Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 3733799, 17 pages
https://doi.org/10.1155/2023/3733799

WILEY | Hindawi

Research Article

Fine-Grained Point Cloud Semantic Segmentation of Complex Railway Bridge Scenes from UAVs Using Improved DGCNN

Shi Qiu,[1,2,3] Xianhua Liu,[1,2,3] Jun Peng,[1,2,3] Weidong Wang,[1,2,3] Jin Wang,[1,2,3] Sicheng Wang,[1,2,3] Jianping Xiong,[4] and Wenbo Hu[5]

[1]*School of Civil Engineering, Central South University, Changsha 410075, China*
[2]*MOE Key Laboratory of Engineering Structures of Heavy-Haul Railway, Central South University, Changsha 410075, China*
[3]*Center for Railway Infrastructure Smart Monitoring and Management, Central South University, Changsha 410075, China*
[4]*Guangxi Transportation Science and Technology Group Co., Ltd., Nanning 530006, China*
[5]*Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China*

Correspondence should be addressed to Wenbo Hu; hwbphd@csu.edu.cn

Received 15 May 2023; Revised 4 September 2023; Accepted 21 September 2023; Published 3 October 2023

Academic Editor: Yi-Qing Ni

Automatic semantic segmentation of point clouds in railway bridge scenes is a crucial step in the digitization process and is required for a variety of subapplications including digital twin reconstruction and component geometric quality verification. This paper details a method for reliably and effectively segmenting point clouds acquired from complex railway bridge scenes by unmanned aerial vehicles (UAVs). The method involves segmenting seven common infrastructure elements in railway bridge point clouds using an improved DGCNN after processing low-quality point clouds from UAVs with a score-based denoising algorithm. The segmentation performance of the network is measured by averaging the intersection to union ratio between the segmentation results and the true labels of different elements, i.e., the mean intersection over union (mIoU). The proposed method is evaluated on three different scenes of railway bridges and achieved mIoU values of 99.18%, 90.76%, and 85.84%, respectively, at three levels of complexity ranging from easy to difficult. The results demonstrate that the proposed method captures the most discriminative features from low-quality point clouds, allowing for the accurate and efficient digital representation of railway bridge scenes.

1. Introduction

Over fifty percent of the high-speed railways constructed in China are comprised of bridges, making them an indispensable element of railway lines. Construction of bridges shortens routes, crosses terrain obstacles, and improves the smoothness of railway lines. To ensure the safety of railway traffic, it is necessary to establish digital models of railway bridge infrastructure. Due to their precision, comprehensiveness, and efficacy, point clouds have gradually become the dominant form of scene information representation as a result of the continuous development of information acquisition equipment and technology. Point cloud segmentation serves as the foundation for a variety of application domains, including building information modeling (BIM) reconstruction [1], geometric quality inspection [2, 3], and construction progress tracking [4].

For practical applications, rapidly obtaining point cloud data and accurately extracting key components from actual railway bridge scenes are essential. The terrains of railway bridges are typically complex, spanning valleys, and mountainous regions. Due to terrain limitations and railway maintenance windows, traditional point cloud acquisition equipment, such as vehicle-mounted LiDAR and stationary scanners, can only acquire partial point cloud information. However, UAVs offer unique advantages when acquiring point cloud data for railway bridges. They can be equipped with LiDAR or cameras for panoramic photography and the collection of point cloud data and multiple-view images. UAVs are flexible in operation, minimally constrained by terrain and railway maintenance windows, and have a broad data collection coverage, making them an effective solution for the rapid collection of railway bridge point cloud data.

Semantic segmentation of point clouds for railway scenes means assigning a classification value to each point's corresponding infrastructure object. Thus, all points in the point cloud that correspond to the same object type will receive the same classification value. This segmentation step enables the location of the various infrastructure objects. Most current railway scene point cloud segmentation relies on heuristic algorithms [5], including random sampling consensus (RANSAC) [6], region growing [7], and clustering algorithms based on normal vectors or intensity features [8, 9]. However, the segmentation performance of these algorithms is highly dependent on the designer-specified parameters, requiring a high level of a priori knowledge of the point cloud features. Moreover, heuristic algorithms are typically applied to point cloud types with a specific structure (e.g., point clouds with regular geometries such as lines, planes, and spheres). For the task of massive point cloud data segmentation, heuristic algorithms typically require an extensive number of costly iterative computations. In the past, researchers developed heuristic algorithms for railway scene point cloud segmentation applications using highly normalized and standardized characteristics. To classify railway cables, for example, the authors in [10] designed a RANSAC algorithm based on the height information of the cable relative to the rail structure and the horizontal distance of the cable relative to the mast, which relies on highly consistent point cloud structure distribution features. Reference [11] combined with survey data to achieve precise location and segmentation of rails in the point cloud by the railway gauge corner, but the accuracy is easily affected by the quality of the sampled data and the accuracy of the survey data. The authors in [12] proposed a heuristic method, which first voxelizes the point cloud scene to make the point cloud index regular and then designs rules to extract different railway elements based on a priori structural spatial distribution information. This simplified method requires a highly regular railway scene and is unsuitable for segmenting multiple elements in complex scenes.

With the increasing maturity of the application of deep learning technology in the field of railway infrastructure [13–15], deep learning-based point cloud segmentation methods can segment point cloud elements more efficiently than heuristic algorithms and show better adaptability to complex point cloud scenes. Deep learning-based segmentation methods can be divided into three categories based on the type of point cloud data employed: projection-based, voxelization-based, and point-based.

The projection-based method entails that the point cloud is converted to a 2D image by projection, and then, the image is segmented using convolutional neural network to obtain the classification result of each pixel of the image, and finally, the pixels are mapped to the actual points to complete the segmentation of the point cloud. In [16], the authors used surface projection to convert the point cloud into a pseudodistance image and proposed a network architecture called FarNet to extract the rail structure. This method, however, is not suitable for large-scale and multicategory point cloud segmentation as it focuses solely on the spatial information of the rail.

The voxelization-based methods have the additional operation of replacing the disordered points with a regular 3D grid of a given size before the segmentation operation as compared to other methods. In [17], a 3D convolutional neural network is used to segment the voxelized mesh, which labels the point cloud to some extent. However, there are many invalid meshes for a wide variety of point clouds, and the grid size has a significant effect on the segmentation accuracy and memory usage.

