Transformable Dilated Convolution by Distance for LiDAR Semantic Segmentation

JAE-SEOL LEE\(^1\) AND TAE-HYOUNG PARK\(^2\), (Member, IEEE)

\(^1\)Industrial AI Research Center, Chungbuk National University, Cheongju, Chungcheongbuk 28644, South Korea
\(^2\)Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju, Chungcheongbuk 28644, South Korea

Corresponding author: Tae-Hyoung Park (taehpark@cbnu.ac.kr)

This work was supported by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information Technology Research Center Support Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2022-2020-0-01462.

ABSTRACT LiDAR semantic segmentation is essential in autonomous vehicle safety. A rotating 3D LiDAR projects more laser points onto nearby objects and fewer points onto farther objects. Therefore, when projecting points onto a 2D image, such as spherical coordinates, nearer objects appear larger than more distant objects. Recognizing a closer object requires a larger receptive field, whereas recognizing a nearer object requires a smaller receptive field. However, existing CNNs have always used the same receptive field, making it difficult to express objects of various sizes in a single-sized receptive field, restricting their performance in terms of the recognition of larger (or nearer) objects that require a larger receptive field. In response to these limitations, we propose a transformable dilated convolution (TD Conv) to adjust the convolution filter’s size according to the input distance. Leveraging the distance information of LiDAR and dilated convolution, a large convolution was applied to nearby objects, and a small convolution was applied to farther objects. The proposed method yielded good performance when recognizing nearer objects or larger objects such as roads and buildings and showed similar performance to the conventional method for farther or smaller objects. To test the proposed method, we used the SemanticKITTI dataset.

INDEX TERMS Autonomous vehicles, convolutional neural network, LiDAR, point cloud, semantic segmentation, spherical coordinate transformation.

I. INTRODUCTION

Semantic segmentation is used to recognize various complex traffic environments. Several studies [1], [2], [3], [4] on 2D image-based road environments were conducted following the release of the Cityscape [5] and Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) datasets [6]. However, 2D image-based semantic segmentation methods have limitations in measuring the exact distances between vehicles and road obstacles.

Accurate measurements of these distances are essential in autonomous vehicle safety. With the recent release of the LiDAR point-based semantic segmentation datasets SemanticKITTI [7] and nuScenes [8] for road environments, research on point-based semantic segmentation for road environments is actively being conducted [9]. In this study, we considered 3D LiDAR-based semantic segmentation to solve the limitations of 2D methods.

LiDAR point semantic segmentation methods can be divided into point-based, voxel-based, and range-based methods. Other methods include processing LiDAR data by converting them into graph type data [32], [33], [34], [35]. However, these methods are not widely used for autonomous vehicles. Processing is time consuming in many point environments.

Point-based methods extract and directly process a distinct point’s features through a multilayer perceptron (MLP)-based network. Methods such as PointNet [10] and PointNet++ [11] typically process indoor point clouds, while RandLA-Net [12] and kernel point convolution (KPConv) [13] process outdoor point clouds. However, the MLP-based network slows down when many points, such as in a road environment,
are being processed. It is difficult to expect real-time performance from these methods because there are many points to process from the high-channel LiDAR mainly used in autonomous vehicles. If the data are reduced to minimize computation, the performance will become poor, and the results will be difficult to use.

The voxel-based method uses a 3D convolution by dividing the 3D space into a predetermined grid. This type of method includes VoxeNet [14], attentive feature fusion with adaptive feature selection for sparse semantic segmentation network ((AF)2-S3Net) [15], TORNADO-Net[16], and sparse point-voxel neural architecture search (SPVNAS) [17]. However, the appropriate data for this method are limited, and the convolution operation is inefficient. In addition, 3D convolution is slower, as it includes more computation compared to 2D convolution.

The range-based method processes points by projecting 3D point data onto a 2D image using spherical coordinates of 2D convolutional neural networks (CNNs) [18], [19]. CNN can be applied directly, and it is the most used method because it requires the least amount of computation among the above three methods. The RangeNet [20] and SalsaNext [21] methods use an encoder-decoder structure and improve their performance using the CNN method in postprocessing. However, the standard CNN has a fixed-size kernel, i.e., it has the same receptive field regardless of the object’s size. In addition, 3D spatial information is lost as it is projected as 2D data.

