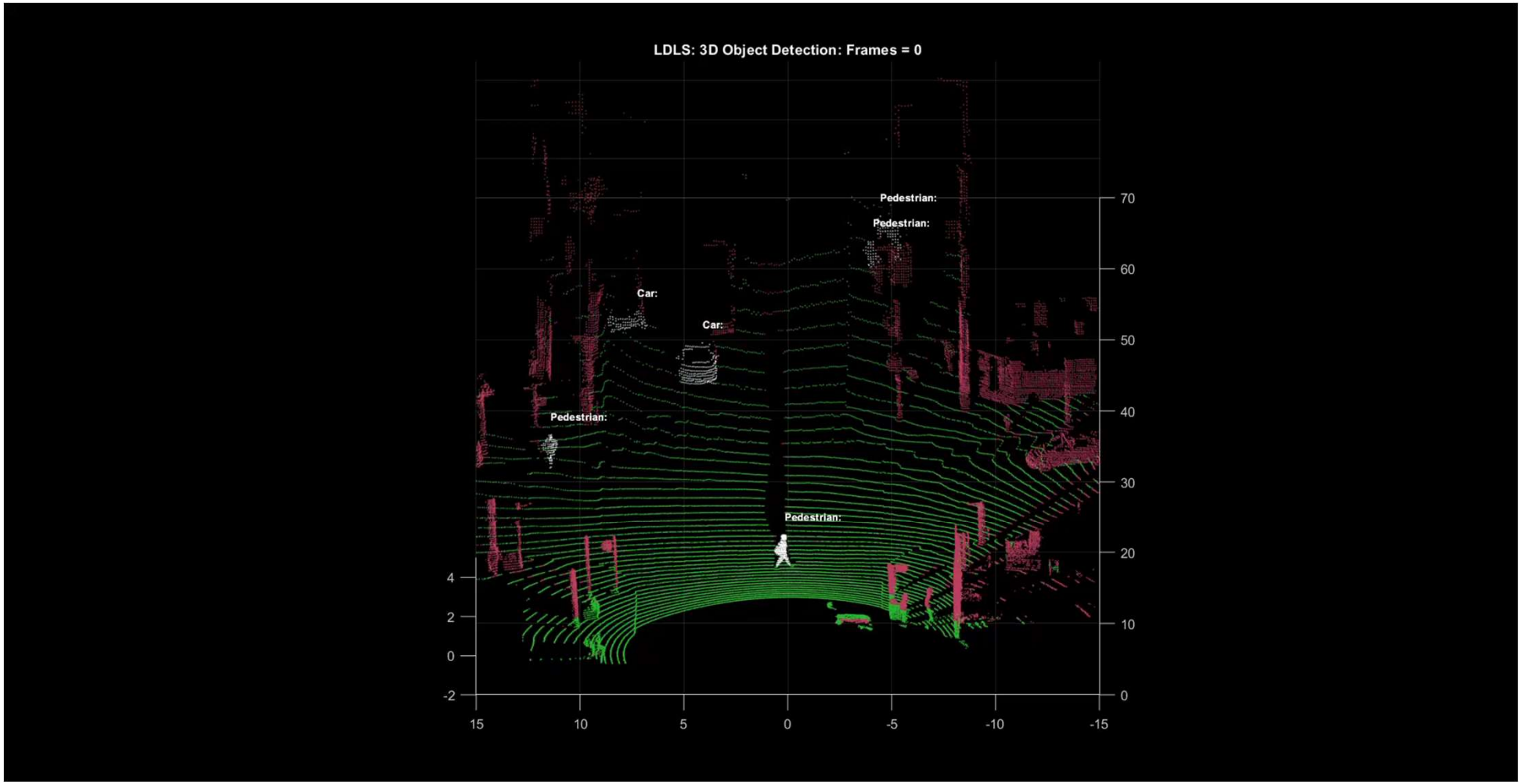
3D Cluster Recognition

- With the correspondence of chosen cluster index and BBox label generated by YOLOv4, we can recognize the category of each clustered point cloud.

