# Semantic Segmentation of 3D LiDAR Data at Dynamic Urban Scenes

Biao Gao*, Jilin Mei*, Donghao Xu*, Xijun Zhao†, Wen Yao†, Huijing Zhao*
*Peking University, Beijing, China
†China North Vehicle Research Institute, Beijing, China

***Abstract*— This work studies semantic segmentation of 3D LiDAR data at dynamic urban scenes. LiDAR data plays an important role of perception in autonomous driving system. However, most semantic segmentation methods and datasets are designed for camera data nowadays. In this work, we propose a method which can generate semantic segmentation of LiDAR data and we evaluate its performance on a new 3D point cloud dataset collected in dynamic urban scenes by our driving platform. The experiments show that our method can recognize more kinds of labels and achieve an impressive result in dynamic urban scenes.**

## I. Introduction

Scene understanding is crucial for the safe and efficient navigation of autonomous vehicles in complex and dynamic environments, and semantic segmentation is a key technique. 3D LiDAR has been used as one of the main sensors in many prototyping systems for fully autonomous driving[1]. Semantic segmentation using 3D LiDAR data is illustrated in Fig. 1, where given a frame of input data (a), the problem is to find a meaningful label (i.e., object class in this research) for each pixel, super-pixel or region of the data (b). As 3D LiDAR is a 2.5D sensing of the surroundings, it can be represented equivalently in the form of a range image (c)-(d) in the polar coordinate system, and the problem of semantic segmentation can be solved by using either 3D points or range images as the input.

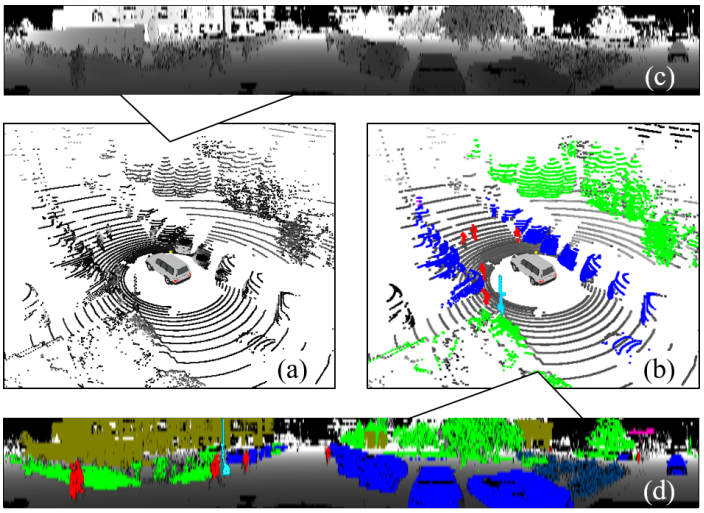



Fig. 1: The semantic segmentation for dynamic scene. (a) and (c) show the input data in two kinds of formats, i.e., the raw 3D point clouds and range frame. (b) and (d) show the semantic segmentation results.

Semantic segmentation using 3D LiDAR data from outdoor scenes has been studied since the past decade [1], [2], [3]. The traditional process in these works [4], [5] includes the following steps: (1) preprocessing to divide a whole dataset into locally consistent small units, such as voxels, segments or clusters; (2) extracting a sequence of predefined features; (3) learning a classifier via SVM, random forest etc.; and (4) refining the results using a method such as conditional random field by considering the spatial consistency among neighboring units. On one hand, the criteria of units in preprocessing are empirically defined, for example, the size of voxels[6]; one the other hand, the traditional methods depend on carefully designed discriminative features, and their adaptability to different scenes remains an open challenge.

The recent success of deep learning in image semantic segmentation has provided new approaches[7]. These methods remove the dependence on handcrafted features in an end-to-end manner. Especially, the propose of FCN-based methods[8] where the semantic segmentation is converted into a pixel-wise classification to reduce the dependence of unit's definition in preprocessing. However, these methods also have substantial demands for finely labeled data[7], and few 3D LiDAR datasets with annotation at the point level to support autonomous driving applications are publicly available. Therefore, the attempts of a point-wise segmentation method for 3D LiDAR data and a fine annotation dataset are necessary.

