Large-scale 3D Point Cloud Semantic Segmentation with 3D U-Net ASPP Sparse CNN

Naufal Muhammad Hirzi*, Muhammad Anwar Ma'sum*†, Mahardhika Pratama†, and Wisnu Jatmiko*
* Faculty of Computer Science Universitas Indonesia, Depok, Indonesia
† STEM University of South Australia, Adelaide, Australia
Email: naufal.muhammad01@ui.ac.id

2022 7th International Workshop on Big Data and Information Security (IWBIS) | 978-1-6654-8950-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWBIS56557.2022.9924988

***Abstract*—3D geometric modelling of urban areas has the potential for further development, not only for 3D urban visualization. 3D point cloud, as 3D data commonly used in 3D urban geometry modelling, is needed to extract objects from point clouds to analyze urban landscapes. An automated method to analyze objects from the 3D point cloud can be achieved by using the semantic segmentation method. Unlike other segmentation tasks in 3D point cloud data, 3D urban point cloud segmentation has the challenge of segmenting different object sizes on various types of landscape contours with imbalanced distribution of the object. Therefore, this study modified 3D U-Net Sparse CNN by adding Atrous Spatial Pyramid Pooling (ASPP) as one of the modules in this model, called 3D U-Net ASPP Sparse CNN. The use of ASPP aims to get the contextual multi-scale information of the input feature map from the encoder part of U-Net. Furthermore, 3D U-Net ASPP Sparse CNN is implemented by using weighted dice loss as the loss function. The experiment result shows 3D U-Net ASPP Sparse CNN with weighted dice loss has achieved the best evaluation score in our experiment, with OA = 96.53 and mIoU = 63.59.**

Keywords— Point Cloud, 3D Semantic Segmentation, Sparse Convolutional neural networks, 3D U-Net ASPP

I. INTRODUCTION

Point clouds are a data type commonly used in 3D urban geometry modelling applications. This data representation is obtained from the acquisition of photogrammetry measurement data, RGB-D, camera Time of Flight (ToF), Light Detection and Ranging (LiDAR), and so on [1]. 3D geometric modelling of urban areas has the potential for further development, not only for 3D urban visualization. For example, energy demand analysis, infrastructure planning, and emergency response. However, an information extraction process is needed regarding these objects, such as buildings, roads, vegetation, and others [2], [3].

Conventionally, experts manually extract the information. However, this process takes a long time and has expensive costs. The development of technology, especially in the field of artificial intelligence (especially deep learning methods), could accelerate the process of extracting information [2], [3]. Semantic segmentation is one method that makes it possible to analyze objects from point clouds. This method aims to classify the collection of points based on the class labels [4].

Point clouds are a collection of sample points that shrouds objects in 3D dimensions. The characteristics and challenges of point clouds are: 1) not evenly distributed in each scene, 2) not structured, and 3) not organized. Therefore, point clouds feature approaches are classified by input data type, that is raw point cloud and projection based [1], [5]. Furthermore, various object sizes, the imbalanced distribution of objects, and different landscape contours between one and another area are arguably a challenge for segmenting 3D urban point clouds, which distinguishes this segmentation task from other segmentation tasks such as indoors and autonomous driving cars.

Previous studies have used raw point cloud-based deep learning as their methods to perform segmentation in the 3D urban area. Bayu *et al.*[4] used PointNet on Aerial LiDAR Scanning (ALS) in Kupang, Indonesia, Wicaksono *et al.*[6] used DGCNN on ALS Depok, Indonesia, and Dublin, Ireland, and Lowphansirikul *et al.*[7] used SPGraph and PointCNN. Furthermore, several open datasets have to implement raw point cloud-based deep learning as their, namely Dales [8], SensatUrban [9], and STPLS3D [10], for which Kpconv model has the best mIoU evaluation score, respectively. However, this approach has constraints that process a limited number of points. Thus, the receptive field size is sufficiently limited [11].