The point-based method dominates the current point cloud segmentation methods [13] because it directly utilizes the coordinate information (sometimes including color or intensity information) of the points, maximizing the use of point cloud data for feature extraction. In [18], the authors used two network architectures, PointNet [19] and KPConv [20], to perform segmentation of point clouds of railway tunnels. The segmentation results are more accurate and generalizable than their previous heuristic-based work [21]. However, the method only applies to tunneling scenes and has a limited number of classifications. Regarding the railway infrastructure, the literature [22] proposed a segmentation method based on PointNet++ [23]. The method adapts the network architecture to various point cloud scenes by adjusting the sampling point density and local feature radius. The results demonstrate that the method achieves great segmentation accuracy for better-quality point clouds acquired by mobile survey systems, but that the segmentation accuracy for lower-quality point clouds acquired by UAVs still needs to be improved.

Aiming at the problems mentioned previously, this paper proposes a deep learning-based method for point cloud segmentation. The method is suitable for complex railway bridge scenes and maintains high segmentation accuracy for low-quality point clouds acquired by UAVs. The main contributions of this paper are summarized as follows:

(i) A point cloud segmentation method based on deep learning is proposed for segmenting assets associated with complex railroad bridge scenes. The method maintains high accuracy and generalizability in three scenes with varying segmentation challenges.

(ii) The method involves an improved model based on DGCNN. In contrast, the improved model is more lightweight and employs a PE-EdgeConv module to enrich the ordering information of local features in the point cloud, consequently improving the sensitivity of the model to point cloud features.

(iii) The method includes a point cloud enhancement denoising strategy to deal with the low-quality point clouds acquired by UAVs. The strategy consists of two stages: window sliding denoising and contour enhancement denoising. The preprocessed point cloud has a higher point cloud density, more distinct contours, and nearly no visible scatters.

The rest of the paper is organized as follows. Section 2 presents a detailed description of the proposed specific method for segmenting the point cloud of complex railway bridge scenes captured by UAVs. Section 3 demonstrates the segmentation results of the proposed method under different conditions. The accuracy, efficiency, and generalizability of the proposed method are discussed in Section 4. Finally, Section 5 summarizes the results, limitations, and future perspectives of this study.

2. Methodology

The proposed method for segmenting point clouds of complex railway bridge scenes captured by UAVs consists of three parts: data acquisition, point cloud preprocessing, and semantic segmentation using the improved DGCNN. The overall workflow is shown in Figure 1.

2.1. Data Acquisition. The dataset used for the point cloud segmentation task consists of several sections of Chinese high-speed railway bridges captured by a UAV, with ballastless track serving as the mainline railway structure. The dataset contains three scenes with respective track lengths of 130 m, 523 m, and 313 m. A DJI FC6310R camera mounted on a DJI M300 RTK UAV captured the images at 30 m, 40 m, and 40 m above the ground, respectively. Each scene captures 480, 645, and 283 images, which are saved in JPG format with a resolution of 5472×3648 and includes latitude, longitude, and altitude information. As illustrated in Figure 2, the multiview images of different scenes are converted into point cloud data using the 3D reconstruction software DJI Terra (version 2.2.0.15) and saved in the LAS format.

In order to evaluate the generalizability of the segmentation capability of the proposed method, three different types of point cloud scenes are selected in this study. Track length, ballast or ballastless, roadbed or bridge, and the number of main lines are the primary distinctions. In addition, for each of the three scenes, the capture frequency (image density) and flight altitude of the UAVs are varied in order to control the difference in point cloud density. Less challenging the scene, the lower the flight altitude and the higher the capture frequency.

According to Table 1, the difficulty of the point cloud segmentation task increases from (a) to (c) for each of these three scenes. Scene (a) is a single-type railway bridge with the highest point cloud density; scene (b) has the longest route, with bridges and roadbeds alternating, including three bridge sections and two roadbed sections, and a moderate point cloud density; scene (c) has a medium-length route with two connecting lines on both sides of the main line, and the track structure of the connecting lines are ballasted track, with a high background complexity and a low point cloud density. In conclusion, the element types and point cloud density of these three railway bridge scenes pose a challenge to the accuracy and generalizability of the proposed method.

The objective of this study is to segment the elements in the point cloud scenes of various complex railway bridges, including cable, mast, rail, track bed, protective wall, guardrail, and cluster, for a total of 7 categories, as shown in the annotations in Figure 3(a). The raw point cloud data contain a large amount of background data unrelated to the segmented elements, which can affect the balance of the dataset. Therefore, the unrelated background data are removed from the scenes and the segmented elements are annotated. The 7 elements marked with different colors are shown in Figure 3(b).

2.2. Data Preprocessing. Although UAVs have significant advantages in terms of image acquisition speed and terrain insensitivity, the point cloud data obtained from multiview images are less dense and accurate, noisier, and have less distinct point cloud contours than data acquired by radar or laser scanners. For this reason, this paper proposes a point cloud preprocessing method for large-scale scenes of railway bridges. The method has two main purposes: on the one hand, the coordinate direction of the point cloud is adjusted by coordinate transformation, and the regular distribution information of the railway bridge point cloud is utilized to extract the point cloud blocks with certain physical significance; on the other hand, a point cloud enhancing denoising strategy is adopted to reduce the noise level and enhance the contour of the point cloud, so as to improve the quality of the point cloud.

2.2.1. Coordinate Transformation. The initial coordinate system obtained from the railway bridge point cloud is usually not suitable for describing the spatial position of the railway bridge, and thus the point cloud coordinate system needs to be adjusted. The adjustment aims to make the X and Y axis parallel to the width and length direction of the railway bridge, respectively, while the Z axis is parallel to the building height and does not need to be adjusted.

