Semantic Segmentation of 3D Lidar Points Cloud for Autonomous Driving

Chinmay Kate
M.S. Robotics Engineering
Worcester Polytechnic Institute (WPI)
Worcester, MA 01609
Email: cskate@wpi.edu

***Abstract*—Previous work has shown a promising results with deep-learning based methods for 3D point cloud semantic segmentation. In this paper, we study the problem of semantic segmentation for full 3D Lidar point cloud in real time. Most of the current techniques are only able to be trained and operate in small-scale point clouds. We Introduce new architecture called *RangeNet-PS* which is the modified version of RangeNet architecture. This pipeline exploits computationally efficient pre/post processing techniques combining with the 2D Convolution Neural Network (CNN) which further exploits the range images obtained as an intermediate representation. RangeNet-PS can accurately performs semantic segmentation of 3D Lidar Point cloud at sensor frame rate. RangeNet is encoder-decoder *DarkNet53* based architecture, we modified encoder of RangeNet with a new dilated convolution stack increasing receptive fields followed by *Pixel Shuffle (PS)* layer in the decoder. We also provide a complete evaluation on Semantic-Kitti Dataset, which shows our architecture outperforms baseline RangeNet architecture in runtime and class accuracy.**

***Index Terms*—Semantic Segmentation, DarkNet53, KNN Post Processing, RangeNet, RangeNet-PS, Optimization.**

Fig. 1. Velodyne HDL-64E Raw Laser scan data from Semantic-Kitti Dataset

I. Introduction

Semantic scene understanding is critical in Autonomous driving working in dynamic and real scenes. It enables the vehicle to perceive and interpret its surrounding accurately. By analyzing the scene, the autonomous vehicle can make informed decisions and take appropriate actions to navigate through complex environments safely. The Autonomous vehicle can detect and recognize objects such as pedestrians, other vehicles, road signs, traffic signs etc. This information is crucial in making decisions about speed, direction and braking. It also helps vehicle to plan its path and navigate through complex environments. The vehicle can identify potential obstacles and plan alternative routes to avoid them. By understanding scene, the autonomous vehicle can detect potential hazards such as potholes, debris or construction zones and take appropriate actions to avoid them. Also, by accurately perceiving and interpreting its surrounding, autonomous vehicles can reduce risk of accidents and ensure safe navigation which is an important component of Autonomous vehicle.

For Perception in high end vehicles, high precision modality is required. Cameras provide rich information but has it's drawbacks like its sensitivity to lightning conditions in environments where glare, show, shadow affects the quality of the images. In conditions of low lightning and noisy environment(Intense/bright light) can cause inaccuracies in depth measurement. Cameras can see what is within their field of view and many times objects in the scene are occluded. This can be solved having multi-camera system to cover field-of-view.

LiDAR (Lighting and Distance Ranging) sensor provides distance measurement and field-of-view exceeds that of radars and ultrasonic sensors. These sensors are robust in intense and low lightning conditions. Also it has little-no effect on performance in adverse weather conditions with snow, rain , fog and works better than the cameras. Previous works has shown promising results with deep-learning based methods for 3D LiDAR point cloud semantic segmentation. In case of 3D points obtained by LiDAR, Semantic segmentation assigns a class labels to each data point and segment the entire scene to perceive and interpret its surrounding accurately.

In this work, we focus on implementing Semantic Segmentation using RangeNet-PS architecture for LiDAR point cloud on Semantic-Kitti dataset. We project 3D points to spherical representation and use range images as feed to our architecture. RangeNet architecture produces state of art results, but reported run-time is significantly higher. We modify original RangeNet architecture with new modified dilated convolution stack which has improved receptive fields. Global context information gathered by receptive fields plays a crucial role in learning complex co-relation between classes. We also add