In contrast, projection-based convert point clouds into structured form. Sparse Convolutional Neural Network (Sparse CNN) is one of the projection-based methods. Sparse CNN converts point clouds into sparse a voxel representation. Therefore, possible to store more points efficiently. Sparse CNN has modified U-Net [12] into 3D U-Net Sparse CNN [13]. Previous research has applied this architecture to a 3D segmentation semantic task on an indoor area of ALS point cloud 3D segmentation semantics [14]. Interestingly, different task has been performed by using this approach, such as instance segmentation [15] and object detection [11], [16]. Different from segmentation in the 3D dataset, CNN-based models dominate the semantic segmentation task in 2D images, for example, Deeplab [15],[16] and Pspnet [19]. Furthermore, modification of the basic architecture of U-Net [20]–[22], implemented the ASPP module as the bridge between the last encoder dan the first decoder.

Motivated by the success of 3D U-Net Sparse CNN as state-of-the-art 3D point cloud semantic segmentation. This study, modified 3D U-Net Sparse CNN by combining it with ASPP module, called U-Net ASPP Sparse CNN (as shown in Fig. 1), to overcome the problem and improve the performance of semantic segmentation in 3D urban point clouds. Points in point cloud are scattered around the object's surface and occupy a small part of the scene. Therefore, we implement a region-based loss function, which is weighted dice loss. Our experimental result reveals that 3D U-Net ASPP Sparse CNN with weighted dice loss achieve the best performance.

In summary, the contributions of this study are as follows:

1). Modified the 2D CNN layer in the ASPP module with 3D ResNet Sparse CNN to change the Convolution kernel dimensions and get the benefits of Sparse CNN and Residual connections from Resnet.

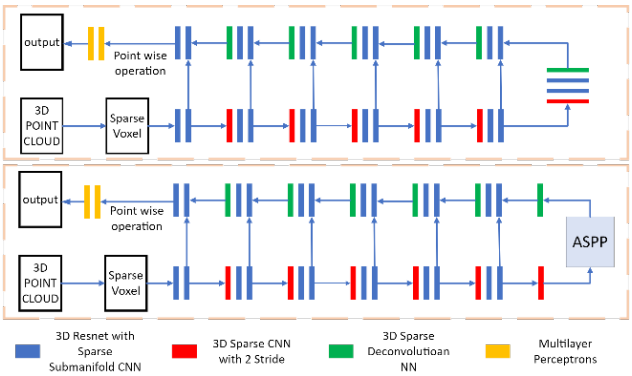


Fig. 1. Comparison between 3D U-Net Sparse CNN (top) and purpose method (bottom)

2). Replacing the loss function with weighted dice loss, due to handling the imbalanced dataset and learning the dice coefficient between the predicted object and ground truth label.

II. Proposed Method

3D U-Net ASPP Sparse CNN, modified from 3D U-Net Sparse CNN, which is built from different types of Sparse CNN that is: Sparse CNN, Sparse Submanifold CNN, and Sparse Deconvolution Neural Network (Sparse Deconvolution NN). 3D data is usually characterized by a very inhomogeneous density of spatial distribution, most of the space (voxels) is not occupied at all [11]. Sparse CNN overcomes this by convoluting only on data that has a value, while the voxel part that is not occupied at all is assumed to be zero. On the other hand, Sparse Submanifold CNN is an enhancement of Sparse CNN, which limits the output location to active if and only if the corresponding input location is active. This avoids initializing too many active locations, which causes a decrease in the speed of subsequent convolutions. Sparse Deconvolution NN is the up-sampling layer version of Sparse CNN [13].

To perform Sparse CNN, point clouds needs to convert to a sparse voxel. This process begins by grouping the point sets based on the voxel size threshold. Then, the points set are stored on the voxel threshold in one voxel and the location of the voxel relative to the input point cloud. For each voxel that has values not zero, the voxel value and location index are stored in the hash map. Further, this sparse voxel is used as input for the 3D U-Net model [16].

