A Brief Survey on 3D Semantic Segmentation of Lidar Point Cloud with Deep Learning

Authors
name of organization (of Aff.)
Email address

2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES) | 978-1-6654-2157-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/NILES53778.2021.9600493

***Abstract*—3D semantic segmentation is a fundamental task for many applications like Autonomous Driving. Recent work shows the capability of Deep Neural Networks in labelling 3D point clouds of major sensors like: LiDAR and Radar. The main challenge that faces this task is the nature of 3D point clouds being unordered and spatially-uncorrelated, making it different in terms of processing algorithms than the images. In addition to that, a point cloud usually needs higher processing power than the images if it's processed in its raw nature. In this paper, we will review different deep learning methods for 3D semantic segmentation, examples of the widely used datasets in addition to the evaluation metrics.**

***Index Terms*—Deep Learning, 3D Semantic Segmentation, Lidar point cloud, Autonomous Driving**

I. Introduction

The major role of image Semantic Segmentation is to assign a class label to each pixel, which can be used as a scene understanding task in applications like autonomous driving perception. similarly, 3D Semantic Segmentation of Lidar point cloud is to assign each point with a class label, which is a challenging task in computer vision due to the nature of point clouds being unordered and spatially-uncorrelated, in addition to high computationally power usually needed to process point cloud in its raw nature.

In this paper, we introduce a brief survey of the important state-of-the-art methods for 3D semantic segmentation, also categorised them into two main categories: Projection-based Semantic Segmentation which is concerned by the methods that transform the representation of the 3D nature of point clouds to 2D grid image to make it easy to apply different Convolutional Neural Network (CNN) architectures and other deep learning techniques that can automatically learn good features from images [1]. Fig.1 is an example of the expected output of 3D semantic segmentation, it's a scene of lidar point cloud with each point labelled with a class label like: vehicle, road, pedestrians, trees, .. etc.

In a nutshell, the projection-based methods for 3D semantic segmentation usually takes less inference time and has simpler network architecture than point-based methods, on the other hand, it suffers from two major issues: 1) The discretization error when needs to recover the 3D point cloud points from 2D image grid. 2) The hardness of detecting over-driving classes like bridges and tunnels.

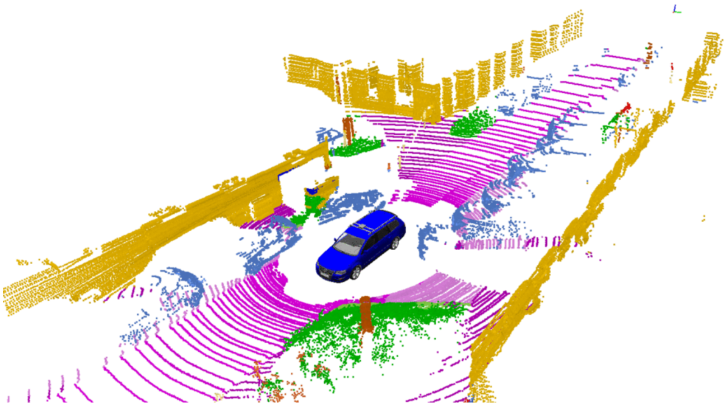

Fig. 1. Example of 3D Semantic Segmentation scene. The figure is borrowed from [2]

II. State-of-the-art methods

In this section, we will explore some methods for the two main categories of 3D pointcloud semantic segmentation, Projection-based and Point-based semantic segmentation. Table I summarizes some of leaderboard of some published methods on a popular dataset Semantic-KITTI [2] for both categories sorted by the mIoU percentage [III-B1] and source code link if available. Figure 3, shows mean IoU vs. reported run time in msec for some Networks on Semantic-KITTI dataset.

A. Projection-based Semantic Segmentation

Projection-based approaches are aiming to change the representation of the 3D point cloud to be a 2D image grid which can be processed by networks with same ideas as Image-based Semantic Segmentation mentioned in the previous section. One of the first networks to follow this paradigm is SqueezeSeg [14], it's network architecture (Fig.2) is derived from SqueezeNet [15], a lightweight CNN that achieved AlexNet [16] level accuracy with 50X fewer parameters, it projects the Lidar point cloud onto a spherical grid-based representation to transform sparse, invariant order of 3D point clouds to 2D grid, the spherical coordinate is preferred than Cartesian coordinate, as the way Lidar sensor work is scanning the environment in a spherical-like way. SqueezeSeq architecture also contains Conditional random fields (CRF) as recurrent neural networks which can refine the output point-wise classification.