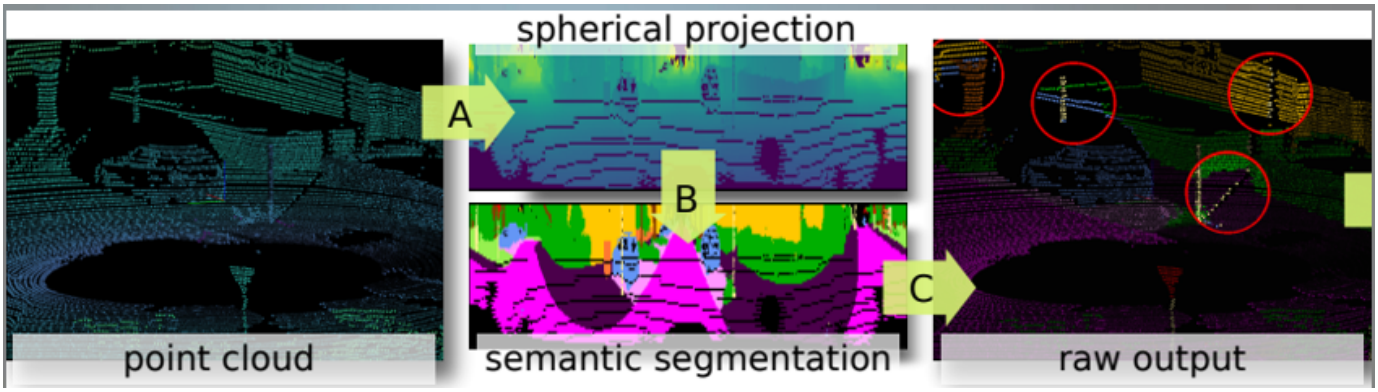

Fig. 2. Overview of this approach.

pixel shuffle layer in decoder which reduces no of parameters in the network and improve its run time. We effectively reconstruct original point cloud with semantics without discarding any points from original LiDAR point clouds.

A. Research Contribution

As contribution to the research community, we try to introduce some changes in the RangeNet architecture and present our RangeNet-PS architecture with complete evaluation on Semantic-Kitti dataset. In Summary, this work will have following contributions:

1) Added dilated convolution stack in encoder which has improved receptive fields and thus the global context information between the classes.
2) Addition of Pixel shuffle layer (first introduced in [1]) in the decoder, which significantly decreases no of parameters in the architecture thus improving run time.
3) Comparison of Results with an existing RangeNet architecture.

II. Related Work

Semantic segmentation of 3D LiDAR point cloud has been active area of research in recent years, and there have been numerous approaches proposed to address this problem. As broad comparison in [2], there are two approaches in deep learning based semantic segmentation of 3D LiDAR point cloud: Pointwise and Projection-based Neural network. Pointwise approach operates on raw 3D point cloud without any pre-processing, later project the un-skewed into 2D representation such as 2D images exploiting range values of the point cloud. Projection based approach has achieved significantly higher accuracy and is faster when it comes to run time. Whereas Pointwise approach has less no of parameters and can't scale up to large point efficiently due to less representational capacity.

PointNet (Qi et al. 2017 [3]) introduced neural network architecture that can directly process unordered point set as input, which is well suited for 3D point cloud processing. It uses raw unordered point clouds and apply symmetrical operators that are able to deal with this ordering problem. Further it uses Max Pooling to gather features and produce permutation-invariant feature extractor. This method looses the ability to correlate with global context and capture spatial relationship between features. This is further solved by PointNet++ using Hierarchical approach.

VoxelNet (Zhoul et al. 2018 [4])proposed a method that voxelies 3D LiDAR point cloud and applies 3D CNN to perform object detection. This method is extended to perform semantic segmentation by predicting the object category for each voxel. SPLATNet (Su et al. 2018 [5]) used this technique and represented this problem by projecting in high dimension sparse lattice. But due to High computational and memory cost this method couldn't be further elevated.

SqueezeV2 (Wu et al. 2019 [6]) using spherical projection of point cloud exploring 2D convolution, improved model from previous version performing better with addition of Context Aggregation Module (CAM) used to combat dropout noise in LiDAR point cloud. Furthermore light-weight fully CNN with conditional random fields are applied to boost its accuracy. Achieved good accuracy with runtime faster tan sensor rate 10Hz.