Our proposed method used a similar approach as [13], [14], down-sampling in the encoder path used layer Sparse CNN with $2 \times 2 \times 2$ kernel size and two strides. Up-sampling is performed using Sparse Deconvolution NN with $2 \times 2 \times 2$ kernel size and two strides. Resnet block with 3D Sparse Submanifold CNN with $3 \times 3 \times 3$ kernel size, attached in each encoder and decoder block. Different from those, between the last encoder and first decoder, we placed ASPP as a bridge. At the end of the decoder, implement a Multilayer perceptron that classifies feature map as point-wise operation, as shown in Fig. 1 (bottom).

A. Atrous Spatial Pyramid Pooling

Previous study, Deeplab [18] proposed Atrous Spatial Pyramid Pooling (ASPP) network structure. This structure uses a few parallel atrous convolutions layers with different dilation dan combines those layers. This approach allows CNN to obtain contextual multi-scale information. Inspired by the ASPP scheme, there are several modifications of U-Net which use ASPP at the end of encoder layers [20]–[22].

Different from that approach, we adopted an ASPP network structure with three modules of ResNet Sparse Submanifold CNN layers with different padding and dilatation rate (6, 12, and 18), which are connected in a parallel manner. Further, those layers are concatenated and applied $1 \times 1 \times 1$ Sparse CNN, as shown in Fig. 2.

B. Loss Function

The class imbalance problem is one of the semantic segmentation problems that are likely to occur. The loss-weighting strategy has the potential to tackle this problem. This strategy used weight for each class, which was obtained by calculating the distribution of the object in every related class. This weighting strategy has been applied to several loss functions such as cross-entropy and dice loss. Weighted cross-entropy is classified as a distribution-based loss, which measures the dissimilarity of distribution based on cross-entropy [24], [25]. Weighted cross-entropy loss defines as:

$$L_{CE} = -\frac{1}{N}\sum_{c=0}^{C} W_c \sum_{i=0}^{N} g_{i,c} \log p_{i,c} \quad (1)$$

Different from cross-entropy loss, dice loss classifies as a region-based loss. Dice loss used dice score coefficient that measures overlap between the ground truth point cloud and the predicted segmentation probability [24], [25]. Weighted dice loss (L_{Dice}) defines as:

$$L_{Dice} = 1 - \frac{2\sum_{c=0}^{C} W_c \sum_{i=0}^{N} g_{i,c} p_{i,c}}{\sum_{c=0}^{C} W_c \sum_{i=0}^{N} (g_{i,c} + p_{i,c})} \quad (2)$$

Where C are the numbers of classes, N are the numbers of points, W_c is the weight of a specific class, g is the ground truth label and p denotes predicted segmentation probability.

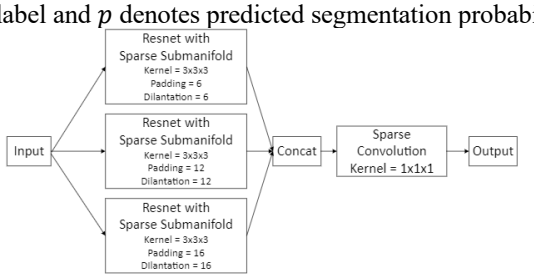


Fig. 2. Atrous Spatial Pyramid Pooling with Sparse CNN

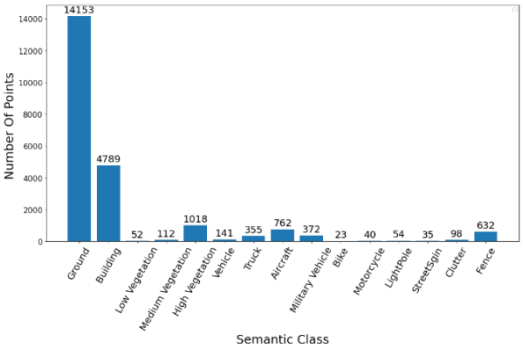


Fig. 3. Average Number of Points for Each Class

Previous research has used weighted cross-entropy on 3D U-Net Sparse CNN for the semantic segmentation ALS point cloud [14]. Inspired by V-Net [23], dice loss has benefited to learn region of the object since the object of interest only occupied a small portion of the scene. To tackle the imbalance class, used enhancement of dice loss which is weighted dice loss. Furthermore, we compare 3D U-Net ASPP Sparse CNN weighted cross-entropy and dice loss, to see the different results of both loss functions.