Therefore, it is difficult to express various types of objects. To solve this problem, convolution methods that are more flexible than existing convolution methods have been proposed. Deformable convolutional networks [22] dynamically learn the sampling locations by predicting the offsets for each image location. Pixel-adaptive convolution [23] dynamically multiplies the weights of the filters and a spatially varying kernel to make the standard convolution content adaptive. For ADCNN [24], a method of acquiring various spatial information is proposed by learning the expansion coefficient for each location. Since these methods are adaptive convolution methods based on 2D images, they are not considered for 3D point cloud data.

In this paper, we propose the TD (transformable dilated) convolution method for LiDAR semantic segmentation. This method takes advantage of the characteristics of 3D LiDAR data; these data are radial, objects closer to the sensor are larger, and objects farther from the sensor are smaller. 1) Considering these features, TD Conv uses dilated convolution [25] to apply a large kernel for near objects and a small kernel for far objects. 2) In addition, the proposed TD Conv method is added to the spatial attention block structure of SqueezeSegV3 [26] to solve the problem that 3D information is lost in transformed 2D data; thus, its performance is improved. Moreover, the inference time is reduced by replacing the $7 \times 7$ convolution kernel of the spatial attention block [26] with the $3 \times 3$ TD Conv proposed in this study. The contributions of this paper can be summarized in two parts.

1. We propose TD convolution, which is computationally efficient by considering the characteristics of 3D LiDAR data. 2. By replacing the SAC (spatially adaptive convolution) module of SqzV3 with the proposed TD block, the speed of the network and mIoU for objects close to LiDAR are improved.

This paper continues as follows. In Chapter II, we describe the data transformation process. In Chapter III, we present the proposed convolution. In Chapter IV, we present the network structure, and in Chapter V, our experimental results using SemanticKITTI are described.

**II. DATA TRANSFORMATION**

The SemanticKITTI dataset consists of 3D Cartesian coordinate data, while LiDAR data are a set of points scanned by laser scanners. Let $p_k \in \mathbb{R}^{1 \times 4}$ ($k = 1, \cdots, K$) be the $k$th point of LiDAR data as:

$$
p_k = [x_k, y_k, z_k, \bar{d}_k]
$$

where $x_k, y_k, z_k$ are the Cartesian coordinates of the point, and $\bar{d}_k$ is the intensity data. Then, the point cloud of LiDAR is represented by the row vector $P \in \mathbb{R}^{1 \times 4K}$ as:

$$
P = [p_1 \cdots p_K]
$$

We transform the LiDAR point cloud data into spherical coordinates before applying it to a 2D CNN.

**A. SPHERICAL COORDINATE TRANSFORMATION**

The spherical coordinate system is a coordinate system for a 3D space, where the position of a point is specified by the radial distance ($r$), the azimuth angle ($\theta$), and the elevation angle ($\phi$). Note that the azimuth angle is the angle in the horizontal plane, and the elevation angle is in the vertical plane.
A spherical coordinate system was used in some studies to process ground LiDAR data.

The original raw data of a 3D LiDAR are expressed in spherical coordinates (Fig. 1). In the LiDAR coordinate system $\{L\}$, the azimuth angle ($\theta$) is in the horizontal plane (X-Y), and the elevation angle is in the vertical plane (Y-Z). The scanning resolution of the LiDAR in the horizontal direction determines the resolution of the azimuth angle ($1\theta$). Additionally, the number of LiDAR channels in the vertical direction determines the resolution of the elevation angle ($1\phi$).

If the point of LiDAR data $(x_k y_k z_k i_k)$ is projected onto the spherical coordinates $(\theta, \phi)$, $(x_{\theta \phi}, y_{\theta \phi}, z_{\theta \phi}, i_{\theta \phi})$ is defined as the values of the spherical coordinates. For the LiDAR data point $(x_k y_k z_k i_k)$, the corresponding values of the spherical coordinates are obtained by

\begin{align}
x_{\theta \phi} &= x_k \\
y_{\theta \phi} &= y_k \\
z_{\theta \phi} &= z_k \\
i_{\theta \phi} &= i_k
\end{align}

Then, the position of spherical coordinate $(\theta, \phi)$ is determined from $(x_k y_k z_k)$ as

\begin{align}
\theta &= \tan^{-1} \frac{x_k}{z_k} \\
\phi &= \tan^{-1} \frac{z_k}{\sqrt{x_k^2 + y_k^2}}
\end{align}

If $(\theta, \phi)$ space is divided by $\Theta \times \Phi$ grids, then the X-channel, Y-channel, Z-channel, and I-channel of LiDAR data can be represented by the matrices $S_X, S_Y, S_Z$ and $S_I \in \mathbb{R}^{\Theta \times \Phi}$ as

\begin{align}
S_X &= [x_{\theta \phi}]_{\Theta \times \Phi} \\
S_Y &= [y_{\theta \phi}]_{\Theta \times \Phi} \\
S_Z &= [z_{\theta \phi}]_{\Theta \times \Phi} \\
S_I &= [i_{\theta \phi}]_{\Theta \times \Phi}
\end{align}

Fig. 2 shows the final input data of the LiDAR data projected in spherical coordinates. From the data, there are a total of four channel images for semantic segmentation.