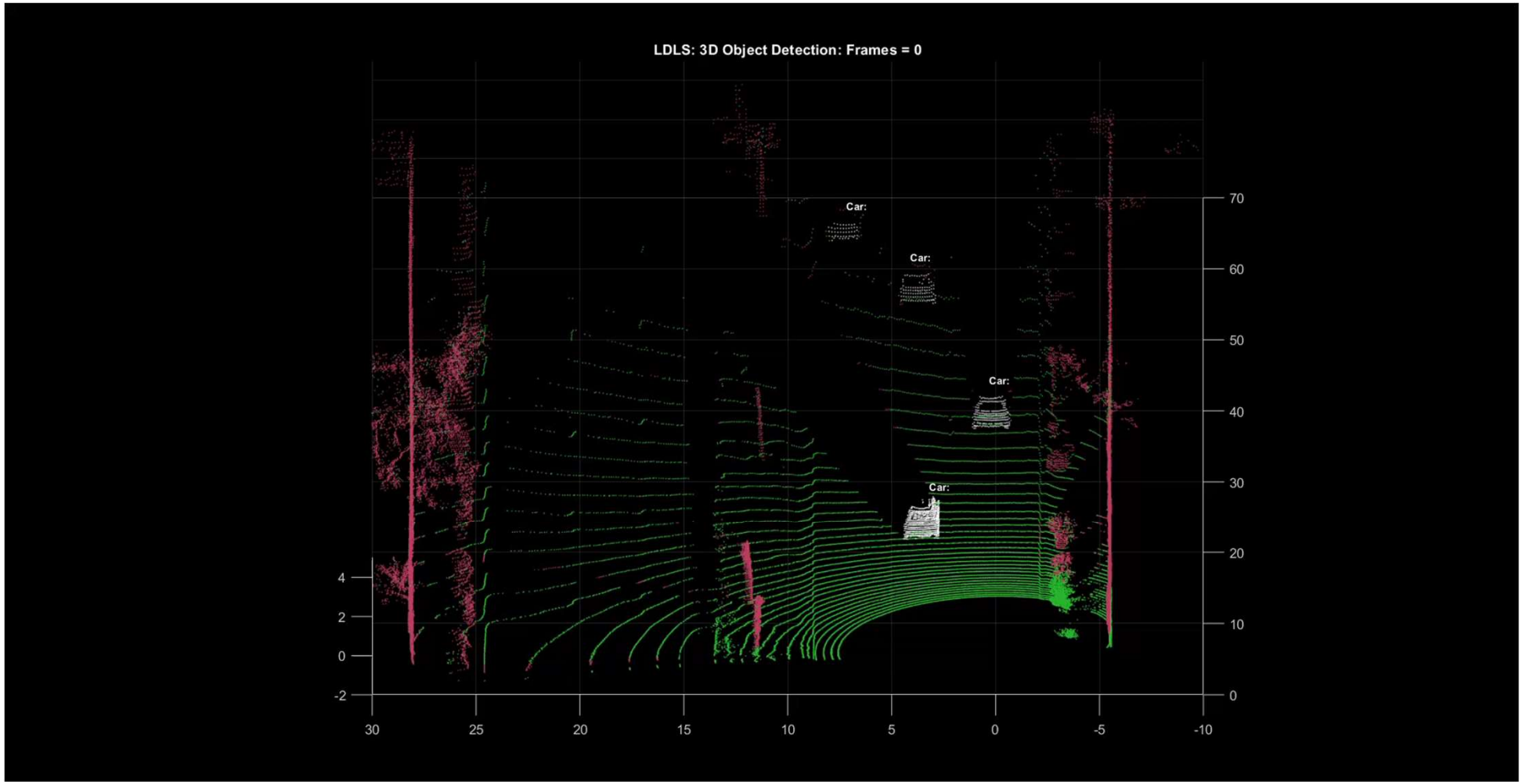
- Different color representative:
  - Green: Ground Points
  - Red: NonGround Points
  - White: Labeled Point Clusters