RangeNet++ (milioto et al. 2019 [7]) proposed two-stage neural network architecture that first performs spherical projection representation then to image coordinates and Secondly, full 2D CNN is applied to range images to perform semantic segmentation. Later they perform GPU based KNN technique to reconstruct back to 3D Lidar points with improved accuracy. The last step enables retricals of labels for all original point cloud. This method accurately segments entire 3D point cloud scan faster than sensor frequency (10Hz). We use this as our base approach.

III. Proposed Method

Goal of this proposed implementation is to achieve faster and accurate semantic segmentation on 3D LiDAR point cloud than the Velodyne HDL-64E sensor for Autonomous Driving. Our approach is based on RangeNet architecture which uses range image projection as input to the 2D fully CNN architecture. We then reconstruct back to original points cloud with labels to each points in 3D space. Following are the overview of these steps:

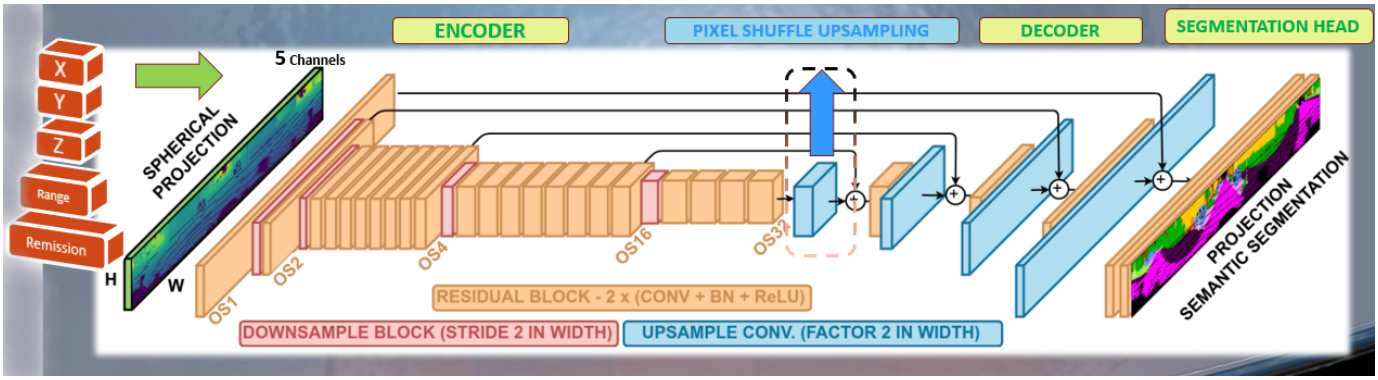

Fig. 3. RangeNet-PS Architecture- Fully 2D convolution architecture.

The figure 2 overviews the proposed method.

- A: We collect raw 3D LiDAR point cloud which is unordered and unstructured and covert to spherical projection then to image coordinates. This representation is explained further below. We input range image to the 2D full CNN architecture.
- B: Here we feed our processed data to the modified Darknet53 architecture and we semantically segment the each range images.
- C: These segmented images are projected back to the original 3D spaces without the loss of original 3D LiDAR point cloud.

A. Range Image Point cloud Representation

Most of the LiDAR sensors input 3D data can be converted to the range image like representation where each column is range value of a single point at a time transmitted by array of laser range-finders and each row is the different turning of the range finder sensor which is fired at constant rate.

When the vehicle is accelerating faster than the sensors rotation, there is geometrically inconsistent in the points cloud for each scan. This phenomenal is called skewing generated by "rolling shutter" behavior. To tackle this problem we usually consider the vehicles motion as the constraint. We just now no longer interested in range value for each pixel but X, Y, Z, and intensity values as well to address the above problem. First step is to convert these de-skewed point cloud into spherical projection and then to images coordinates. Below formulation is used to convert this to image coordinates.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y,x)\,\pi^{-1}\right] & w \\ \left[1 - \left(\arcsin(z\,r^{-1}) + \mathrm{f_{up}}\right)\mathrm{f}^{-1}\right] & h \end{pmatrix}$$

Fig. 4. Spherical co-ordinate representation of 3D LiDAR Point Cloud [7]

Here u, v are the image co-ordinates, $f = f_{up} + f_{down}$ Here f is the vertical field of view. h and w are the height and width of the desired range image representation. $r = \sqrt{x^2 + y^2 + z^2}$ is the range value for each 3D point. From [7] we construct full 360° field of view projection. 3D point coordinates (x,y,z), I and r range index stored as separate RV image channel. we get $[5\,w\,h]$ image that needs to be fed to the Neural network.