TABLE I
LEADERBOARD OF SOME PUBLISHED METHODS ON SEMANTIC-KITTI

Method	Year	Source code (github.com/)	mIoU%	Contribution
AF2S3Net [3]	2021	N/A	69.7	Multi-branch Attentive Feature Fusion in the encoder and Adaptive Feature Selection with feature map re-weighting in the decoder
Cylinder3D [4]	2020	xinge008/Cylinder3D	67.48	Cylinder partition and asymmetrical 3D convolution networks.
SPVNAS [5]	2020	mit-han-lab/spvnas	66.4	3D module boosts the performance on small objects, AutoML framework for 3D scene understanding, model size reduction and computation reduction
FusionNet [6]	2020	N/A	61.3	Avoid ambiguous/wrong predictions when a voxel has points from different classes and Effective feature aggregation operations
SalsaNext [7]	2020	Halmstad-University/SalsaNext	59.5	Context module before encoder consists of a residual dilated convolution stack fusing receptive fields at various scales.
SqueezeSegV3 [8]	2020	chenfengxu714/SqueezeSegV3	55.9	Spatially-Adaptive Convolution (SAC) to adopt CNN filters for different locations according to the input image.
MPF [9]	2021	N/A	55.5	Processing spherical and bird's-eye view projections using two separate 2D fully convolutional Networks then fuses the segmentation results of both views.
PolarNet [10]	2020	edwardzhou130/PolarSeg	54.3	Polar bird's-eye-view representation for points grouping.
RangeNet++ [11]	2020	PRBonn/lidar-bonnetal	54.3	Pipeline: Transformation of point cloud into a range image representation, 2D FCN applied, reconstruct from 2D to 3D then 3D post-processing to clean the point cloud from undesired discretization.
PointNet++ [12]	2017	charlesq34/pointnet2	20.1	Solved the problem of PointNet of capturing the local features via introducing a hierarchical neural network that applies PointNet recursively on a nested grouping of the input point cloud.
PointNet [13]	2016	charlesq34/pointnet	14.6	A novel architecture of a neural network that directly consumes point clouds and well respects the permutation invariance of points in the input.

There are two other versions of SqueezeSeg, SqeezeSegV2 [17] which proposed the Context Aggregation Module (CAM) to increase the network robustness to dropout noise in addition to introducing a domain adaptation way to make use of the simulator - such as GTA-V - that can be used to collect an unlimited amount of data with annotations and SqueezeSegV3 [8] which introduce Spatially-Adaptive Convolution (SAC) to adopt CNN filters for different locations according to the input image.

A similar approach is RangeNet++ [11] which transform the input point cloud into a range image representation, then apply it to a 2D Fully Convolutional Neural Netork, then recover the 3D point cloud structure from the 2D segmentation and finally uses KNN to refine the output from the discretization effect.

On the other hand, Multi-Projection Fusion (MPF) [9] processing spherical and bird's-eye view projections using two separate 2D fully convolutional Networks then fuses the segmentation results of both views, its results achieved a mIoU of 55.5 which is higher and faster by 1.6x than RangeNet++.

B. Point-based Semantic Segmentation

Point-based approaches are aiming to capture the point cloud scene features - to be used in classification or semantic segmentation - directly from the point cloud raw data without converting it to another domain like projecting it

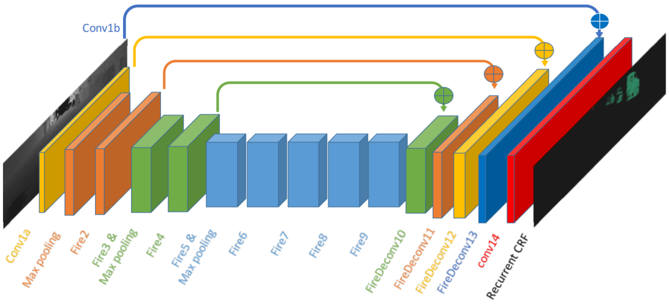

Fig. 2. The architecture of SqueezeSegV1, The figure is borrowed from [14]