The key to coordinate system adjustment is to calculate the rotation and translation matrix P of the XoY plane. As shown in Figure 4(a), first, two points $A(x_1, y_1)$ and $B(x_2, y_2)$ are manually selected from the point cloud, disregarding the z-coordinates, with the vector $(x_2 - x_1, y_2 - y_1)$ directed parallel (or close to) the bridge direction. Then, the distance L between the two points is calculated. The coordinates of A and B in the new coordinate system are $A'(0,0)$ and $B'(0,L)$, satisfying the relationship in equation (1):

$$\begin{cases} L = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \\ P = \begin{pmatrix} \cos(\theta) & \sin(\theta) & a \\ -\sin(\theta) & \cos(\theta) & b \end{pmatrix}, \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} = P \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}, \\ \begin{pmatrix} 0 \\ L \end{pmatrix} = P \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}. \end{cases} \tag{1}$$

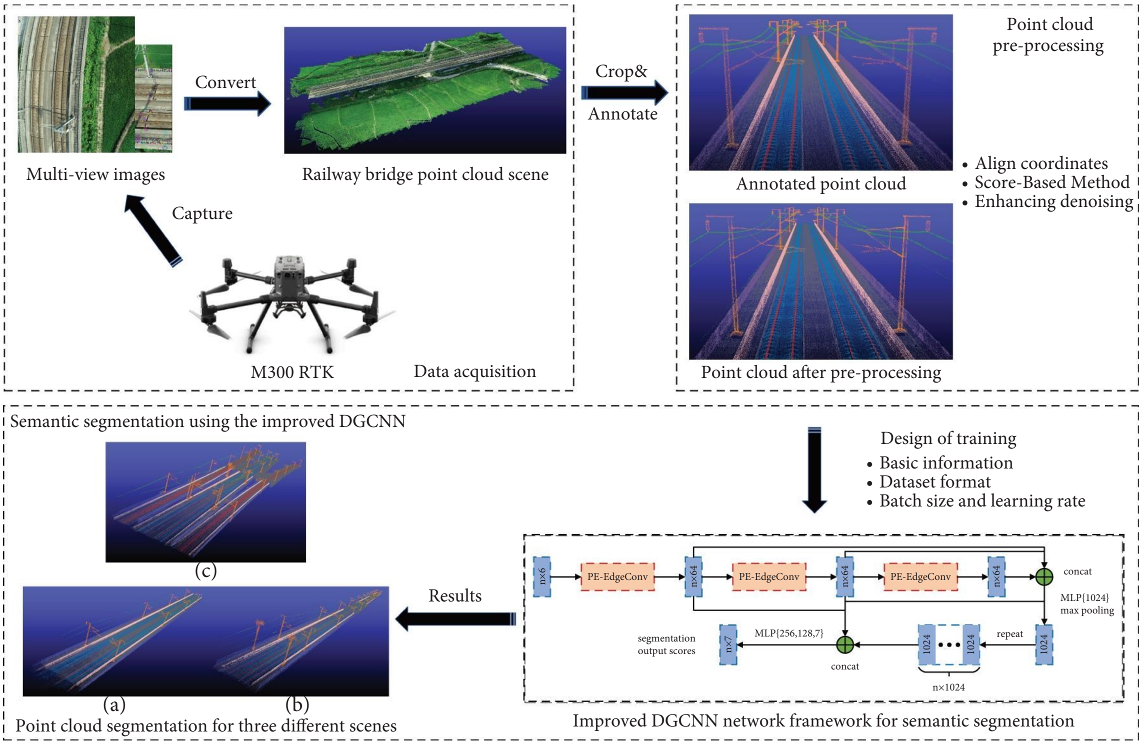


Figure 1: The workflow of the railway bridge point cloud segmentation method based on the improved DGCNN includes three parts: data acquisition, data preprocessing, and semantic segmentation using the improved DGCNN.

Here, θ, a, and b denote the rotation angle, $X-$axis translation value, and $Y-$axis translation value of the new coordinate system with respect to the original coordinate system, respectively. Solving the system of equations obtains the rotation and translation matrix P of the point cloud in the XoY plane, which is then applied to the initial point cloud to obtain the transformed coordinates x' and y', as shown in equation:

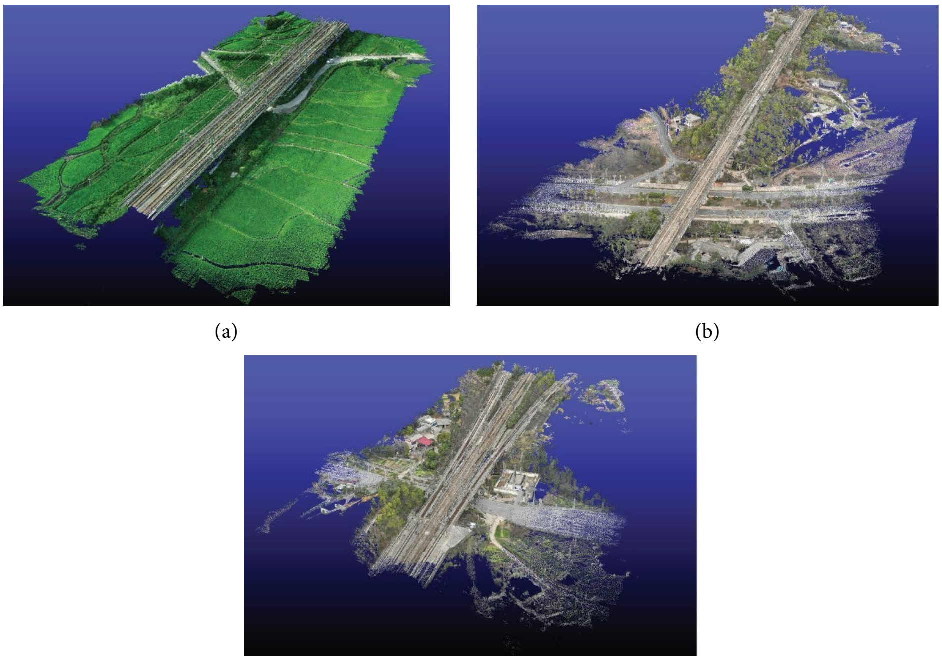
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = P \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{2}$$

2.2.2. Point Cloud Enhancing Denoising. Taking a cue from a state-of-the-art score-based point cloud denoising algorithm proposed in [24], this paper presents a strategy for enhancing denoising of point clouds. The algorithm considers the noisy point cloud as a convolution of noise-free samples with some noisy model and iteratively computes the gradient direction of the point positions to update the positions and accomplish denoising.

The gradient's direction is dependent on the noise model and noise-free samples. The default noise model is Gaussian noise, and the noise-free samples have an implicit estimate based on the point cloud's distribution. For a large-scale point cloud of a railway bridge scene, the point distribution is heterogeneous, which will affect the gradient direction of the points due to an inaccurate estimation of the noise-free samples. The point cloud can be denoised in blocks using the trick of window sliding denoising, which enables each small region of the point cloud to achieve a more effective denoising effect. A further feature of the algorithm is that the denoising process does not reduce the number of points; rather, it denoises by shifting the points to the surface of the estimated sample. Therefore, this characteristic of the algorithm can be used to up-sample the point cloud. This is accomplished by adding additional Gaussian noise to the point cloud after the first denoising of each small area, followed by a second denoising to improve the point cloud's contour. Examples of denoising results using different strategies are shown in Figure 5.