In this paper, we propose a new 3D LiDAR data semantic segmentation method that takes the range frame as input and make a point-wise classification via deep neural network. We evaluate its performance on a new 3D point cloud dataset collected in dynamic urban scenes by our driving platform. The experiments show that our method can recognize more kinds of labels and achieve an impressive result in dynamic urban scenes.

The remainder of this paper is organized as follows. Related work is discussed in Sect. II. In Sect. III, the proposed method is presented. Then, we present the experimental results in Sect. IV and draw conclusions in Sect. V.

## II. Related works

### A. Feature-based Methods

Feature-based methods belong to traditional machine learning, and the general process of these methods consists of feature selection, classifier design and graphical model description.

A straightforward technique is to convert semantic segmentation into a point-wise classification that includes extracting the features on each unit, concatenating the features as a vector and determining the label via a well-trained classifier. [9] presents a common pipeline from feature selection to classifier training. Due to the irregular arrangement of point clouds, the authors test 5 definitions of neighborhood to achieve the best representation. Similar research is conducted in [10], which demonstrates the ability to address varying densities of data. A single point cloud usually contains millions of points, so evaluating the label for each point is typically computationally expensive (on the order of minutes according to [10]).

[11] proposes an efficient approach where speed and accuracy are satisfied simultaneously; furthermore, the average classification time can be reduced to less than 1 s. [5] represents the raw point cloud as a 2D range image and proposes a framework for simultaneous segmentation and classification of the range image that considers both the 2D range image and 3D raw data. Straightforward approaches assume that each data unit is independent and ignore the spatial and contextual relations between units. Consequently, they can produce good results based on distinctive features. However, when the features are not discriminative, the point-wise classification will be noisy and locally inconsistent[4].

The neighbor elements are taken into account to make the segmentation results spatially smooth. For this purpose, graphical models such as Markov random Field (MRF) and conditional random field (CRF) are usually exploited to encode the spatial relationships. In [12], the node potentials and edge potentials are both formulated with a parametric linear model, and the functional max-margin learning is used to find the optimal weights. [13] proposes a simplified Markov network to infer the contextual relations between points. Instead of learning all the weights for the node and edge potentials in graphical models, the node potentials are calculated from a point-wise classifier, and the edge potentials are determined by the physical distance between points.

The performance of the above methods largely depends on handcrafted features. These methods are effective in fixed or regular scenarios, but for dynamic scenes, the features are empirically designed and the performance decreases.

### B. Deep Learning Methods

Deep learning, especially the convolution neural network (CNN) without handcrafted features, has shown effectual performance on 2D image segmentation[7]. However, the semantic segmentation of 3D point clouds(i.e., from LiDAR sensors) is still an open research problem[14] due to the irregular, not grid-aligned properties. Therefore, recent studies project the point clouds into 2D views, and some of them attempt to directly ways, for example, volumetric/voxel representations.

Inspired by the success of CNN in image segmentation, the state-of-the-art image-based algorithms can be used directly after rendering 2D views from the 3D raw data. [15] projects point clouds into virtual 2D RGB images via Katz projection. Then, a pretrained CNN is used to semantically classify the images. However, this projection removes all the points that are not visible; for example, if a car is projected, all the points behind it are removed. [16] unwraps $360^\circ$ 3D LiDAR data onto a spherical 2D plane without point loss. Spherical projection is also applied in SqueezeSeg[17], where the CNN directly outputs the point-wise label of the transformed LiDAR data and a CRF is applied to refine the outputs. [18] uses cylindrical projection to create the depth and reflectivity images. In [19], the point clouds are encoded by top-view images and a simple fully convolutional neural network (FCN) is used. This method can be used in real time because only elevation and density features are extracted. In [20], the input point cloud is projected into multiple views, such as color, depth and surface normal images.