FIGURE 2. Examples of channel data in the spherical coordinate system. (width × height: 2048 × 64 pixels).

III. TRANSFORMABLE-DILATED CONVOLUTION

In this study, we propose a convolution that changes the kernel size to apply a receptive field that fits the object size. As seen in Fig. 3, the rotating 3D LiDAR has a rotation resolution of $\Delta \theta$ and radial data characteristics. Therefore, many LiDAR points are projected on a nearby object and fewer points are projected on an object that is farther away, as confirmed in Fig. 4, where LiDAR data are transformed to spherical coordinates. Thus, the required receptive field for recognizing a particular object is different for each object.

The spherical coordinate data using the LiDAR data can be used to measure the distance value, making it possible to approximate an object’s size. The proposed TD Conv changes the size of the convolution kernel according to the distance using such features.

The number of computations increases if the kernel size is increased. Therefore, dilated convolution, a method of expanding the receptive field without increasing the number of computations, is used. As the dilation factor is adjusted...
TABLE 1. Example of transformable dilated convolution & standard convolution by distance.

| Distance | $d_{\theta \phi} = 20m - 40m$ | $d_{\theta \phi} = 40m - 60m$ | $d_{\theta \phi} = 60m - 80m$ |
|----------|--------------------------------|--------------------------------|--------------------------------|
| Standard Convolution | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ |
| Transformable Convolution | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ | $f_{\theta \phi} = \sum_{i,j=1}^{1} a_{\theta+i,j} \times w_{i,j}$ |

according to the distance, the kernel size is also consequently adjusted.

**A. STANDARD CONVOLUTION & DILATED CONVOLUTION**

In a 2D image of size $\Theta \times \Phi$, the conventional convolution (a $3 \times 3$ kernel with a dilation of 1) operation at the $\theta, \varphi$ position is calculated as

$$f_{\theta,\varphi} = \sum_{i,j=-1}^{1} a_{\theta+i,\varphi+j} \times w_{i,j} \quad (13)$$

Here, $a_{\theta,\varphi}$ and $f_{\theta,\varphi}$ are input and output features at the $\theta, \varphi$ positions, respectively, $w$ is the trained weight of the convolution, and $i, j$ are the kernel coordinates. Let $n$ be the dilation factor; then, the dilated convolution operation can be expressed as

$$f_{\theta,\varphi} = \sum_{i,j=-1}^{1} a_{\theta+i\times n,\varphi+j\times n} \times w_{i,j} \quad (14)$$

**B. TRANSFORMABLE DILATED CONVOLUTION**

The TD Conv method proposed in this paper is shown in Fig. 6. The order of transformable convolutions at the positions of $\theta, \varphi$ is as follows:

1) Calculate the dilation factor from the distance data and create a dilation index map.
2) In the dilation index map, create a dilated Conv kernel that fits the size of the dilation factor at positions $\theta, \varphi$. 

Fig. 5 shows the difference between the standard convolution and the dilated convolution with dilation factors of 1 and 2, respectively. It shows that the sum of the standard convolution and the 2-dilated convolution is the same; however, the receptive field of the dilated convolution is wider. Therefore, it is possible to control the receptive field effectively.
3) Execute the convolution operation using the created dilated conv kernel.

The distance value \( d_{\theta \varphi} \) of the distance map is calculated by

\[
d_{\theta \varphi} = \sqrt{x_{\theta \varphi}^2 + y_{\theta \varphi}^2 + z_{\theta \varphi}^2}
\]

(15)

The LiDAR distance data can be represented by the matrices \( S_D \in \mathbb{R}^{\Theta \times \Phi} \) as

\[
S_D = [d_{\theta \varphi}]_{\Theta \times \Phi}
\]

(16)

**FIGURE 7.** Graph of Equation (17).

The dilation factor \( k_{\theta \varphi} \) in the dilation index map from \( d_{\theta \varphi} \) is calculated by

\[
k_{\theta \varphi} = \max(1, [-a \times d_{\theta \varphi} + c])
\]

(17)

where \( c \) is the maximum dilation factor and \( a \) is the gradient considering the distance and the maximum dilation factor. The minimum dilation factor is 1, and a floor function is applied. Equation (17) is expressed as a graph, as shown in Fig. 7.