B. 2D Semantic Segmentation

We obtain fully 2D semantic segmentation of this range image representation through our RangeNet-PS architecture which is based on RangeNet and contains DarkNet53 blocks of encoder, decoder and finally semantic head block at the end. It is modified to take different input sizes for our comparison and form factor. Entire architecture has total 50,377,364 number of trainable parameters. Each section in architecture is explained more:

1) Encoder: This type of Hour-glass representation has encoder at the top which is characterized to down-sample and collect the contextual information. We change the early block of the encoder to collect more context information of semantics. We added dilated convolution which improves receptive fields. Global context information gathered by receptive fields plays a crucial role in learning complex co-relation between classes or local information. It consist of layer of dilated convolution layers followed by batch normalization and Leaky-ReLU activation. Entire encoder can be divide into number of blocks with different parameters. Each residual block consists of 2 dilated convolution with 1,3,5 as stride factor and output of this block is passed in 2 direction one to next convolution block and another to decoder. This residual blocks are then grouped together with 1,2,8,8,4 downsampling rate and added with droupout layer.

2) Decoder: The decoder consist of pixel shuffle layer which is modified in the architecture instead of transpose convolution. It doses the upsampling like deconvolution with less parameters causing less number of trainable parameters. It is followed by deconvolution layer and residual block with dropouts.

This network contains skip-connection that connect from the encoder to its corresponding layer in decoder block followed by dropout layer. This help in recovering high frequency edge information which is especially lost during our down-sampling process.

3) Segmentation Head: After encoding and decoding layers we have left with the new segmentation head block which

has set of [1 x 1] convolution generating output of n classes [n x h x w] logits. This is followed by Softmax activation which assigns each pixels to different classes. The logits are unbounded output to its corresponding classes. The loss function we use is cross-entropy loss using stochastic gradient descent as optimization.

$$\mathcal{L} = -\sum_{c=1}^{C} w_c\, y_c \log\left(\hat{y}_c\right), \text{ where } w_c = \frac{1}{\log\left(f_c + \epsilon\right)}$$

Fig. 5. Weighted cross-entropy loss [7]

Here $\mathcal{L}$ is weighted cross entropy loss with class c and it penalize with inverse f_c handling imbalances of data. example in Semantic segmentation, the dataset contains most of the "road" class followed by "cars" and "pedestrians" class respectively.

C. Point Cloud Reconstruction from Range Image

Now we need to reconstruct back our original 3D points from the 2D semantic labelled outputs. Normally we use range information along with the pixel coordinates and sensor intrinsic information to map from $\prod^{*} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. However since we used non liner mapping in part **A** which converted from spherical representation to image coordinates, there was loss of 3D points or we can say each pixel contained 1 or more 3D points(neighbors) information. This could mean there was significant drop in the number of points. This is usually critical if we take height and width [64 x 1024] and below. Example one scan with 130,000 points projected to [64 x 512] just contains 32,768 points represented sampling closest point in pixels frustum. To infer all the points, we use (u,v) pairs obtained during initial rendering and index the range image with the image co-ordinates that corresponds to each point.

IV. Experimental Evaluation

We run our experiments on Semantic Kitti dataset. Our experimental evaluation supports two claims 1) There is improvement in class accuracy from original RangeNet. 2) There has been significant improvement in the runtime.