Result: City

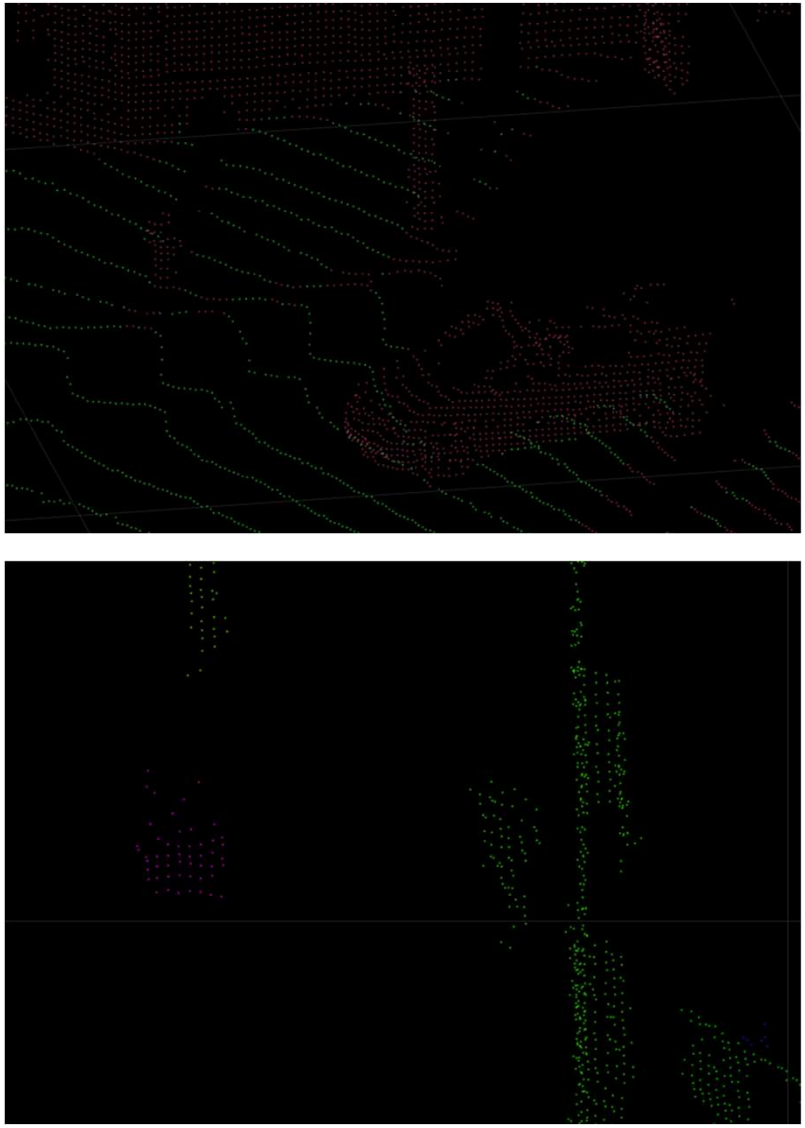

LDLS: 3D Object Detection: Frames = 0
Pedestrian:
Pedestrian:
Car:
Car:
Pedestrian:
Pedestrian:
70
60
50
40
30
20
10
0
4
2
0
-2
15
10
5
0
-5
-10
-15

Result: Highway

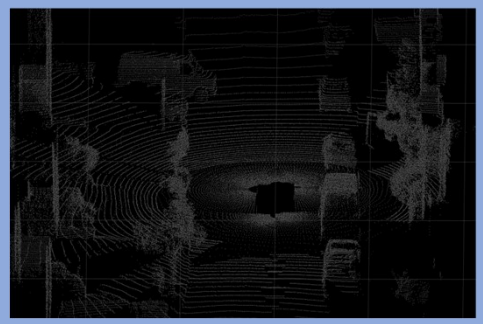


Evaluation

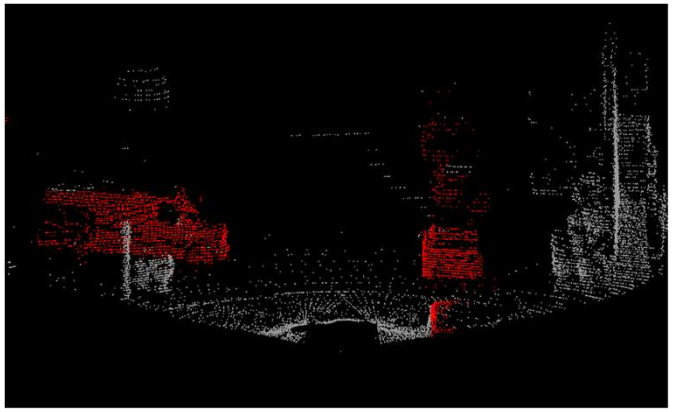
- The LDLS method has various constraints that needs to be improved:
  - The region of sensor fusion is limited in front camera, which leads to great amount of point cloud cluster unlabeled.
  - The quality of ground segmentation and point cloud clustering have a huge impact on 3D object detection, especially on label of pedestrian.