Another type of method models the raw data in direct ways. [21] proposes SEGCloud, where the raw 3D point cloud is preprocessed into a voxelized point cloud with a fixed grid size. Although [21] is simple and effective, how to set the voxel size is a problem in large-scale scenes. Thus, the scene is voxelized at five different resolutions in [6], and each of the five scales is handled separately by the CNN. Rather than using a fixed grid size, [22] proposes OctNet, where the hybrid grid-octree data structure is applied to represent the raw 3D data, and each leaf of the octree stores a pooled feature representation. PointNet[23] is a unified architecture that directly takes raw point clouds as input and outputs the label of each point. The scene is divided into blocks. Then, the points in each block are passed through a series of multilayer perceptrons (MLPs) to extract the local and global features. Based on [23], [14] extends the method to incorporate a larger-scale spatial context, and improved results are reported in both indoor and outdoor scenarios. [24] proposes a more elegant architecture to capture contextual relations. The first step is to partition the raw point cloud into geometrically simple shapes, called super-points. The super-points are then embedded by PointNet[23].

Semantic segmentation with deep learning is usually implemented in a supervised manner, which requires detailed annotations. However, obtaining point-wise annotations for 3D point clouds is labor intensive and time consuming. Furthermore, few public datasets support this level of annotation. Our method belongs the former brand where the the raw 3D point clouds is converted into 2D range frame, and FCN is applied .....

## III. Methodology

### A. Data Preprocessing

The point cloud data for LiDAR is sparse and unorganized, so it is time-consuming to find neighboring relations between different points. In order to process these unorganized point

cloud data with deep convolutional neural network, we convert the point cloud data into 2D range image by cylindrical projection. After that, it will be easier to implement deep convolution neural network on the LiDAR data.

After cylindrical projection, the point cloud data will be encoded as a dense matrix with shape of $[H, W, C]$. $H$ means the number of lines for the specific LiDAR sensor (such as $H = 32$ for Velodyne HDL-32E). $W$ equals to the number of points within each LiDAR scan line. $C$ is the channels' number in the range image. Here, $C$ is set to 3, which represents $[Range, Intensity, Height]$ three channels.

For each point $p_k = \langle x_k, y_k, z_k \rangle$ from the raw point cloud set $P$, the value of $Range$ in range image $R$ is defined as $r_k$:

$$r_k = \sqrt{{x_k}^2 + {y_k}^2 + {z_k}^2}, r_k \subset [0, 255] \tag{1}$$

Similarly, the values of $Intensity$ and $Height$ are normalized into $[0, 255]$. These channels are very important properties of LiDAR data, which are enough to describe various objects in dynamic urban scenes.

### B. Problem Definition

Let $X$ denotes the range image extracted by cylindrical projection of 3D point cloud data $S$. In this kind of projection, there is a one-to-one correspondence between a 3D point in one frame point cloud data and a pixel in the range image $X$. As a result, the semantic segmentation task of 3D point cloud data is equal with giving each pixel $x$ in the range image $X$ a label $y$. The problem of this work is formulated as learning a semantic segmentation model $f_\theta$ which maps each pixel $x$ to a label $y \in \{1, ..., K\}$, and subsequently associate $y$ to the 3D points of $S$.

$$f_\theta : x \to y \in \{1, ..., K\} \tag{2}$$

The data samples are in the form of range images $X$. Given a set of supervised data samples $X_l = \{x_i, y_i\}$, where $\{x_i\}$ traverses each pixel of $X$ and $\{y_i\}$ are labels for $\{x_i\}$, annotated manually by human annotators. In order to learning a semantic segmentation model $f_\theta$, we need to find the best parameter set $\theta^*$ that minimize a loss function $L$ as below.

$$\theta^* = \arg\max_\theta L(X_l; \theta) \tag{3}$$

### C. Network Architecture and Loss Function

We use a FCN (Fully Convolutional Network) architecture for this semantic segmentation task. Compared with common deep convolutional networks, it removes last fully connected layers, and replaces them with the in-network up-sampled or deconvolutional predictions of convolutional layers as predicted feature maps. During training procedure, it generally computes cross-entropy like losses in pixel-wise, between the predicted labels and ground truths.