Dilation index data can be represented by the matrices \( S_K \in \mathbb{R}^{\Theta \times \Phi} \) as

\[
S_K = [k_{\theta \varphi}]_{\Theta \times \Phi}
\]

(18)

Let the proposed TD Conv function be called \( F \), and it can be expressed as follows.

\[
F(a_{\theta \varphi}, k_{\theta \varphi}) = f_{\theta \varphi}
\]

(19)

\[
f_{\theta \varphi} = \sum_{i,j=1}^{1} a_{\varphi+i \times k_{\theta \varphi} + \theta+j \times k_{\theta \varphi}} \times w_{i,j}
\]

(20)

where \( a_{\theta \varphi}, k_{\theta \varphi} \) are the input feature values and dilation factors of the proposed convolution at the \( \theta, \varphi \) positions, and \( f_{\theta \varphi} \) is the output of the proposed convolution at the \( \theta, \varphi \) positions.

Using (17), an example of changing the convolution kernel according to distance when \( a \) is \(-0.05\) and \( c \) is 5 is shown in Table 1. The dilation factor calculated from 0 m to 10 m is 4; for 20 m to 30 m, it is 3; for 40 m to 30 m, it is 2; and for 50 m to 80 m, it is 1.

The dilation factor is calculated from a distance through (17), while a convolution kernel is generated according to the dilation factor. Subsequently, the proposed convolution creates a large kernel with a near distance and a small kernel with a far distance so that the receptive field can be adjusted according to the object’s size.

**IV. NETWORK STRUCTURE**

Fig. 8 shows the proposed network structure using TD Conv. The inputs for TD Conv are \( S_X, S_Y, S_Z, S_I \), i.e., 4-channel images projected in spherical coordinates, and distance data \( S_D \). The size of the output data is 20 \( \times \) 64 \( \times \) 2048 because the output has to classify 20 classes.

The overall network structure uses an encoder-decoder, and the encoder’s features are added to the decoder through a skip connection [31], [32]. This structure is then used in SqueezeSegV3. The SAC module of the existing SqzV3 encoder is changed to the TD block, which is the proposed method.

The encoder consists of a TD block and downsampling. The TD block is a convolution module that applies TD Conv, and a 2 \( \times \) 2 convolution with a stride of 2 used for downsampling to reduce only the width. Only the width is reduced.
because the height value of the converted input data size is small enough to 64.

In the decoder, a 3 × 3 transpose convolution with a horizontal stride of 2 is used for upsampling. Two residual blocks [30] are used for feature extraction, and loss is added for each size for multiscale accuracy. In this case, the loss consists of 3 × 3 conv and softmax.

To train the proposed network, we use a multilayer cross-entropy (CE) loss. [33], [34]. The loss is calculated by

\[
Loss = \sum_{i=1}^{5} CE(out_i)
\]  

where \( out_i \) is the output from stage 1 to stage 5 (Fig. 8). \( i \) is each stage number. For each step, the output size is 1/8, 1/8, 1/4, 1/2, and 1 times the size of the input data.
V. EXPERIMENT

We used a computer with an Intel Core i9 10th Gen. CPU, Nvidia GeForce RTX\textsuperscript{TM} 3090 GPU 2ea, Ubuntu 16.04 OS and PyTorch were used to perform the experiment. Specifically, 64 × 2048 (height by width) input data from SemanticKITTI were used to experiment with TD Conv. Semantic KITTI’s sequences labeled 1–10 were used for training and validating (8), while the unlabeled sequences 11–21 were used for testing. In addition, the size of the input data was 64 × 2048. That is, input data were a very large image on the horizontal axis, but a small image on the vertical axis. Therefore, only the horizontal axis was dilated.

The evaluation metric used was the mean intersection over union (mIoU) and the IoU of each class. mIoU is the mean value of IoU over all classes. In addition, we used stochastic gradient descent (SGD) as the optimizer, with a starting learning rate of 0.001. The warm-up performed 1 epoch, and then a cyclic learning rate scheduler performed 150 epochs.