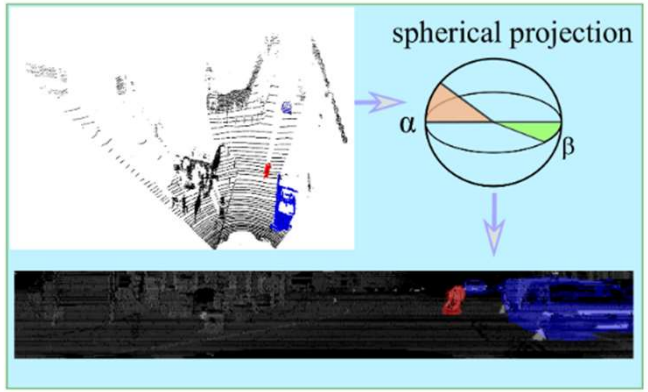


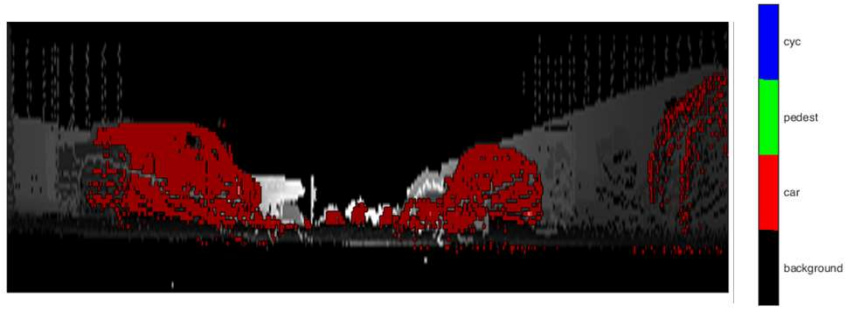
Comparison Method: Point Seg



Camera and LiDAR data from environment

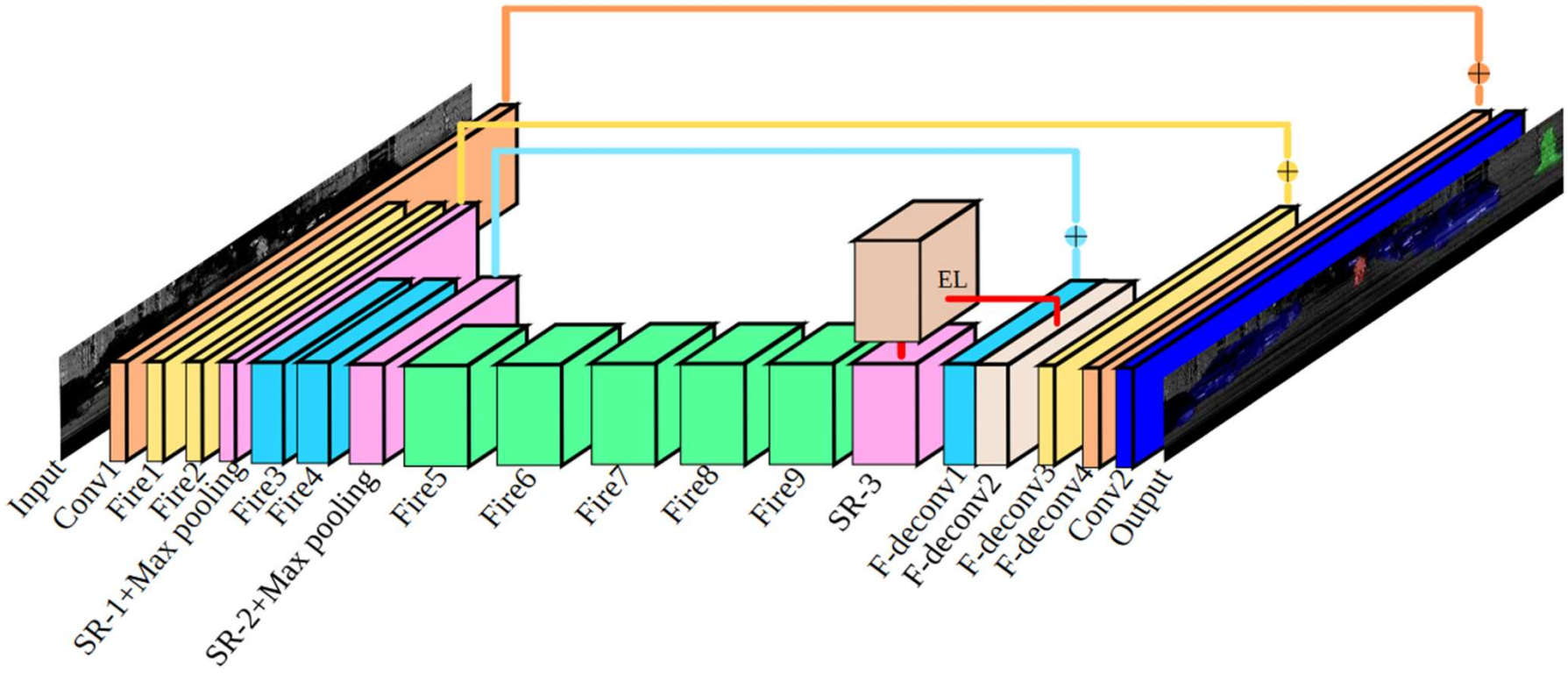