III. Experiment

A. Dataset

In this study, used the Semantic Terrain Points Labeling - Synthetic 3D (STPLS3D) dataset. This dataset is a large-scale urban 3D point cloud dataset reconstructed from a 2D photogrammetric Unnamed Air Vehicle (UAV) with a 3D Geographic Information System (GIS) virtual environment. This study focuses on the STPLS3D Synthetic V3 instance segmentation dataset to perform semantic segmentation [10].

Since this dataset have a large-scale area (that is x = ±490 and y = ±490), to reduce computation cost and augment data, we used similar strategies as STPLS3D based line methodology, which is to patch the data into smaller size (x = ±50 and y = ±50) with a sliding window approach. However, we apply the overlapping patch with a size ±25 (for both x and y sizes) on training dataset. For testing dataset, we used a similar approach with baseline methodology. This dataset consists of 25 sets of 3D point clouds, where the 5th, 10th, 15th, 20th, and 25th data are used as the testing dataset. While the rest is randomly split into training and validation datasets with a ratio of 80:20, respectively, after patching data was implemented.

We use coordinates (in the XYZ representation) and colors (in the RGB representation) as inputs. For the label, we use a semantic label that aims to segment 15 classes, namely: 1) ground, 2) building, 3) low vegetation, 4) medium vegetation, 5) high vegetation, 6) vehicle, 7) truck, 8) aircraft, 9) military vehicle, 10) bike, 11) motorcycle, 12) light pole, 13) street sign, 14) clutter, and 15) fence [10]. In addition, we used -100 for the class which does not need to be segmented. Fig. 3 is an illustration of the distribution of point for each class, which show the imbalances of the class distribution. We calculate class weight with $\frac{max(\sum_{i=0}^{N} g_{i,c})}{\sum_{i=0}^{N} g_{i,c}}$ as parameter on loss function to tackle imbalanced dataset.

B. Experiment Setup

This experiment was implemented in Pytorch version 1.4 and Cuda toolkit version 10.2 at NVIDIA DGX-1 Tokopedia-UI AI Center of Excellence. We are training three models that are: 1) 3D U-Net Sparse CNN with weighted cross-entropy loss function (baseline), 2) 3D U-Net-ASPP Sparse CNN with weighted cross-entropy loss function and 3) U-Net-ASPP Sparse CNN with weighted dice loss. Those models are trained with similar datasets and weight classes. Training parameters for those models are listed as: 1) epoch = 300, 2) batch size = 32, 3) learning rate = 1-e4, 4) optimizer = Adam, and 5) voxel size = 0.02m.

C. Evaluation

After those models were trained, we evaluated the result with evaluation metrics that are Overall Accuracy (OA), Intersection Over Union (IoU) for each class, and mean Intersection Over Union (mIoU). OA evaluation metrics calculated the ratio between the numbers of the sample that are classified in a certain class and the number of the total samples, which is defined in Equation (3).

$$OA = \sum_{i=0}^{K} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}. \quad (3)$$

In contrast, IoU calculates the intersection of between predicted value and ground truth divided by the union of both, which is defined in Equation (4). Therefore, mIoU can see as a summary of IoU, which averages all IoU values by the number of classes, which is defined in Equation (5).

$$IOU = \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{i}^{k} p_{ji} - p_{ii}} \quad (4)$$

$$mIOU = \frac{1}{k+1} \sum_{i=0}^{k} IoU_i \quad (5)$$

Where k denotes the number of class and p_{ii} represents the predicted point cloud which is predicted as class i denote as p_{i-} and *ground truth* denote as p_{-i} class i. With this representation, p_{ii} represents a true positive. Similarly, p_{ij}, p_{ji} and p_{jj}, represent false positive, false negative, and true negative, respectively.

