Object Detection in Autonomou Driving Vehicles

Adil Sadik [1] Qianli Song [1] Rui Chen [1]

[1]Department of Computer Science, Stanford University

Introduction and Problem statement

We worked on object detection and segmentation for autonomous driving vehicles. Given the snapshot images taken and LIDAR data points collected when the vehicle drives down the road, we need to output predicted 3D bounding volumes of objects on the road. The result is evaluated using mAP (mean average precision) under different IoU threshold levels. We implemented three different algorithms: YOLO algorithm, PointPillars network and feature pyramid network (FPN) with ResNet-18. Using the PointPillars network at 0.5 IoU, we achieved mAP of **0.6022** for car detection, **0.2313** for all objects detection and **0.2615** for three objects detection (car, pedestrian and cyclist). Using the FPN at 0.5 IoU, we achieved mAP of **0.97** for three objects detection (car, pedestrian and cyclist) under moderate difficulty in 3D object detection task.

Dataset

1. **Datasete 1**: For YOLO and PointPillars network, the dataset we use is the Lyft Object Detection for Autonomous Vehicles Dataset. Training image data has 158,757 images. Test image data has 192,276 images. Training LIDAR data has 30,744 items. Test LIDAR data has 27,468 items. YOLO mainly used the image data. PointPillars network mainly used the LIDAR data. We used 25% of data due to various constraints.
2. **Dataset 2**: For FPN, the dataset we use is the KITTI 3D Object Detection dataset. It has 7,481 training images and 7,518 test images
   -**Detection:** We mainly used the point cloud, which includes Velodyne point cloud data and training labels 15,000 point clouds. The training data is split into 3000/1000 training and validation sets.
   -**Segmentation:** We have to find another dataset for segmentation because the 3d detection dataset doesn't contain any labels for segmentation. The segmentation dataset contains 100 images and we manually annotated segmentation mask for road and lanes.
   -**Pre-processing:** The original image size is (160, 576). We reshape the images into 152x608. For detection, random horizontal flips are used to augment the data. For segmentation, no augmentation is performed.

Conclusion and Future Work

1. Both PointPillars network and FPN achieved good performances.
2. PointPillars network slightly underperformed. Potential reasons: difference in datasets, and insufficient data used for training.
3. For future work, we should run more experiments with various hyperparameters and model configurations, and also train and test the PointPillars network with full Lyft Object Detection Dataset.

Methods

YOLO-2D Object Detection: YOLO (You Only Look Once) involves a single deep convolutional neural network that splits the input into a grid of cells and each cell directly predicts a bounding box and object classification. We used the pre-trained weights of YOLOv3 system. YOLO is only used for qualitative analysis. Results are not shown here due to space constraints.

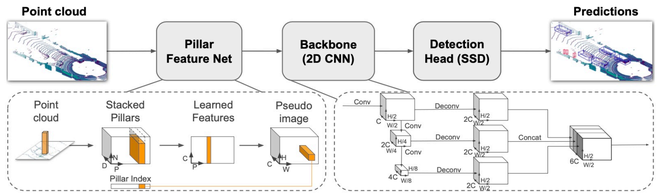

Figure 1. PointPillars Network Architecture overview

PointPillars Network-3D Object Detection (above): The main components of the network are a Pillar Feature Network, Backbone, and SSD Detection Head. The raw point cloud is converted to a stacked pillar tensor and pillar index tensor. The encoder uses the stacked pillars to learn a set of features that can be scattered back to a 2D pseudo-image for a convolutional neural network. The features from the backbone are used by the detection head to predict 3D bounding boxes for objects.