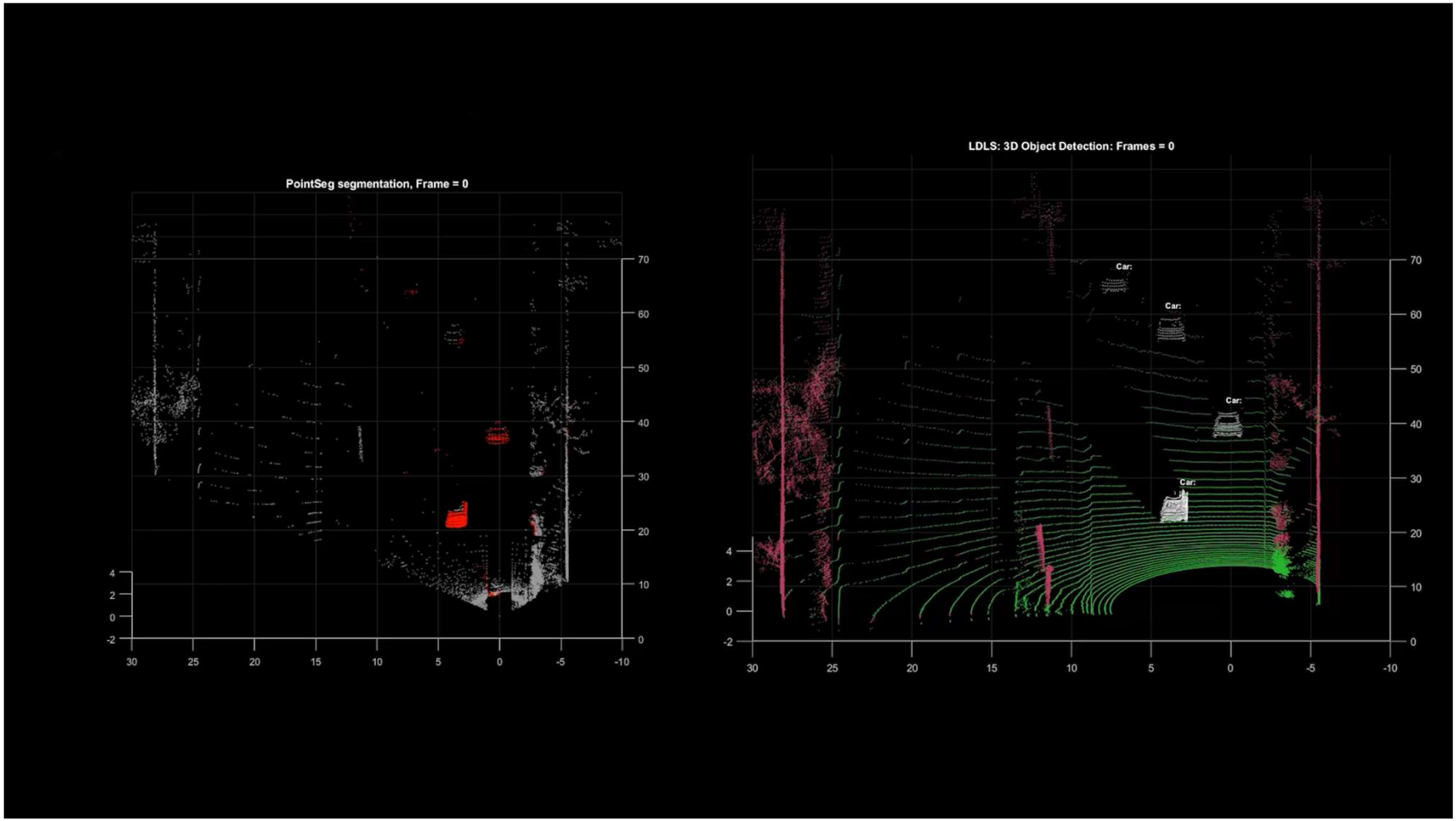


Wang, Yuan, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. "PointSeg: Real-Time Semantic Segmentation Based on 3D LiDAR Point Cloud." *ArXiv:1807.06288 [Cs]*, September 25, 2018.