A. Dataset

Semantic-Kitti is vast dataset with 21 sequences of 3D LiDAR point clouds. We train and evaluate with this proposed method that provide point wise annotation for entire Kitti Odometry Benchmark. The dataset contains 43000 scans over 21 sequences, But 21000 scans from 00-10 are available for training and validation. Sequence 11-21 are not public and are generally used for testing. Due to time constraint, we have used sequence 00 for Testing, 08-09 sequence as Validation and 01-07 sequence for testing. We have used total 19 classes for training and evaluation.

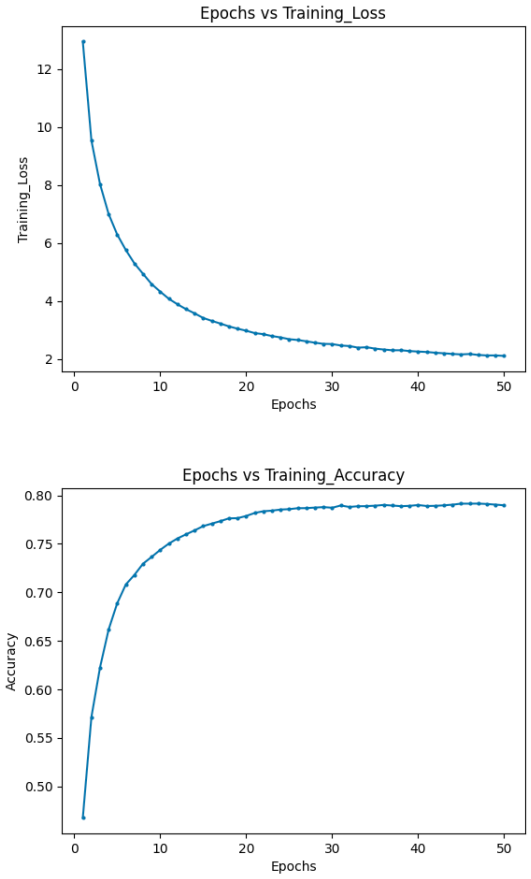

Fig. 6. Training Plots. **a.** Training Loss Vs Epochs **b.** Training Accuracy Vs Epochs

B. Metrics

To evaluate our labelling performance of our dense prediction task of semantic segmentation where we predict class for each 3D LiDAR point cloud. We evaluated with metrics like mIoU (Mean Intersection over Union), IOUs for all classes, Precision and Recall.
Intersection over Union for all classes are given in Fig 3:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}$$

Fig. 7. *mIoU* over all the classes.

Here TP, FP, FN are True positive, False positive and False negative for class c over all the classes C. We also evaluated on Precision and recall where Precision measures the fraction of correctly predicted positive pixels out of all pixels that the model has predicted as positive. High Precision value indicates that the model is accurate over the classes and is not prone to false positive. Whereas Recall measures the fraction of correctly predicted positive pixels out of all pixels that truly belong to object class. A high recall value indicates that the

TABLE I
IOU [%] ON TEST SEQUENCE 00. COMPARISION BETWEEN BASELINE ARCHITECTURE RANGENET AND OUR IMPLEMENTATION RANGENET-PS ARCHITECTURE.

Architecture	Size	Accuracy	MeanIoU	Car	Motorcycle	Other-vehicle	Motorcyclist	Road	Building	Fence	Terrain	Traffic-sign
RangeNet	64 x 1024px	69.75	50.36	**85.80**	**12.35**	10.36	3.57	**95.84**	70.10	25.31	**70.94**	**21.69**
RangeNet-PS	64 x 1024px	62.67	**50.87**	83.53	6.91	**12.40**	**4.6**	89.81	**70.98**	**26.84**	59.85	15.43

model is accurate in identifying all the pixels that belongs to object class.

C. Performance of Modified RangeNet w.r.t orignal RangeNet

We evaluate our RangeNet-PS performance w.r.t original RangeNet architecture in Table 1. We can see that by adding Pixel shuffle layer in the decoder has improved its runtime where we evaluate runtime in table 2. From Table 1, we see the overall accuracy of base RangeNet is slightly less than ours RangeNet-PS , but our approach excels in mean IOUs and overall class IOUs. Some classes like "bicycle" , "truck", "other-ground", "motorcycle" has not been identified accurately due to less training sequence and scarcity of these classes in overall Semantic-Kitti sequences. We hope to improve this in our future work with addition of post processing with GPU based KNN for better recovery of original 3D point cloud with accurate labels.