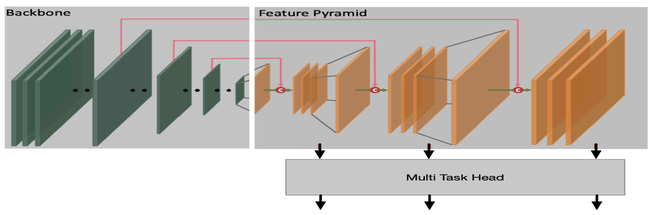

Figure 2. Feature Pyramid Architecture

FPN-3D Detection and Segmentation (above): We have used feature pyramid network (FPN) with ResNet-18 as the backbone. The model consists of a backbone layer and a feature pyramid network. The feature pyramid network a set of 1x1 conv layers with up-sampling to increase the spatial size of the final layers of backbone network. Skip connection is added between the backbone layers and Feature Pyramid layer to improve accuracy. For object detection, we used CenterNet detection algorithm. CenterNet detects the center point of each object class in the image as heatmaps and uses regression loss to find the bounding box of the detected object. The model is multi-headed.

Experiments and Analysis

IOU threshold	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
Animal	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bicycle	0.1096	0.0953	0.0817	0.0686	0.0571	0.0481	0.0412	0.0361	0.0321	0.0289
Bus	0.3925	0.3873	0.3794	0.3669	0.3500	0.3242	0.2909	0.2569	0.2285	0.2056
Car	0.6022	0.5866	0.5644	0.5331	0.4914	0.4407	0.3880	0.3410	0.3032	0.2729
Emergency Vehicle	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Motorcycle	0.0936	0.0716	0.0538	0.0419	0.0340	0.0284	0.0244	0.0213	0.0190	0.0171
Other Vehicle	0.4316	0.4226	0.4112	0.3957	0.3747	0.3464	0.3107	0.2743	0.2439	0.2195
Pedestrian	0.0728	0.0569	0.0440	0.0343	0.0277	0.0231	0.0198	0.0173	0.0154	0.0139
Truck	0.3798	0.3717	0.3606	0.3452	0.3258	0.2995	0.2686	0.2375	0.2113	0.1902
Average Over Car, Pedestrian and Bicycle/Cyclist	**0.2615**	**0.2463**	**0.2300**	**0.2120**	**0.1921**	**0.1706**	**0.1497**	**0.1315**	**0.1169**	**0.1052**
Average Over All	**0.2313**	**0.2213**	**0.2106**	**0.1984**	**0.1845**	**0.1678**	**0.1493**	**0.1316**	**0.1170**	**0.1053**

Figure 3. PointPillars network: mAP Per Category Per IoU Threshold

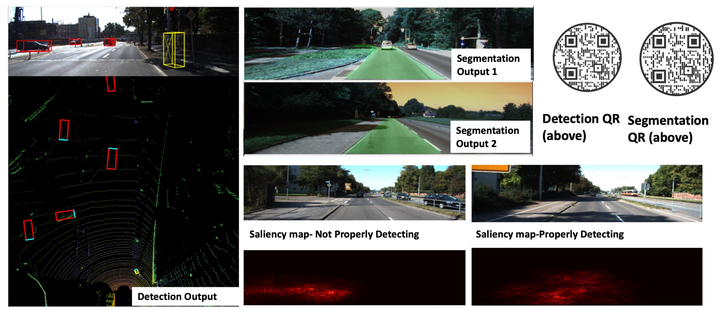

Figure 4. This figure contains outputs using FPN-3D Detection and Segmentation. **Detection Output:** bird's eye view of lidar image and the top image shows camera image. Please scan **Detection QR** code to see full detection output on Youtube. **Segmentation Image 1, 2:** Output from model-road/lane segmentation. Please scan **Segmentation QR** code to see full segmentation output on Youtube. **Saliency map** helps with error analysis in segmentation.

Task	Hard	Moderate	Easy	Num recall	Num epochs
bbox	30.04	33.28	37.45	11	10
3d	12.13	12.30	13.99	11	10
bbox	30.29	34.32	36.05	40	10
3d	12.79	12.85	13.08	40	10
bbox	72	78	80	11	300
3d	95	97	97	11	300
bbox	72	84	87	40	300
3d	97	97	98	40	300

Figure 5. FPN: mAP for 3D Detection Tasks (Average Over Car, Pedestrian and Cyclist)