3D Semantic Segmentation for Autonomous Cars

Yuntao Ma, Albin Mosskull, Alana Xiang

{yma42, mosskull, zxiang} @ Stanford.edu

Introduction

Solving the problem of making vehicles fully autonomous would bring great benefit to society. It could enable cheaper, faster and safer transportation of both people and goods, improving quality of life for all. In order to make autonomous vehicles truly safe, one of the most critical components is to have a good perception system.

A common and important task within the perception system is that of 3D Semantic Segmentation; classifying the semantic class for each pixel in a Lidar Image. We investigate this problem as part of the Waymo Open Dataset Challenge.

Background

There are two common classes of methods for 3D Semantic Segmentation.

Point-Based Methods

Point-based methods operate on a point-cloud that is generated from the Lidar data.

Projection Methods

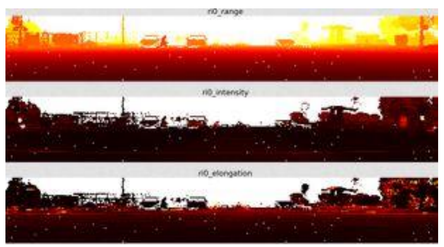
Projection methods project the point-cloud to a 2D representation, so that 2D methods can be used more easily. Since these methods could be applied directly to our data without using point-cloud representations, we choose to use a projection-based approach.

Problem Statement

- Input: Lidar Range Image, Camera Images
- Algorithm: Projection-based semantic segmentation network
- Output: A semantic label (such as “Car”, “Pedestrian”) for each pixel in the Lidar Image
- Main metric: mean Intersection over Union (mIoU)

Dataset

- Data from Waymo Open Dataset
- Time-aligned Lidar and Camera images from Waymo Vehicles driving around in the real world
- 23691 Training Samples
- 5976 Validation Samples
- 2982 Test Samples



Sample Lidar Image plotted as heat map, for the three channels Range, Intensity, Elongation

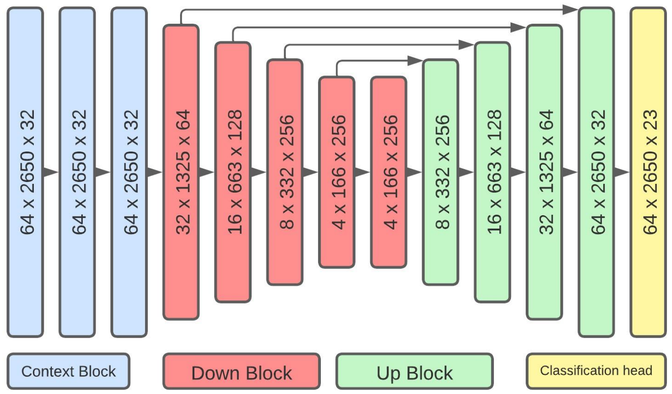
Sample Camera Image for two of the five cameras

References: Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast semantic segmentation of lidar point clouds for autonomous driving.

Method

SalsaNext

SalsaNext is one of the best projection-based networks, and we choose to implement it first. It uses a standard encoder-decoder architecture as seen in the figure below.



Two of the main adaptations of SalsaNext is the Dilated Convolution that is done to ensure the receptive fields capture a large enough part of the spatial structure. The other adaptation is a pixel-shuffle layer that is used in the upsampling.

ConvNeXt Encoder

The encoder block of the SalsaNext is based on ResNet, so we therefore tried using an improvement on ResNet, ConvNext, for the encoder. Thanks to our modular design of the models and training, a change like this could be easily implemented.

Late Fusion

To use pixel data, we project the pixels onto a Lidar image, so that we have a label for each pixel. This is a drastic downsampling as the cameras have a higher resolution.

Sample Projected Pixel Image

We then train a network on this input and combine its output with the Lidar-only model using a weighted mean, which we call naïve late fusion.

Results

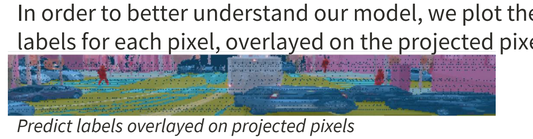
Result of Models

The results for the different models, losses and adaptions are shown in the Table below.

Backbone	Sensor	Loss Function	mIoU
SalsaNext	Lidar	Unweighted CE	0.38
SalsaNext	Cam	Unweighted CE	0.28
SalsaNext	Cam	WCE + LS	0.37
SalsaNext	Lidar	Poly1 WCE + LS	0.45
ConvNeXt	Lidar	WCE + LS	0.479
SalsaNext	Lidar+Cam	WCE + LS	0.490
SalsaNext	**Lidar**	**WCE + LS**	**0.492**

Visual Analysis

In order to better understand our model, we plot the predicted labels for each pixel, overlayed on the projected pixels:

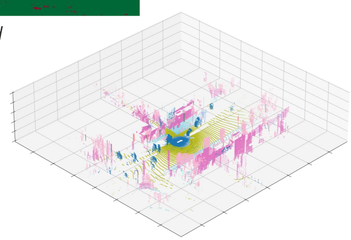


Predict labels overlayed on projected pixels

We can also plot which predictions the model gets correct:

Correct predictions in green, incorrect in red

Projecting the labels onto 3D yields a result as shown to the right



Competition Results

- Our best model achieves a mIoU of 0.56.
- Our results for a few classes are shown in the table to the right.
- The winner achieved a mIoU of 0.71, using a point-based method called Cylinder3D.

Class	IoU
Building	0.8938
Car	0.8705
Road	0.8702
...	
Traffic Light	0.2292
Other Vehicle	0.1918
Motorcyclist	0.0001

Conclusions

- Lidar-only SalsaNext worked the best
- Using Lovasz-Softmax + Class-weighted Cross-Entropy as loss was crucial for performance