Table 2 shows a comparison of the proposed model with previously published methods of projecting onto 2D images [20], [21], [23], [24], [25], [26]. The proposed method’s performance in projecting larger objects, such as roads, parking areas, and sidewalks, was better than that of the previous methods. Because the proposed method was able to obtain a suitable receptive field area for a large object, it also showed better performance on average-sized objects, such as people and vehicles, and no significant performance reduction was observed.

However, the method yielded relatively low performance on smaller objects or objects with details that needed to be recognized, such as bicyclists, motorcyclists, and traffic signs. This issue is attributable to the fact that detailed recognition is difficult when the convolution size is expanded.

Fig. 10 shows a model of the mIoU by distance among the projection methods. For the evaluation data, the 8th sequence, which is the validation set, was used. For the comparison model, a high-resolution network (HRNet) [1] among 2D semantic segmentation methods and SqueezeSegV3 [23], a previous LiDAR projection method, were compared. In the proposed method, the proposed convolution was modified on the existing SqueezeSegV3 (Fig. 8).

HRNet is a model with good performance in 2D image-based segmentation and consists of a total of 4 stages. HRNet + TD Conv is a model in which TD Conv is additionally applied at the starting point of each stage to the existing HRNet, and the input and output are the same as the proposed network. Fig. 10 shows that the smaller the distance is, i.e., the closer to the sensor the object is, the greater the performance improvement. Thus, it can be assumed that the previous methods did not have a sufficient receptive field for nearby
objects. HRNet and HRNet + TD Conv greatly improved the overall performance, especially at close distances.

Table 3 shows the comparative results between SqzV3 [23] and the proposed method at different distances using the validation dataset. “Other-ground” and “Motorcyclist” were excluded from the performance comparison because the IoU was close to 0 for both methods. The comparison was carried out using the same method as in Table 2. From 0 to 20 m, the proposed method showed better performance than SqzV3 for most classes. The finding indicates that the proposed method yields better results when the distance to the object is shorter. However, the performance at 20 m was slightly lower for smaller objects such as bicycles and traffic signs.

From 20 m to 50 m, the proposed method showed relatively high IoUs for larger objects such as roads, sidewalks, and parking lots. For smaller objects, the proposed method yielded similar or slightly lower performance. The proposed method showed a slightly higher mIoU, which is approximately 3% higher at distances up to 20 m and slightly higher at farther distances. These findings show that the proposed method yielded slightly lower performance on smaller objects but higher performance on larger objects.

Fig. 11 shows a comparison between the segmentation results generated by the proposed (SqueezeSegV3 + TD Conv) and existing (SqueezeSegV3) methods. As indicated by the blue circle in Fig. 11(a) and Fig. 11(c), the existing method could not match the ground truth elements. Likewise, as indicated by the red circles in Fig. 11(a) and Fig. 11(b), the existing method could not distinguish the elements ‘Road’, ‘Parking,’ and ‘Sidewalk,’ which were close to each other and of similar shape. In contrast, the proposed method accurately determined the elements and distinguished each element regardless of shape similarity and proximity to other elements.

Table 4 shows a comparison of the performance and time for different segmentation input types. The performance of the point-based method was measured using PointNet-based KPConv, and the performance measurement of the voxel-based method was carried out using TORNADO-NET. The performance of the range image-based method was compared with that of the proposed method. Speed was measured by calculating the average speed of identical 1,000 LiDAR sequences.

In addition, Table 4 shows that the proposed method achieved the highest speed, 5 fps faster than the existing SqzV3. In the conventional SqzV3, converting the 7 × 7 kernel to 3 × 3 TD Conv increased the speed by 30%. The total number of parameters of SqzV3 and the proposed method were 26 M and 21 M, respectively, and the proposed method was approximately 20% lighter than the existing SqzV3. The point and voxel methods achieved better performance than the proposed method. However, the speed was <10 fps, while LiDAR acquired data at 10 fps. Considering this, a method offering <10 fps is difficult to use in an autonomous vehicle.

Overall, the proposed method’s projection of large objects, such as roads and buildings, was similar to the ground truth with a high intersection over union (IoU) for roads, sidewalks, and buildings, as seen in Table 2. Furthermore, the road and sidewalk parking areas were accurately located, similar to the results of more accurate findings for nearby objects, as shown in the graph in Fig. 10(b).