IV. Result And Analysis

To facilitate, models with weighted cross-entropy loss are written as the related model names and weighted dice loss is written as dice loss. Table I present OA and mIoU evaluation result from different models. In general, the modified, 3D U-Net ASPP Sparse CNN, improve performance compared to the baseline model, with a difference in the value of OA = ±0.62 and mIoU = ±0.56. Interestingly, enhancement of 3D U-Net ASPP Sparse CNN with dice loss, giving major improvement performance compare both of 3D U-Net Sparse CNN dan 3D U-Net ASPP Sparse CNN with OA = 96.53 and IoU = 63.59. To investigate further, we present IoU evaluation for those models in Fig. 4 and the visualization of the predicted point cloud from those models in Fig. 5.

TABLE I. AO and mIoU Evaluation Result

Model	OA (%)	mIoU (%)
3D U-Net Sparse CNN (Baseline)	92.19	46.23
3D U-Net ASPP Sparse CNN	92.81	46.79
3D U-Net ASPP Sparse CNN with Dice Loss (proposed method)	96.53	63.59

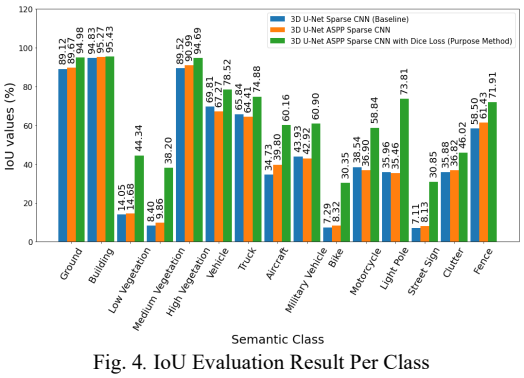
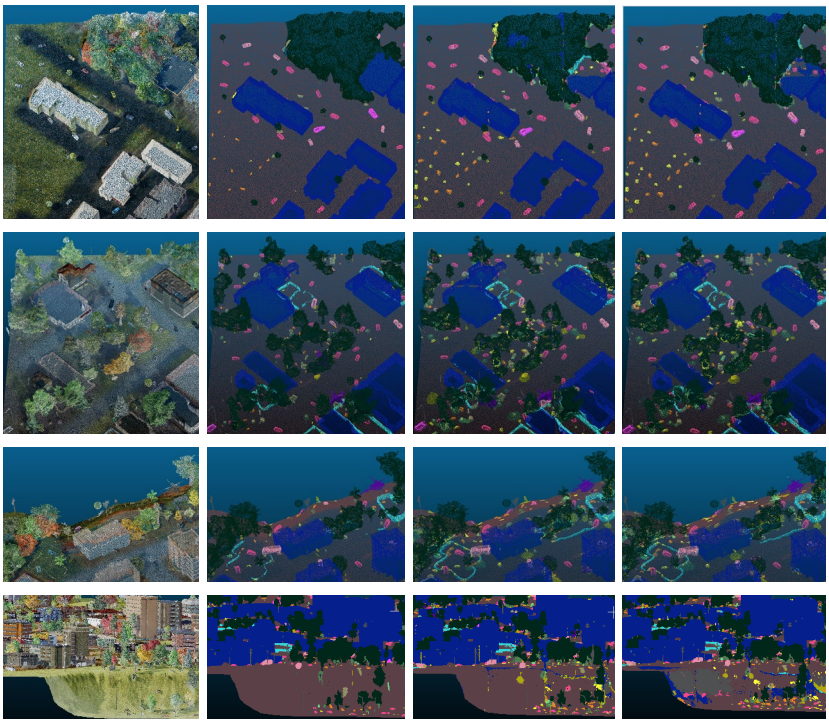
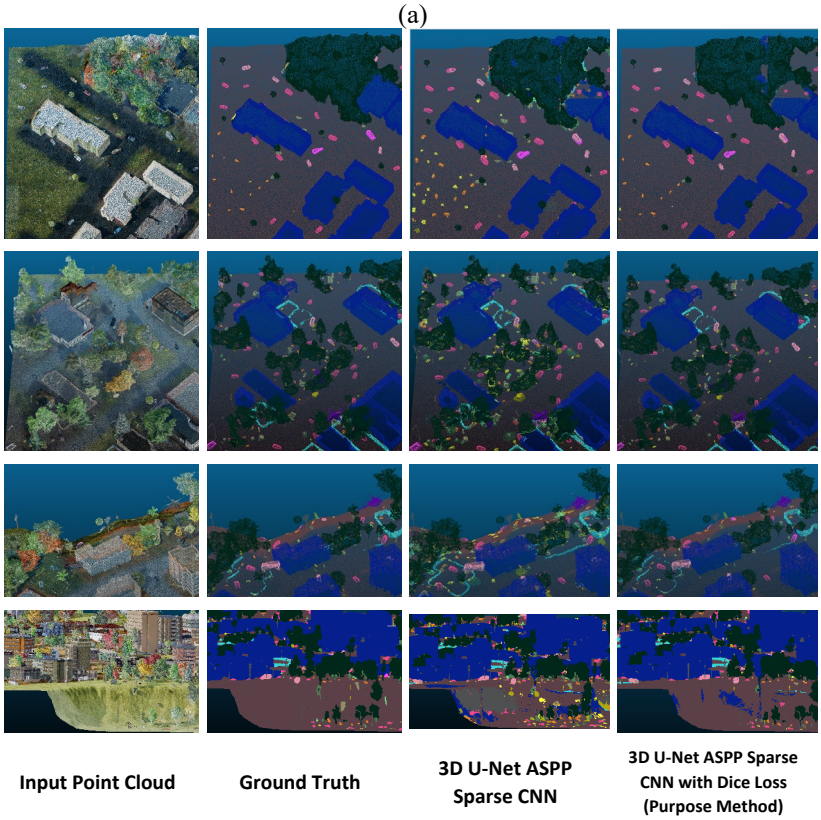