Compared to (a), (b) obtained after denoising by the original algorithm has denser points and clearer contours in Figure 5. However, because the algorithm is a direct denoising of a large-scale point cloud scene, the denoising detail is still lacking, and there are still some scatters around the contours of the point cloud. The window-sliding denoising trick effectively makes up for this deficiency, and (c) has significantly fewer distinct scatters than (b). After adding Gaussian noise to (c) for secondary denoising, the point density in (d) is obviously higher than in (c), and the contour of the point cloud is further clarified.

2.3. Network Architecture. Graph-based point cloud segmentation methods are the current mainstream of point-based methods [5]. DGCNN [25] is a representative network among the graph-based methods, and it contains the pioneering EdgeConv module. This module constructs a graph



(a)

(b)

(c)

Figure 2: Three scene point cloud data synthesized from multiview UAV images. (a) Scene of a simple segmentation task; (b) scene of a moderate segmentation task; (c) scene of a difficult segmentation task.

Table 1: Parameters of the three railway scenes.

Items	Scene (a)	Scene (b)	Scene (c)
Track length (m)	130	523	313
Flight altitude of UAVs (m)	30	40	40
Image density (images/m)	3.69	1.23	0.90
Ballastless or ballast	Ballastless	Ballastless	Both
Bridge or roadbed	Bridge	Both	Both
Number of railway lines	1	1	2

that dynamically represents the neighborhood relationship between points at various feature levels. The EdgeConv solves the problem of previous studies [19, 23] that focus only on the point features and ignore the relationships between points. EdgeConv incorporates the relationships between points into the point feature extraction process, making it highly suitable for extracting identical structural features. Many scholars have improved upon DGCNN to adapt it to various tasks, achieving significant results [26, 27]. In this study, the DGCNN architecture is improved to enhance its generalizability when dealing with the complex railway bridge scenes.

As shown in Figure 6, the improved DGCNN employs PE-EdgeConv for feature extraction, while multilayer perceptron (MLP) is used to increase and decrease feature dimensions. The representation of each point's features in multiple dimensions is obtained via shortcut connections. In the end, the extracted features include both the global and local features of the input point cloud. MLP is used to decode the features of each point, generating classification scores for each point belonging to different categories.

The main improvement of the DGCNN network framework is presenting a PE-EdgeConv module that contains Pos-Encoding, as shown in Figure 7. This module is similar to the original EdgeConv in that only the MLP layer contains trainable parameters, and the remaining steps include common feature processing operations such as k-nearest neighbors algorithm (k-NN), repeat, subtract, and pooling, which ultimately outputs a vector representation of the local features of each point in the point cloud. PE-EdgeConv differs from EdgeConv in that it includes the Pos-Encoding operation, which adds neighboring feature ordering information to the local feature vector expression, which more closely matches the actual point cloud's local relationships.

The original EdgeConv module dynamically calculates the k-neighborhood edge feature set for each point and uses it as a representation of the relationship between that point and other points. However, this representation does not account for the order of different point features in the edge feature set, resulting in the loss of ranking information for similarity. K-neighborhood sorting is based on the distance between neighborhood points and the central point in multidimensional features, reflecting the relationship between the point and the central point to a certain extent. This is described using the position encoding rules in Transformer [28] and Pos-Encoding to encode the ranking of various points in the edge feature set.

For the edge feature $E_k \in \mathbb{R}^{k\times f}$ of a single point, position encoding is performed according to the input order, as shown in equations (3) and (4):

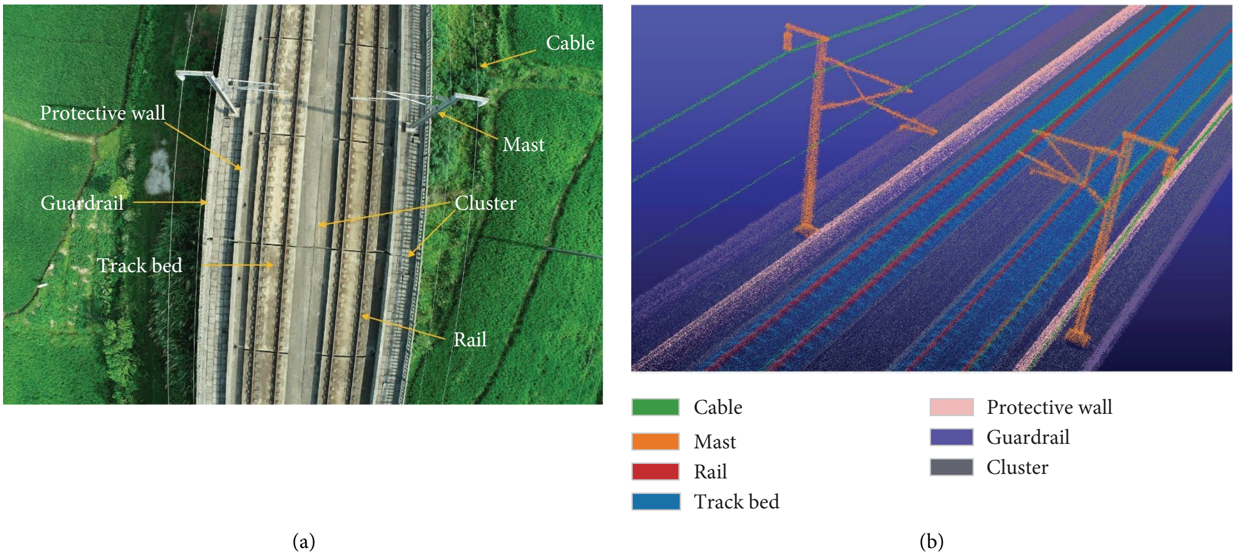


Figure 3: Schematic diagram of 7 segmentation elements. (a) Element annotation in real-world scenes; (b) element annotation in point cloud scenes. The different elements are defined as follows: cable refers to all visible cables; mast refers to all kinds of masts; rail refers to all railway tracks; track bed represents the collection of structures under the tracks; protective wall refers to the protective walls of the bridge section; guardrail refers to the guardrails on both sides of the bridge; cluster refers to the collection of points in the point cloud that do not have any semantic meaning.

