# An Exploration of Neural Net Architectures for 3D Region-Proposal Networks and Object Detection

Shawn Hu
shawnghu

## Motivation and Problem Statement

Classically, in autonomous driving systems' perception modules, 2D object detection algorithms such as Faster R-CNN are used to get a 2d projected bound on obstacles in the images received from cameras. Frequently, radar or LiDAR systems or classical stereo vision techniques are used separately to determine the distance to these obstacles, and various task-specific algorithms may be used to combine both sets of input data into an internal representation of obstacles' 3D locations. These algorithms may be sensitive to various parameters, which may sometimes require a repeated, painstaking recalibration process.

However, instead of dealing with the task of fusing the 2D object detections with LiDAR input, it is possible to learn the task of inferring a 3D representation of obstacles directly. More specifically:

- Inputs: Images from a front facing camera and corresponding LiDAR point clouds
- Outputs: For each instance of a car, an oriented 3D bounding box parameterized by center coordinate, height, width, depth, and orientation angle.

## Dataset and Preprocessing

We work with the KITTI Vision Benchmark Suite [1], the canonical dataset and evaluation metric for 3d object detection in autonomous driving scenarios.

- 7481 training datapoints, split evenly between training and validation
- Mix of rural, urban, and highway scenes
- Up to 30 cars in a single scene

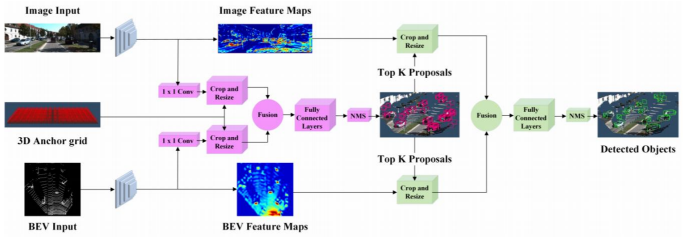


Figure: The model architecture. Architecture and figure inherited from AVOD [2]

## Model Architecture/Training Details

- VGG-like architecture of multiple convolutional layers used as extractor to produce 256-filter feature map
- 100k 3d anchors projected onto 7x7 "feature crops" in both inputs, crops fused and resulting vector used as input to RPN (multilayer perceptron) which outputs 1k proposals
- 1x1 conv used to "compress" feature crops before RPN to drastically reduce number of parameters, allowing RPN to fit in memory
- Uncompressed feature crops used as input to second network to predict class and regress exact location and orientation vector
- Optimized on weighted sum of cross-entropy for classes, smooth L1 loss for regression, orientation
- Trained with Adam optimizer with initial learning rate of 0.0001, exponentially decaying by a factor of 0.8 every 30k steps; multiple layers with 0.5 probability of dropout, l2 regularization on most weights of 0.0005

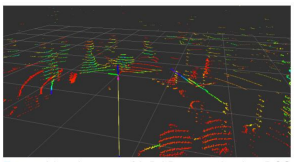

Figure: Visualization of LiDAR point cloud in ROS (not actually in the dataset), courtesy of [3]

Figure: Sample input image from the dataset.

## Modifications and Experiments

- Compressing 256 filters to 1 bottlenecks expression, hurts RPN performance- changed to 4
- Larger weighting on objectness loss from AVOD by factor of 2, to encourage better RPN performance
- "Fusion" of feature crops is just element-wise mean in original paper, at changed to concatenation and experimented with fusing after and in the middle of FC layers
- L2 regularization increased by factor of 4, dropout keep probability decreased to 0.35 to accommodate extra parameters

## Results

| AVOD | easy | medium | hard | mean |
|---|---|---|---|---|
| 2D Object Detection | 89.2 | 79.9 | 79.7 | 83.1 |
| BEV Object Detection | 87.8 | 78.4 | 78.0 | 81.4 |
| **3D Object Detection** | **80.0** | **66.0** | **65.7** | **70.6** |
| Our Model | easy | medium | hard | mean |
| 2D Object Detection | 89.6 | 87.0 | 79.7 | 85.4 |
| BEV Object Detection | 88.9 | 85.6 | 78.3 | 84.3 |
| **3D Object Detection** | **81.1** | **67.1** | **65.8** | **71.3** |

Table: Mean average precision of AVOD compared to our model on three tasks. "easy", "medium", and "hard" are labels given by the dataset, based on the amount of occlusion and the size of the ground-truth bounding box.

Figure: A sample detection from our trained model. Red/yellow represent ground truth where green presents our model's prediction.

## Future Work/Extensions

- Train models which detect multiple classes simultaneously
- Experiment with deeper feature extractors for the LiDAR input
- Explore other ways of obtaining a high-recall RPN within memory constraints, e.g, a smaller feature map with more filters for the 1x1 convolution

## References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun.
Are we ready for autonomous driving? the kitti vision benchmark suite.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander.
Joint 3d proposal generation and object detection from view aggregation.
*arXiv preprint arXiv:1712.02294*, 2017.

[3] from the public page of the rochester institute of tech senior design program.
http://edge.rit.edu/edge/C15504/public/Pictures.

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun.
Vision meets robotics: The kitti dataset.
*International Journal of Robotics Research (IJRR)*, 2013.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.
Faster R-CNN: Towards real-time object detection with region proposal networks.