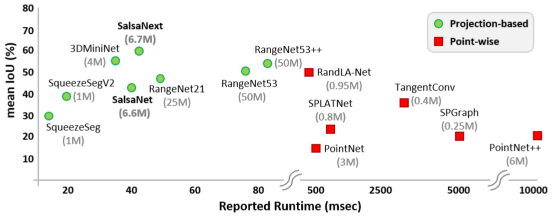

Fig. 3. Mean IoU vs. Reported run time in msec for some 3D point cloud semantic segmentation Deep Neural Networks on Semantic-KITTI dataset [2]. Inside parentheses are the total number of neural network parameters in Millions. The figure is borrowed from [7]

to an image. Previous approaches pipeline mainly consist of Removing the ground, clustering the remaining points into instances, extracting (hand-crafted) features from each cluster, and classifying each cluster based on its features [18]. This approach can be efficient in terms of computational cost. However, it's not efficient in terms of accuracy of point-wise classification as the ground removal algorithms may fail to generalize, this pipeline approach aggregates compound errors and the clustering algorithms can't capture the context of the environment. The other approach is to use end-to-end machine/deep learning architectures pioneered by PointNet [13] that could end-to-end capture the features of point cloud via applying the (nx3) point cloud - n is the number of points - as input directly to series of Multi-Layer Perceptron (MLP), then a max-pooling layer - which respects the permutation invariance of points in the input - aggregate the information from all the point into a feature vector representing the global features in the input, then upsampling this global feature vector by concatenating it with an early-stage feature vector to be applied to another series of MLPs to form the output of (nxm) dimension representing the semantic segmentation, which m is the number of classes.

The successive version of PointNet is PointNet++ [12] which solved the problem of PointNet of capturing the local features via introducing a hierarchical neural network that applies PointNet recursively on a nested grouping of the input point cloud. Fig.4 show the architecture of PoitNet++ where sampling and grouping here is trying to mimic the kernels in image based Convolution Neural Networks, which are responsible to capture the spatial features in images.

A similar approach to PointNet++ is the PointCNN [19] a generalization of CNN into leveraging spatially-local correlation from a point cloud. The main contribution is the X-Conv operator that weights and permutes input points and features before they are processed by a standard convolution layer. Also, Pointwise Convolutional Neural Networks [20] architecture is based on a Pointwise convolutional layer, where the kernel is a grid (ex: 3x3) centered at each point and the neighbour points can contribute to the center point.

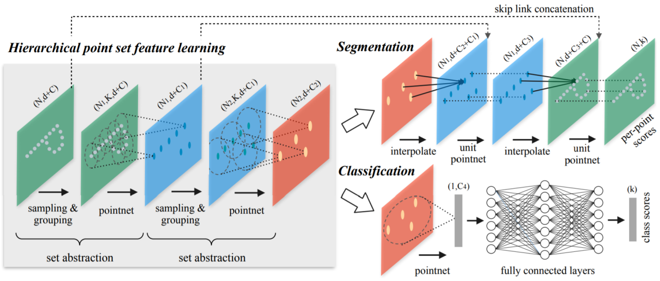

Fig. 4. The architecture of PointNet++, The figure is borrowed from [12]

III. Datasets and Evaluation Metrics

A. Datasets

1) Semantic-KITTI [2] : It's a lrage scale dataset that actually depend on another prestigious dataset (KITTI [21]), it contains the same KITTI Odometry Benchmark Velodyne point clouds in additional to annotation for semantic scene understanding, like semantic segmentation and semantic scene completion. Contains 23201/20351 of scans for train/test set, 28 annotated classes Fig. 5

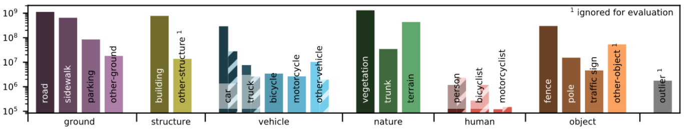

Fig. 5. Label distribution. The number of labelled points per class and the root categories for the classes are shown. For movable classes, the number of points showed on non-moving (solid bars) and moving objects (hatched bars), The figure is borrowed from [2]

2) Stanford 3D Indoor Scene Dataset (S3DIS) [22] : It's an indoor dataset collected in 6 large-scale indoor areas, contains a total of 70,496 regular RGB, along with their corresponding surface normals, depths, semantic annotations of d 3D meshes and point clouds, global XYZ and camera metadata Fig.6.