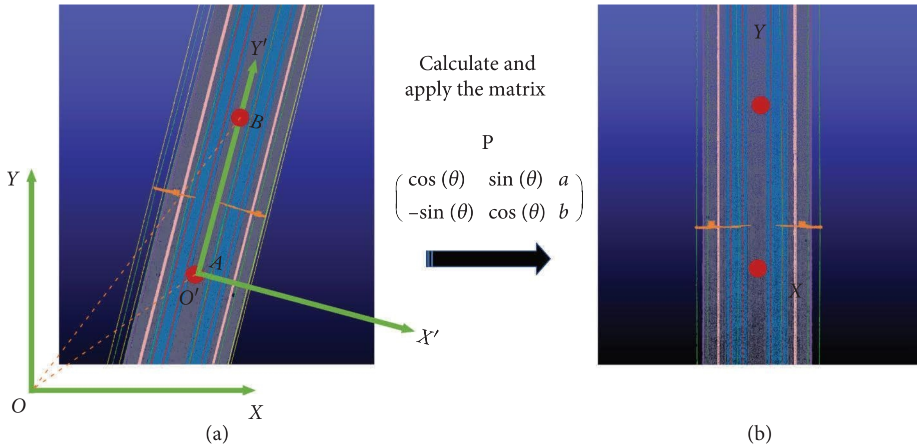

Y′
B
Y
A
O′
X′
O
X
Calculate and apply the matrix
P
$\begin{pmatrix} \cos(\theta) & \sin(\theta) & a \\ -\sin(\theta) & \cos(\theta) & b \end{pmatrix}$
Y
X
(a)
(b)

Figure 4: Coordinate transformation. (a) Before transformation; (b) after transformation.

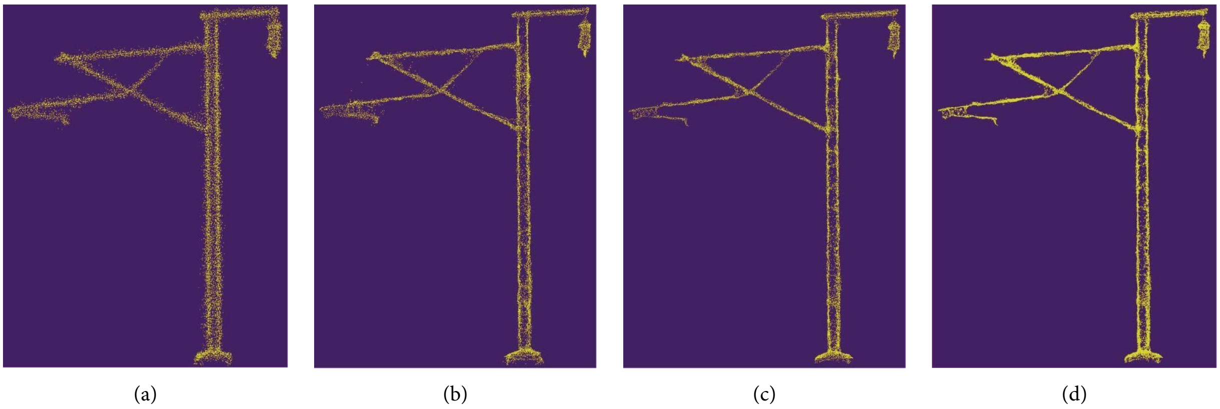


Figure 5: Denoising results under different methods. (a) Point cloud before denoising; (b) direct denoising results using the original algorithm; (c) applying window sliding denoising trick based on the original algorithm; and (d) add Gaussian noise to the result of (c) for secondary denoising to enhance the point cloud contours. The example in Figure 5 shows the result of adding 30% Gaussian noise and then secondary denoising.

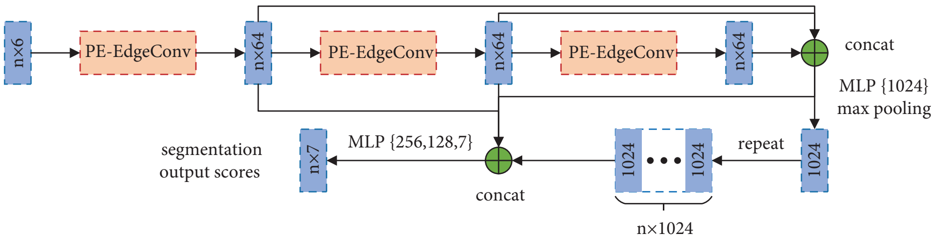
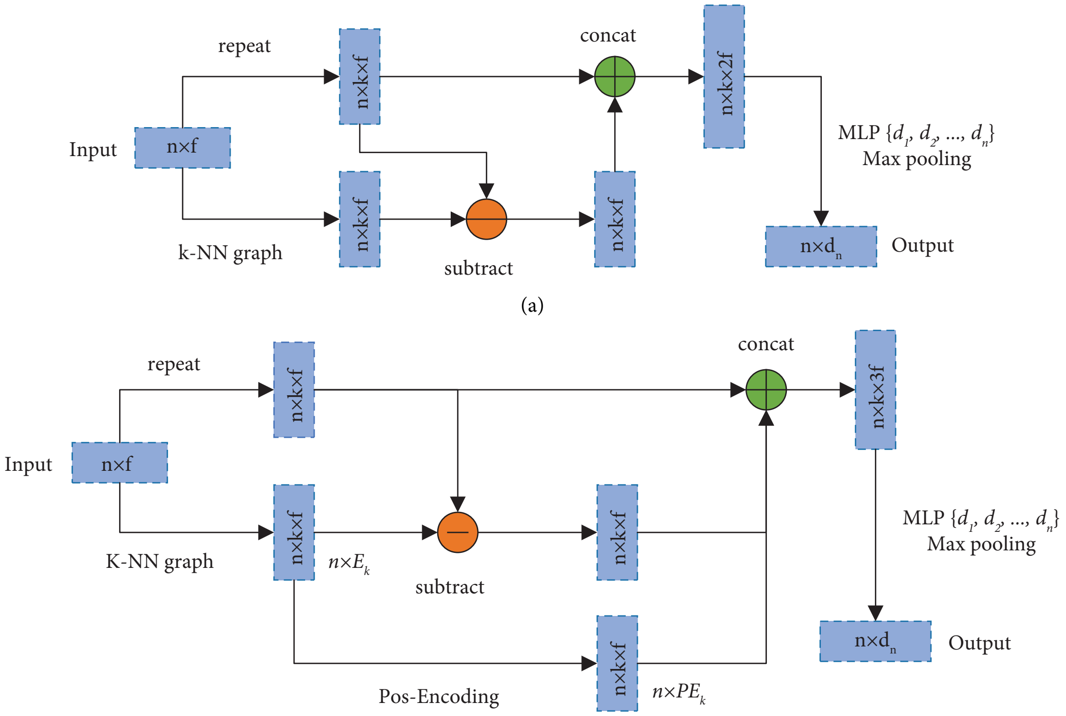


Figure 6: Improved DGCNN framework.