D. Runtime

We evaluated runtime of both the approaches in table 2. Hardware we used were High Computing performance "NVIDIA's Tesla T4 single GPU" with 14.75GB memory and 7.5 as compute capability.We have kept Batch size as 2 and Number of epochs as 50 for entire comparison.

1) Pixel Shuffle layer: We see by adding pixel shuffle layer in the encoder has significantly improved computationally expensive operation of up-sampling. This has reduced number of trainable parameters significantly as compared to transpose convolution. It leverages from learnt feature maps to produce upsampled feature maps by shuffling the pixels from channel dimension to spatial dimension. The pixel shuffle layer reshapes from $(Cr^2$ x W x $H)$ feature map to $(C$ x Wr x $Hr)$ where H, C, W, r are height, channel, width and upsampling factor.

After Pixels shuffle in the decoder followed by dropout layer, the channels dimension did not matched with the input the next decoder block of the original RangeNet, so we perform 1x1 convolution layer to match its dimension. Thus we see that after introduction of Pixel layer the runtime has improved significantly from original RangeNet architecture.

TABLE II
COMPARISION ON RUNTIME BETWEEN BASELINE RANGENET AND OUR IMPLEMENTATION RANGENET-PS ARCHITECTURE

Architecture	Hardware	Resolution	Processing Time (ms)
RangeNet	NVIDIA's Tesla T4	64 x 1024px	74
RangeNet-PS	NVIDIA's Tesla T4	64 x 1024px	43

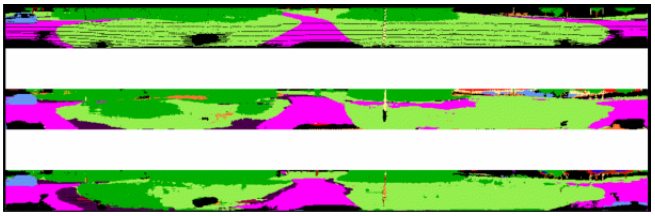

Fig. 8. Segmented Output: Ground truth (top), RangeNet-PS (Middle), Orignal RangeNet (Bottom)

V. RESULTS

Complete detailed quantitative analysis is performed on Semantic-Kitti LiDAR point cloud dataset. Table 1 shows comparison between RangeNet and modified RangeNet (RangeNet-PS) architecture. Evaluation metrics like MeanIoU, Classes IoU, Accuracy, Precision and Recall were used for comparison.

Implementation of dilated convolution stack to improve receptive fields and thus the context between Global and local information and its co-relation between classes has improved its meanIoU but has slightly decreased its accuracy. Decrease in the accuracy can be due to lack of proper training and may require more training up to to 150 epochs as given in the RangeNet++ [7]. Some classes like "Bicycle", "Truck", "Pedestrians", "other-grounds" has not been identified accurately due to less training sequence, few and smaller instances compared to other classes with very fine details in the Semantic-Kitti dataset.

Pixel shuffle layer in decoder has significantly decreased number of training parameters and thus we can see in Table 2, there is significant drop in runtime for RangeNet-PS architecture. This approach was tried and tested in SalsaNext [8] and achieved excellent results.

VI. CONCLUSION AND FUTURE WORK

We achieved goal of this proposed implementation to achieve faster and accurate semantic segmentation on 3D LiDAR Point cloud than the Velodyne HDL-64E sensor for Autonomous Driving.

In Future we could post process with GPU based KNN technique which can improve reconstruction of semantics for entire labelled point cloud even for smaller range image resolution. This will help to improve border-IoU for different distances. This proven technique not only improves IoU by some % but significantly improves border IoU score for low values of distances to border parameter. This also eliminates problem like "bleeding" and "shadowing" as explained in [7].

REFERENCES

[1] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[2] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[5] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2530–2539.

[6] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4376–4382.

[7] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 4213–4220.

[8] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*. Springer, 2020, pp. 207–222.