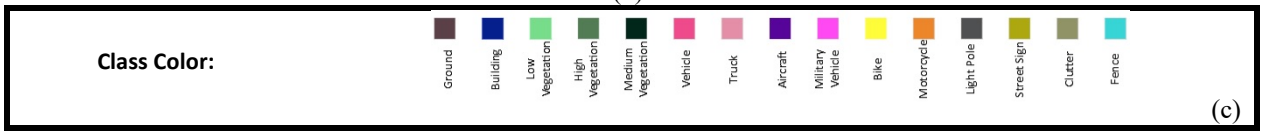
Fig. 4. IoU Evaluation Result Per Class



(a)



(b)



(c)

Fig. 5. Visualization of Predicted Label, (a) Comparison between 3D U-Net Sparse CNN and 3D U-Net ASPP Sparse CNN, And Comparison between 3D (b) U-Net ASPP Sparse CNN and 3D U-Net ASPP Sparse CNN with weighted dice loss, and (c) Legend of color mapping for each class

Compared to 3D U-Net Sparse CNN, 3D U-Net ASPP Sparse CNN shows better for segmenting in class: ground, building, vegetation (low, medium, and high), aircraft, bike street sign, clutter, and fence, based on Fig. 4. Based on the visualization in Fig. 5 (a) 1st, 2nd, and 3rd row, both of models seem difficult to segmenting small object (such as bike, low vegetation, etc.). Small object classes are often segmented across object boundaries, misclassified into different small object classes, and misclassified ground class to class with small objects. However, 3D U-Net ASPP Sparse CNN shows better results by reducing the number of outliers and small objects misclassified. For a class with large object (namely building and high vegetation), 3D U-Net ASPP Sparse CNN show better for segmenting rather than 3D U-Net Sparse CNN. In a steep valley with various types of objects as shown in Fig. 5 (a) 4th row, 3D U-Net Sparse CNN shows better results, because in certain classes that 3D U-Net Sparse CNN is better for segmenting rather than 3D U-Net Sparse ASPP CNN (as shown in Fig. 4).