(a)

(b)

Figure 7: Schematic diagram of the two modules. (a) EdgeConv; (b) PE-EdgeConv.

$$\overrightarrow{p_t}^{(i)} = \begin{cases} \sin(\omega_i t), & \text{if } i = 2k, \\ \cos(\omega_i t), & \text{if } i = 2k+1, \end{cases} \tag{3}$$

$$\omega_i = \frac{1}{10000^{i/f}}. \tag{4}$$

In this case, t represents the order of the edge features, i denotes the encoding dimension, and f indicates the encoding length, which is the same as the input point feature length. After encoding, the position encoding of a single edge feature satisfies $p_t = [\sin(\omega_1 t), \cos(\omega_1 t), \ldots, \sin(\omega_{f/2} t), \cos(\omega_{f/2} t)]$, and the position encoding representation of a single point's edge feature set is $P_k \in \mathbb{R}^{k\times f}$. To utilize the ranking information of the edge feature set, the position encoding is multiplied by the edge feature, as shown in equation:

$$\mathrm{PE}_k = P_k \odot E_k. \tag{5}$$

Here, $\odot$ represents element-wise multiplication. By adding PE_k and E_k, the edge feature with both point encoding information and ranking information is obtained, which is used as a representation of the relationship between points.

2.4. Design of Training. This section presents information about network training, which includes hardware devices, point cloud scene division, data formats, learning rate settings, and some tricks for improving segmentation performance.

2.4.1. Basic Information. The DGCNN used in this study is developed based on the open-source framework PyTorch [29], which employs a modular design, allowing for convenient customization of functions to accomplish various study tasks. During model training, a computer equipped with an NVIDIA GeForce RTX 4090 GPU, an Intel Core i9-13900K CPU, and two 32 GB DDR4 memory modules is used for computation.

2.4.2. Scene Area Division. To facilitate the training and testing of the network, the areas of the three scenes are divided as shown in Figure 8. For scene (a), it is divided into four areas based on the length of the track line; for scene (b), it is divided into five areas based on various track foundation support structures; and for scene (c), it is divided into six areas based on three different track lines and various track foundation support structures. Different areas of the same scene have roughly equal numbers of point clouds, and at least two areas have the same number of types of segmentation elements. For k-fold cross-validation of a single scene, one of the areas is selected as the test set and the others as the training set.

2.4.3. Dataset Format. The point cloud obtained from the multiangle images collected by the UAV contains position and color information. To ensure generality, the point cloud segmentation in the railway bridge scene only considers the position information. Since the point cloud lacks topological information and is unorganized, this may not be conducive to the efficient operation of deep learning networks. Therefore, it is necessary to standardize the format of the input data. Considering the efficiency of computer memory and tensor operations, the input point cloud data are set to a multibatch and regularized format.

Input data standardization presents certain challenges. Referring to previous work [19], the input point cloud area is divided into B blocks, each of which is called a batch. Each batch contains N points, and each point has a dimension of C, which means the input point cloud representation format is (B, N, C). To ensure that the number of points in each block is equal, the number of points in each batch is set to 4096, and this is achieved by two operations, furthest point sampling (FPS) and repeat sampling (RS). The two operations correspond to cases where the number of points in each batch is more or less than the set value, respectively. Each point contains standard space coordinates (x, y, z) and global relative coordinates (x_g, y_g, z_g), where $x_g = x/x_m$, and x_m denote the maximum x coordinates of all points in an area. The meanings of y_g and z_g are similar.

2.4.4. Warm-Start Cosine Annealing. In the early stages of network training, a high learning rate can accelerate the decline of the loss function. To prevent oscillation of the loss function value, the learning rate must be decreased as the loss function approaches the global minimum. The warm-start cosine annealing strategy [30] can be used to effectively tune the learning rate, and the variation in the learning rate can be described by equation:

$$n_t = n_{\min} + \frac{1}{2}\left(n_{\max} - n_{\min}\right)\left(1 + \cos\left(\frac{T_{\mathrm{cur}}}{T_i}\pi\right)\right). \tag{6}$$

The relevant variables in equation (6) are explained as follows: n_t denotes the learning rate of the current epoch; $n_{\min}$ denotes the minimum learning rate; $n_{\max}$ denotes the maximum learning rate; T_{cur} denotes the epochs since the most recent warm restart; T_i denotes the epoch at the next learning rate restart. Here, T_i can be expressed by the following equation:

$$T_i = T_0 + T_{i-1}T_{\mathrm{mult}}, \quad i = 1, 2, \ldots, n. \tag{7}$$

In equation (7), T_i denotes the number of epochs in which the learning rate returns to the initial value for the $(i + 1)$ th time; T_0 denotes the number of epochs in which the learning rate first returns to its initial value; and T_{mult} denotes the restart factor of the learning rate, which controls the speed of change of the learning rate. In order to ensure that the learning rate will not restart again at the late stage of training, T_0 and T_{mult} need to be set in relation to the total epochs of training. In this study, the network is trained for 600 epochs to reach convergence. Setting T_0 to 5 and T_{mult} to 2 will satisfy the requirement.

As shown in Figure 9, the learning rate experiences multiple iterations during the total epochs, and the learning rate changes frequently in the early epochs, which enables the network's rapid convergence. In the late epochs, the learning rate changes slowly, which can help the network converge stably to the optimal value.

2.4.5. Other Tricks. Training and evaluation of the original DGCNN are based on public point cloud segmentation standard datasets such as ShapeNetPart [31] and S3DIS [32]. In comparison to these standard datasets, the railway bridge scenes point cloud is characterized by its broad range of point clouds, relatively uniform distribution of structures, and unique railway elements. These characteristics can be utilized in three primary ways to improve the results of point cloud segmentation:

(i) Reducing the number of decoding neurons.