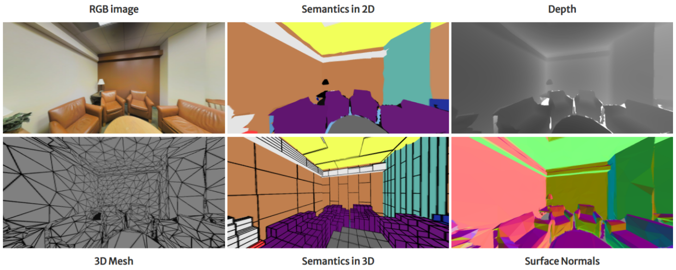

Fig. 6. The supported annotations at S3DIS dataset .The figure is borrowed from [22]

3) nuScenes [23]: It collect scenes usng the full autonomous vehicle sensor suite: 6 cameras, 5 radars and 1 lidar, all with full 360 degree field of view, comprises 1000 scenes, each 20s long and fully annotated with 3D bounding boxes for 23 classes.

B. Evaluation metrics

1) Intersection over Union (IoU): IoU here for point cloud is different from the normal IoU used for 2D images semantic segmentation [24] due to the nature of spatially-uncorrelation in 3D point cloud. It is calculated using the correctly classified points, the incorrectly classified points, and the misclassified points. IoU per class is calculated based on the following equation:

$$IoUClass(i) = \frac{TP}{TP + FN + FP} \tag{1}$$

where TP is the total number of true positive points which are correctly predicted as $class(i)$, FN is the total number of false negative points which belong to $class(i)$, but they are predicted as another classes, and FP is the total number of false positive points which are predicted as $class(i)$, but they belong to another classes. The mean IoU (mIoU) is simply the mean across all classes.

2) Overall Accuracy (OA): The measurement overall accuracy can be calculated as follows:

$$OA(i) = \frac{CorrectlyClassifiedPoints(Class(i))}{TotalPointsNumber} \quad (2)$$

The overall accuracy can be misleading when the used data set has a significant disparity between classes.

Conclusion

In this paper, we briefly reviewed some methods of deep learning based 3D semantic segmentation from different perspective, projection based and point wise based networks. In general, for applications that needs real-time inference processing, it's recommended to use the projected based approaches which in most cases has lower inference time than point wise approaches. However, on the other hand the recent state-of-the-art in terms of the accuracy are point-wise approaches.

References

[1] Y. B. Yann LeCun and G. Hinton, "Deep learning," *Nature 521, 436–444*, 2015.

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[3] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "(af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," *ArXiv*, vol. abs/2102.04530, 2021.

[4] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," *arXiv preprint arXiv:2011.10033*, 2020.

[5] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020.

[6] F. Zhang, J. Fang, B. Wah, and P. Torr, "Deep fusionnet for point cloud semantic segmentation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 644–663.

[7] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving," 2020.

[8] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 1–19.

[9] Y. Alnaggar, M. Afifi, K. Amer, and M. Elhelw, "Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds," 01 2021, pp. 1799–1808.

[10] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[12] C. R. Q. L. Y. Hao and S. L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space supplementary material."

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.

[14] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," *arXiv preprint arXiv:1710.07368*, 2017.

[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv:1602.07360*, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[17] 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 *ICRA*, 2019.

[18] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5067–5073.

[19] M. S. Y. Li, R. Bu and B. Chen, "Pointcnn: Convolution on x-transformed points," *arXiv:1801.07791*, 2018.

[20] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[22] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, Feb. 2017.

[23] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[24] R. L. Barkau, "Unet: One-dimensional unsteady flow through a full network of open channels. user's manual," Hydrologic Engineering Center Davis CA, Tech. Rep., 1996.