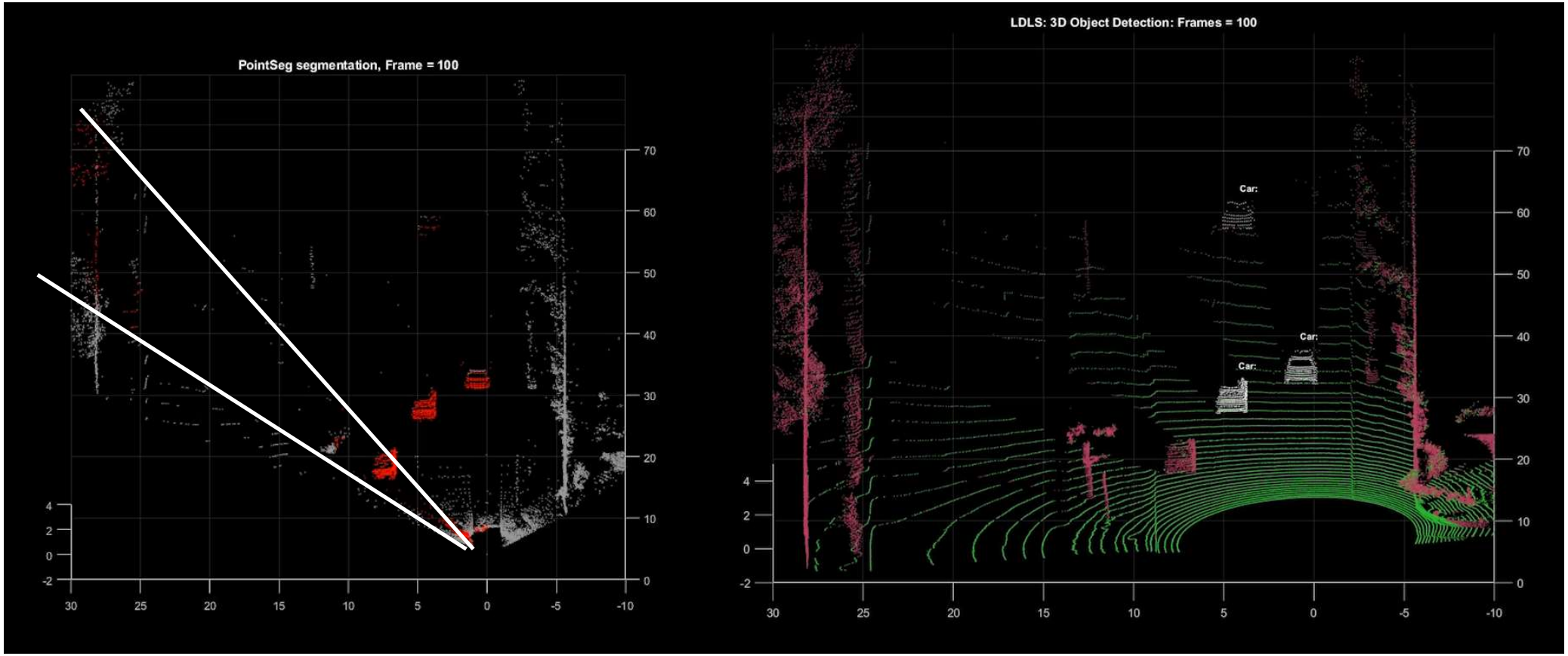
Point Seg Network Structure



Comparison Result on Highway



Point Seg: Object Edge



Evaluation on both Methods

- With the aid of point cloud clustering, LDLS constraints object set into certain area which avoids object edge issue in Point Seg.
- Compare to Point Seg which may lose tiny feature during spherical projection, sensor fusion method(YOLOv4+LDLS) can identify abundant objects which is benefit from precise YOLOv4 result.

Class	Method	Precision	Recall
Car	LDLS	0.951	0.939
	PointSeg	0.753	0.971
Pedestrian	LDLS	1	0.333
	PointSeg	0.106	0.074

Conclusion

- Sensor fusion is an effective solution for surrounding perception and position registration on autonomous vehicles.

- In the combination of accurate 2D object detection in YOLOv4 and corresponding 3D point cluster in LDLS, we can construct spatial environment sensing method with advantage of both camera and LiDAR sensor properties.