VI. CONCLUSION

In this paper, a new convolution method for LiDAR point cloud data projected into spherical coordinates was proposed. Since LiDAR has a radial characteristic of distance, the image is converted to spherical coordinates, nearby objects appear
very large, and distant objects appear very small. In segmentation and object detection, the required receptive field is different for each object size to recognize an object. TD Conv was proposed to fit a suitable receptive field for each object. The experiment was a SemanticKITTI benchmark test for LiDAR segmentation. This confirmed that the TD Conv’s performance was better than previous projection-based methods. In addition, through the distance-specific results (Fig. 10), it was confirmed that the effect of the proposed method at a close distance was greater. However, for distant or small objects, the performance was not significantly improved or was poor.

In the future, research on far-distance attention will be conducted to improve the detection performance of even small objects in spherical LiDAR coordinates. In addition, the results, by applying the proposed method to 3D object detection, will be checked.

REFERENCES

[1] A. Tao, K. Sapra, and B. Catanzaro, “Hierarchical multi-scale attention for semantic segmentation,” 2020, arXiv:2005.10821.

[2] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, “Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 12472–12482, doi: 10.1109/CVPR42600.2020.01249.

[3] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, “Improving semantic segmentation via video propagation and label relaxation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 8848–8857, doi: 10.1109/CVPR.2019.00906.

[4] J. Lambert, Z. Liu, O. Sener, J. Hays, and V. Koltun, “MSeg: A composite dataset for multi-domain semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 2876–2885, doi: 10.1109/CVPR42600.2020.00295.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” Int. J. Robot. Res., vol. 32, no. 11, pp. 1231–1237, Aug. 2013, doi: 10.1177/0278364913491297.

[7] J. Behley, M. Garbade, A. Milioiti, J. Quinell, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Seoul, South Korea, Oct. 2019, pp. 9296–9306, doi: 10.1109/CVPR.2019.00939.

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, Oct. 2019, pp. 6410–6419, doi: 10.1109/ICCV.2019.00651.

[9] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, “(AF)-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation,” 2021, arXiv:2105.04530.

[10] M. Gerdzhev, R. Razani, E. Taghavi, and B. Liu, “TORNADO-Net: Multiview total variation semantic segmentation with diamond island module,” 2020, arXiv:2008.10544.

[11] J.-S. Lee and T.-H. Park, “Fast road detection by CNN-based camera–LiDAR fusion and spherical coordinate transformation,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 9, pp. 5802–5810, May 2020, doi: 10.1109/TITS.2020.2988302.

[12] L. Miao, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 11105–11114, doi: 10.1109/CVPR42600.2020.01112.

[13] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, Oct. 2019, pp. 6410–6419, doi: 10.1109/ICCV.2019.00651.

[14] M. Yan, Z. Li, X. Yu, and C. Jin, “An end-to-end deep learning network for 3D object detection from RGB-D data based on Hough voting,” IEEE Access, vol. 8, pp. 138810–138822, 2020, doi: 10.1109/ACCESS.2020.3012695.

[15] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, “(AF)-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation,” 2021, arXiv:2012.04162.
[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[31] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder–decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 2017, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[32] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015 (Lecture Notes in Computer Science), vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241, doi:10.1007/978-3-319-24574-4_28.

[33] R. Y. Rubinstein and D. P. Kroese, The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization Monte-Carlo Simulation and Machine Learning. New York, NY, USA: Springer, 2013.

[34] W. Shen, B. Wang, Y. Jiang, Y. Wang, and A. Yuille, “Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 2391–2400.

JAE-SEOL LEE received the B.S. degree in electronic engineering, the M.S. degree in control and robot engineering, and the Ph.D. degree from Chungbuk National University, South Korea, in 2013, 2015, and 2021, respectively. From 2016 to 2017, he was with Mirtec, Gunpo, South Korea, where he was a Researcher in image processing. From 2017 to 2018, he was with Smart Car Research Center, Chungbuk, South Korea, where he was a Researcher of the ADAS system. His research interests include deep learning, image processing, object detection, autonomous vehicle, and advanced driving assistance systems (ADAS).

TAE-HYOUNG PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Department of Control and Measurement Engineering, Seoul National University, South Korea, in 1988, 1990, and 1994, respectively. From 1994 to 1997, he worked as a Senior Researcher at the Precision Instrument Institute, Samsung Techwin Company Ltd. Since 1997, he has been a Professor with the School of Electronics Engineering, Chungbuk National University. From 2000 to 2001, he was a Visiting Scholar at the University of Toronto. His main research interests include electronic assembly and testing systems, robot motion planning, and optimization algorithms.

* * *