Our network is trained via end-to-end guided by the designed loss function. Because the number of pixels are imbalanced between different classes and a lot of invalid or unknown-class pixels, the following multi-class weighted cross entropy loss function is designed to regular the weights between imbalanced classes.

For labeled data $X_l$, we implement some changes on the widely-used definition of cross entropy, and define loss function $L_l$ as below:

$$\Gamma_{i,j} = \begin{cases} \overrightarrow{\varphi_k}, & if \quad [y_{i,j} \neq k \quad and \quad y_{i,j} \neq 0] \\ \overrightarrow{0}, & otherwise \end{cases}$$

$$L_l(X_l, Y_l; \theta) = -\frac{1}{H * W} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \sum_{k=0}^{K} \Gamma_{i,j} \omega_k ln(P_\theta^k(x_{i,j})) \tag{4}$$

Where $\Gamma_{i,j}$ is a one-hot vector $\varphi_k$ of label $k$, if $y_{i,j} \neq k$ and $y_{i,j} \neq 0$. Label 0 means invalid or unknown pixels, including many fine fragments belong to background or hard to be annotated, so we don't want to evaluate these pixels if they are predicted as non-zero labels. $\omega_k$ here is used to balance the sample numbers between different labels and $P_\theta^k(x_{i,j})$ is the probability that pixel $x_{i,j}$ be assigned a label $k$ by our semantic segmentation model with the set of parameters $\theta$.

## IV. Experiment

### A. Data Set

The performance of the proposed method is evaluated on a dynamic campus data set collected by an instrumented vehicle, which has a GPS/IMU suite and a Velodyne-HDL32, as shown in Fig. 2. The total route contains 1375 LiDAR frames. 880 frames for training, 220 frames for validation and 275 frames for testing.

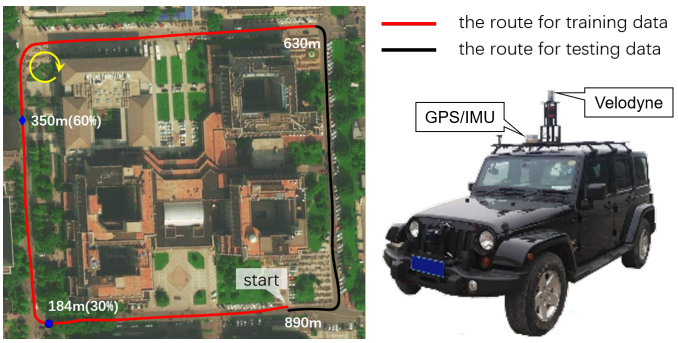



Fig. 2: The routes of data collection and the platform configuration.

High quality pixel-level annotation is necessary for network training. Instead of working on the raw point cloud, human annotators work on the range image where object regions are associated with the ones in adjacent frames. Annotators only need to assign the category of some region in one frame, then a series of associated regions are marked with the same label. Although sometimes the data association brings errors, it largely reduce the annotation time.

TABLE I: Categories Distribution in Dataset

| - | Pedestrian | Car | Vegetation | Sign/Pole | Building | Cyclist | Bicycle | Road |
|---|---|---|---|---|---|---|---|---|
| Pixels | 458,673 | 257,239 | 4,661,880 | 80,384 | 6,579,360 | 274,991 | 1,356,518 | 16,856,614 |

The categories distribution in this dataset is shown in TABLE . I. Obviously, the data distribution is imbalanced between categories. So, we apply each label an unique weight based on data distribution to reduce the influence of data imbalance.

### *B. Setup*

Our method is implemented with a FCN (Fully Convolutional Network). The range image size is 1080x32, which width is down-sampled for efficiency. A small batch size for training sets will be better. The network is implemented with TensorFlow in the environment with NVIDIA TITAN X GPU. We use the AdamOptimizer with 1e-5 learning rate.