The best performance, achieved by 3D U-Net ASPP Sparse CNN with dice loss. This method seems superior rather than the two models before. Arguably, this boosting is caused by improvement in class with small and low point sets, based on Fig. 5 (b), 1st, 2nd, and 3rd rows. The steep valley problem in Fig. 5 (b) 4th row shows better segmented compared to the previous experiment model. Since on preprocessing section performed the process of patching the large-scale 3D point cloud into a smaller size. Due to preprocessing section performed the process of patching the large-scale 3D point cloud into a smaller size, led the problem to perform segmentation. For example, on Fig. 5 (both of a and b) rows 1st, those models are segmenting some points in high vegetation class as building class at similar place.

V. Conclusion

This study has performed semantic segmentation on a 3D point cloud photogrammetry using modified 3D U-Net Sparse CNN, namely 3D U-Net ASPP Sparse CNN. Based on our experiment result, using ASPP shows better results (OA = 92.81and mIoU = 46.79) compared to baseline model (OA = 92.19 and mIoU = 46.23). Furthermore, improvement of 3D U-Net ASPP Sparse CNN by using weighted dice loss as a loss function boosted the performance and achieved the best evaluation in our experiment (OA = 96.53 and mIoU = 63.59). The process of divided (patching) large-scale point clouds into smaller forms causes errors to segment objects due to the shape of objects that are truncated in the data patching process. Future study needs to compare different patching sizes and strategy to optimize the process of semantic segmentation of large-scale urban point cloud. In addition, the different sources of point cloud have different characteristics, for example, ALS has denser points than the photogrammetry point cloud. Therefore, to evaluate the robustness of the model need to test with a different source of 3D urban point cloud.

Acknowledgment

We gratefully acknowledge the support from Tokopedia UI AI Center, Faculty of Computer Science, University of Indonesia and PUTI Q1 from Universitas Indonesia for research project entitled “Asesmen Kerusakan Bangunan Akibat Bencana Gempa Bumi Menggunakan Residual Siamese Neural Network Pada Data Lidar”, with number NKB-395/UN2.RST/HKP.05.00/2022.

References

[1] S. A. Bello, S. Yu, and C. Wang, “Review: deep learning on 3D point clouds,” *arXiv*, 2020.

[2] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, “Applications of 3D city models: State of the art review,” *ISPRS Int. J. Geo-Information*, vol. 4, no. 4, pp. 2842–2889, 2015, doi: 10.3390/ijgi4042842.

[3] Y. Ying, M. N. Koeva, M. Kuffer, and J. A. Zevenbergen, “Urban 3D modelling methods: A state-of-the-art review,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 43, no. B4, pp. 699–706, 2020, doi: 10.5194/isprs-archives-XLIII-B4-2020-699-2020.

[4] A. Bayu, N. S. Intizhami, A. Wibisono, W. Jatmiko, H. A. Wisesa, and A. Gamal, “Semantic Segmentation of Lidar Point Cloud in Rural Area,” pp. 73–78, 2019.

[5] Y. He *et al.*, “Deep Learning based 3D Segmentation: A Survey,” *Proc.*, vol. 1, no. 1, pp. 1–36, 2021.

[6] S. B. Wicaksono, A. Wibisono, W. Jatmiko, A. Gamal, and H. A. Wisesa, “Semantic Segmentation on LiDAR Point Cloud in Urban Area using Deep Learning,” *2019 Int. Work. Big Data Inf. Secur. IWBIS 2019*, no. 2018, pp. 63–66, 2019, doi: 10.1109/IWBIS.2019.8935882.