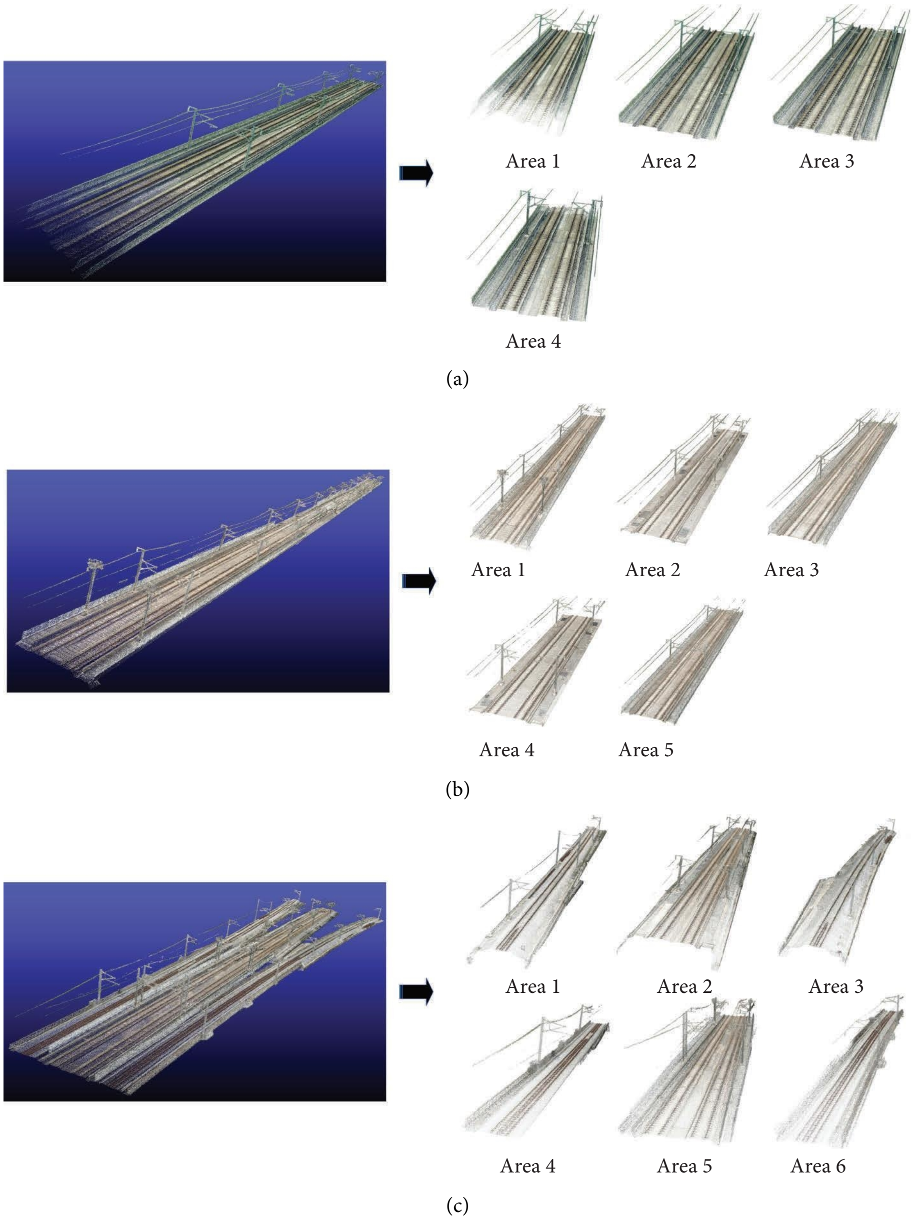
Reducing the number of decoding neurons makes the network model more lightweight. The point cloud segmentation of the railway bridge scene consists of 7 elements, which is less than the 16-element segmentation on ShapeNetPart and the 13-element segmentation on S3DIS; consequently, the decoding task of the network is simplified. Therefore, appropriately reducing the number of decoding neurons while ensuring the network has adequate decoding capability can improve the network's computational efficiency and enhance the lightweight of the network.

(ii) Adjust the orientation of the input point cloud blocks.

The majority of the structures observed in the railway bridge scene display a linear configuration and demonstrate symmetrical or parallel associations. In order to enhance the network's ability to extract point cloud features with greater precision, it is a logical strategy to align the orientation of the point cloud blocks with the linear structures.

(iii) Choosing an appropriate point cloud block size.

The larger the block size, the fewer batches in each epoch, which is equivalent to increasing the batch size. Typically, the block size is determined first, and then the batch size is determined by experiment. The appropriate block size can be



(a)

(b)

(c)

FIGURE 8: Area division of the three scenes. (a) Dividing four areas by the track length in scene (a); (b) dividing five areas by different track foundation support structures in scene (b); (c) dividing six areas by three distinct tracks and various track foundation support structures in scene (c).

selected according to the density of the point cloud. Identical to the standard dataset, the number of points within a block in this study is 4096. It is reasonable to set the block size to 2 m in order to make the actual number of points in a block comparable to 4096, given the point cloud density of the railway bridge scene in this study.

3. Results

This section presents the results of point cloud segmentation using the proposed method under various conditions. First, the metrics used for evaluating the performance of semantic segmentation are introduced. Next, the improvement in segmentation accuracy before and after point cloud denoising is compared. Then, the impact of network improvement and batch size setting are discussed on the experimental results. Subsequently, in order to verify the generalizability of the proposed method to point clouds in different railway bridge scenes, stratified K-fold cross-validation between different areas was conducted in three different scenes, respectively. Finally, the effect of the combination of datasets from different scenes on the point cloud segmentation task was evaluated.

In exception of the comparison experiments, the relevant parameters used in the proposed improved network are shown in Table 2.

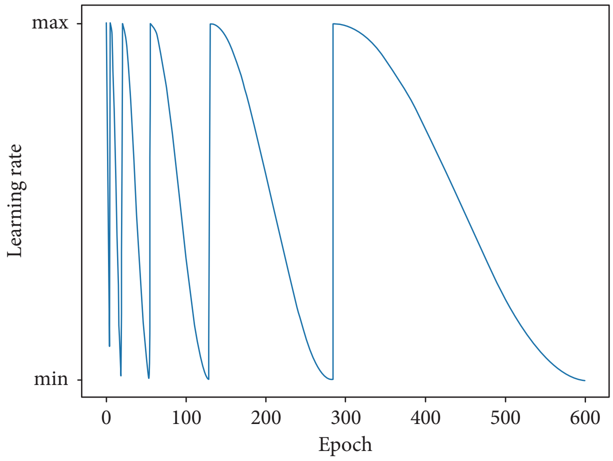


Figure 9: Curve of learning rate with epoch increase.