It's important to aware that our data frames are captured sequentially. In order to avoid data correlation between adjacent frames, shuffling them before training process is necessary.

### *C. Results Evaluation*

We use mPA (mean pixel accuracy) for quantitative evaluation of semantic segmentation performance. Specially, when calculating the mean pixel accuracy, unknown or invalid pixels with label 0 will not be involved in. The mPA is computed as the following formula, while $\dot{Y}_k$ is the predicted pixel set with label $k$ and $Y_k$ is the ground truth pixel set.

$$mPA = \frac{1}{K} * \sum_{k=1}^{K} \frac{\left|\dot{Y}_k \cap Y_k\right|}{\left|\dot{Y}_k\right|} \tag{5}$$

As shown in Table. II , we list the semantic segmentation performance (mPA) of each label, and use the result trained with original cross-entropy as baseline for comparison. Obviously, most categories achieve higher mPA performance after use our proposed method with weighted loss function. Especially, the *pedestrian* label gets 37% accuracy increase, and not only the category with fewer samples gets higher accuracy, but also the category with large amounts of samples gets higher accuracy, such as the *vegetation*. However, some categories like *sign/pole* still don't have ideal pixel accuracy by our method. This is most likely for too few pixel samples of these categories, so the deep learning model can't fit the data well.

The qualitative semantic segmentation results are shown in Fig.3. Dynamic urban scene is very vivid and contains abundant categories objects and details, which is different from simple environment on highway. As we can see in Fig.3(a), the weighted loss leads to better performance of labels with less samples, such as the traffic sign in Box A. Similarly, Box B,C,D in Fig.3(b) also give some improved examples.

Fig.4 shows the confusion matrix of baseline and weighted-loss model. It's easy to find that our model achieves a great improvement of most categories. However, some confusion between categories still exist, such as many *sign/pole* samples are predicted as *vegetation* and some *cyclist* samples are predicted as *pedestrian*. The second situation is mainly caused by the cyclists close to our vehicle, so only the upper part of the body can be seen.

By the way, there is an interesting explanation for low pixel accuracy of label *sign/pole*. As shown in Fig.3(b), we can find that some trees trunks are labeled by *sign/pole*. This is mainly because of the high reflective moth-proofing paint on the trunks, which has high value of intensity and shape as a pole, just like the traffic sign. In order to deal with this situation, maybe object-level information needs to be imported.

## V. Conclusion

This paper proposes a semantic segmentation method of 3D LiDAR data at dynamic urban scenes. We convert the LiDAR data into the form of range images and use a fully convolutional network to predict the semantic segmentation result. We use our new dataset for evaluation, and find the weighted loss distinctly improves the pixel accuracy performance.

There are still much work need to be done in the future. In order to further improve the semantic segmentation performance, object-level information, spacial and temporal constraints between samples will be addressed on.

## References


[1] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[2] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 215–220.

[3] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3d lidar point clouds," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2798–2805.

[4] D. Munoz, N. Vandapel, and M. Hebert, "Onboard contextual classification of 3-d point clouds with learned high-order markov random fields," in *IEEE international conference on Robotics and Automation*. IEEE, 2009, pp. 4273–4280.

[5] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, "Scene understanding in a large dynamic environment through a laser-based sensing," in *IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 127–133.

[6] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, pp. 91–98, 2017.