[7] C. Lowphansirikul, K. S. Kim, P. Vinayaraj, and S. Tuarob, “3D Semantic Segmentation of Large-Scale Point-Clouds in Urban Areas Using Deep Learning,” *2019 11th Int. Conf. Knowl. Smart Technol. KST 2019*, pp. 238–243, 2019, doi: 10.1109/KST.2019.8687813.

[8] N. Varney, V. K. Asari, and Q. Graehling, “DALES: A large-scale aerial LiDAR data set for semantic segmentation,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 717–726, 2020, doi: 10.1109/CVPRW50498.2020.00101.

[9] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, “Towards Semantic Segmentation of Urban-Scale 3D Point Clouds: A Dataset, Benchmarks and Challenges,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4975–4985, 2021, doi: 10.1109/CVPR46437.2021.00494.

[10] M. Chen *et al.*, “STPLS3D : A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset,” *Eccv*, pp. 1–27, 2022.

[11] C. Choy, J. Gwak, and S. Savarese, “4D spatio-temporal convnets: Minkowski convolutional neural networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 3070–3079, 2019, doi: 10.1109/CVPR.2019.00319.

[12] W. Weng and X. Zhu, “UNet: Convolutional Networks for Biomedical Image Segmentation,” *IEEE Access*, vol. 9, pp. 16591–16603, 2021, doi: 10.1109/ACCESS.2021.3053408.

[13] B. Graham, M. Engelcke, and L. Van Der Maaten, “3D Semantic Segmentation with Submanifold Sparse Convolutional Networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9224–9232, 2018, doi: 10.1109/CVPR.2018.00961.

[14] S. Schmohl and U. Sörgel, “Submanifold Sparse Convolutional Networks for Semantic Segmentation of Large-Scale ALS Point Clouds,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 4, no. 2/W5, pp. 77–84, 2019, doi: 10.5194/isprs-annals-IV-2-

W5-77-2019.

[15] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical Aggregation for 3D Instance Segmentation," pp. 15447–15456, 2022, doi: 10.1109/iccv48922.2021.01518.

[16] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors (Switzerland)*, vol. 18, no. 10, pp. 1–17, 2018, doi: 10.3390/s18103337.

[17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," Jun. 2017, doi: 10.48550/arxiv.1706.05587.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," Jun. 2016, doi: 10.48550/arxiv.1606.00915.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6230–6239, 2017, doi: 10.1109/CVPR.2017.660.

[20] H. He, D. Yang, S. Wang, S. Wang, and Y. Li, "Road Extraction by Using Atrous Spatial Pyramid Pooling Integrated Encoder-Decoder Network and Structural Similarity Loss," *Remote Sens.*, vol. 11, no. 9, 2019, doi: 10.3390/rs11091015.

[21] J. Wang, P. Lv, H. Wang, and C. Shi, "SAR-U-Net: squeeze-and-excitation block and atrous spatial pyramid pooling based residual U-Net for automatic liver segmentation in Computed Tomography," Mar. 2021, doi: 10.48550/arxiv.2103.06419.

[22] X. Qiu, "U-Net-ASPP: U-Net based on atrous spatial pyramid pooling model for medical image segmentation in COVID-19," *J. Appl. Sci. Eng.*, vol. 25, no. 6, pp. 1015–1024, 2022, doi: 10.6180/jase.202212_25(6).0012.

[23] F. Milletari, N. Navab, and S. A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," *Proc. - 2016 4th Int. Conf. 3D Vision, 3DV 2016*, pp. 565–571, 2016, doi: 10.1109/3DV.2016.79.

[24] S. Jadon, "A survey of loss functions for semantic segmentation," Jun. 2020, doi: 10.1109/cibcb48159.2020.9277638.

[25] T. Sugino, T. Kawase, S. Onogi, T. Kin, N. Saito, and Y. Nakajima, "loss weightings for improving imbalanced brain structure segmentation using fully convolutional networks," *Healthc.*, vol. 9, no. 8, 2021, doi: 10.3390/healthcare9080938.