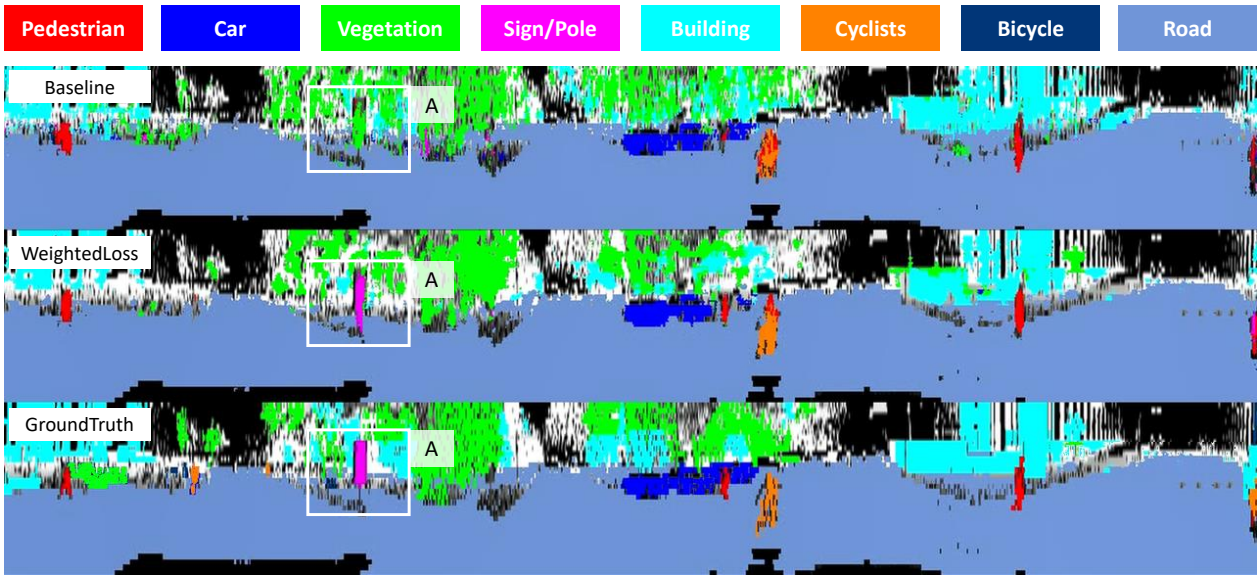
[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

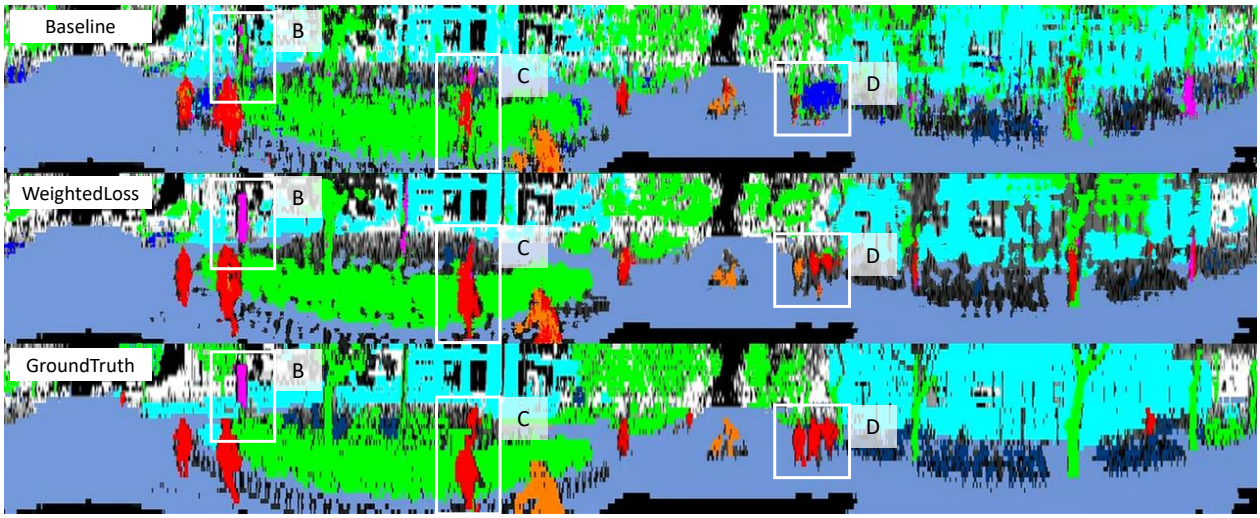
[9] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3, pp. 181–188, 2014.

TABLE II: semantic segmentation performance (mPA)

| - | Pedestrian | Car | Vegetation | Sign/Pole | Building | Cyclist | Bicycle | Road |
|---|---|---|---|---|---|---|---|---|
| Baseline Model | 0.41 | 0.71 | 0.57 | 0.36 | **0.66** | 0.49 | 0.21 | **0.99** |
| Weighted-Loss Model | **0.78** | **0.93** | **0.74** | **0.40** | **0.66** | **0.56** | **0.49** | **0.99** |

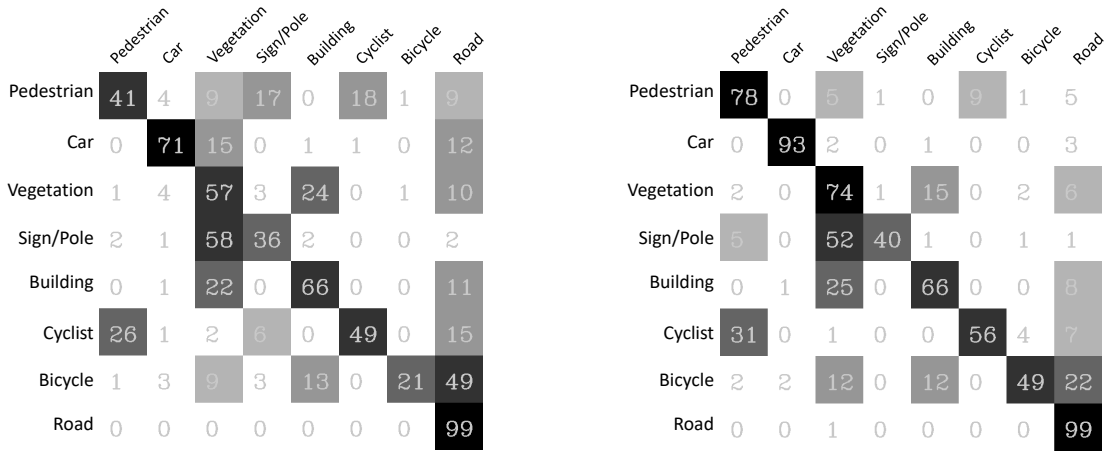



(a) Weighted loss leads to better performance of labels with less samples, such as the traffic sign in Box A.




(b) Weighted loss version segmentation model can achieve more balanced result than the baseline. For example, Box B shows the improvement of a traffic sign's label result, and Box C,D include the results improvement of pedestrain.

Fig. 3: Visualization result of semantic segmentation.




(a) Confusion matrix of baseline model

(b) Confustion matrix of weighted loss model

Fig. 4: The confusion matrix on testing dataset, values in matrices are shown in percentage. Rows means ground truth labels and columns means predicted labels.


[10] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3d point clouds with strongly varying density." *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 177–184, 2016.

[11] H. Hu, D. Munoz, J. A. Bagnell, and M. Hebert, "Efficient 3-d scene analysis from streaming data," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2297–2304.

[12] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 975–982.

[13] Y. Lu and C. Rasmussen, "Simplified markov random fields for efficient semantic labeling of 3d point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2690–2697.

[14] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *IEEE International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 716–724.

[15] P. Tosteberg, "Semantic segmentation of point clouds using deep learning," Master's thesis, Linköping University, 2017.

[16] G. L. O. a. B. Ayush Dewan, "Deep semantic classification for 3d lidar data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 3544–3549.

[17] 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.

[18] F. Piewak, P. Pinggera, M. Schäfer, D. Peter, B. Schwarz, N. Schneider, D. Pfeiffer, M. Enzweiler, and M. Zöllner, "Boosting lidar-based semantic labeling by cross-modal training data generation," *arXiv preprint arXiv:1804.09915*, 2018.

[19] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2017, pp. 1019–1024.

[20] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, "Deep projective 3d semantic segmentation," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 95–107.

[21] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," *arXiv preprint arXiv:1710.07563*, 2017.

[22] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3. IEEE, 2017, pp. 6620–6629.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 77–85.

[24] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," *arXiv preprint arXiv:1711.09869*, 2017.