Table 2: Relevant parameters used in the proposed improved network.

Name	Variable	Value
Input feature	(x, y, z, x_g, y_g, z_g)	—
Block size	l	2 m
Epochs	—	600
Number of sampling points	N	4096
Batch size	BS	32
Range of learning rates	LR	(0, 0.002)
Learning rate fine-tuning strategy	Warm-start cosine annealing, $T_0 = 5$, $T_{\text{mult}} = 2$	
Optimizer	Adam [33]	

3.1. Metrics. As shown in Table 3, the number of points for each element is unbalanced. Using the accuracy metric on such an unbalanced dataset may produce misleading results [34]. To prevent prediction results from being biased towards the majority class in an imbalanced dataset, intersection over union (IoU) and mean IoU (mIoU), as well as balanced accuracy (bACC), are selected as performance metrics for point cloud segmentation. In order to facilitate the discussion of results, this paper focuses primarily on mIoU or IoU, presenting bACC as supplementary judgment in the results.

To facilitate comprehension, the symbols used in the following section are defined below: C is the total number of classes. TP (true positive) represents the number of samples that are correctly classified as positive. FN (false negative) represents the number of positive samples misclassified as negative. TN (true negative) represents the number of samples that are correctly classified as negative. FP (false positive) represents the number of negative samples misclassified as positive.

The abovementioned metrics are applicable to binary categorization tasks; for multicategorization tasks, each categorization element needs to be evaluated individually. For example, each categorization element can be simplified into two categories, the category of the element and the category of the nonelement, thus converting a multicategorization task into a binary categorization task. Combined with point cloud segmentation explained as follows: all elements have been labeled before segmentation, and in the segmentation result, for a particular segmented element C, the number of points correctly identified as element C is denoted as TP, and the number of points incorrectly identified as other elements is denoted as FN; for the other elements, the number of points identified as labels of other elements is denoted as TN, and the number of points incorrectly identified as element C is denoted as FP.

The evaluation metrics IoU, mIoU, and bACC are explained as follows:

(i) IoU

This metric is used to evaluate the segmentation performance of a single class, defined as the number of common points between the ground truth and predicted samples for the current class divided by the total number of points present in both samples, as shown in equation (8):

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{8}$$

(ii) mIoU

The mIoU is defined as the average IoU over all classes, and the network version that achieves the highest validation mIoU is saved as the best version, as shown in equation (9):

$$\text{mIoU} = \frac{1}{K}\sum_{i=1}^{K}\text{IoU}(i). \tag{9}$$

(iii) bACC

This metric is used to evaluate the overall segmentation performance of a point cloud scene. It is defined as the average recall rate across each class, as shown in the following equation:

$$\begin{cases} \text{recall}(i) = \dfrac{\text{TP}}{\text{TP} + \text{FN}}, \\ \\ \text{bACC} = \dfrac{1}{K}\displaystyle\sum_{i=1}^{K}\text{recall}(i). \end{cases} \tag{10}$$

3.2. Denoising Effect. This subsection evaluates the accuracy improvement of the proposed point cloud denoising method for the railway bridge point cloud segmentation task. In the denoising test for scene (b), the details of the denoised point cloud are shown in Figure 10. Compared to the point cloud before denoising (a), the denoised point cloud (b) has a higher density and more distinct shape contours. Compared to the point cloud without denoising, the network segmentation mIoU of the denoised point cloud is 90.76%, which is an improvement of 5.03%. This indicates that the proposed denoising method significantly improves the quality of the point cloud and is beneficial for the network to learn the point cloud features.

Table 3: The percentage of different classification elements present in each of the three scenes.

Scenes	Category (%)							Total points
	Cable	Cluster	Guardrail	Mast	Protective wall	Rail	Track bed	
(a)	5.12	31.28	16.11	3.59	8.59	7.94	27.01	3253843
(b)	3.59	27.07	15.99	4.45	11.71	12.61	24.58	4179984
(c)	4.23	39.06	11.77	8.09	7.28	14.96	14.61	2720482

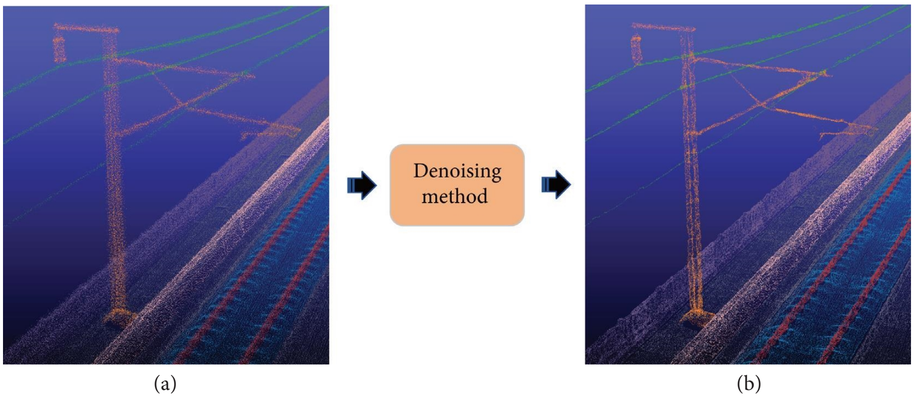


Figure 10: Comparison of the results before and after denoising. (a) Before denoising, the segmentation mIoU is 85.73%; (b) after denoising, the segmentation mIoU is 90.76%.

3.3. Network Improvement. To evaluate the improvement effect of the proposed method, a comparison experiment is designed between the proposed method and various networks. Scene (b) is the experimental test object, with area 3 serving as the test set and the remaining areas serving as the training set. The improved DGCNN is first compared to the original DGCNN. PE-EdgeConv and the decoding layer are the primary differences between the two networks. The experimental results are shown in Figure 11; after 600 epochs, both networks converge. Compared to DGCNN, the improved DGCNN has a faster loss decrease and a more stable loss during training, resulting in increased train and test accuracies. Since the learning rate is restarted and iteratively updated during the training process (see 2.4.4 for details), when the learning rate is abruptly increased, the update step size of the network parameters will also be abruptly increased, which ultimately reflects a large change in the classification prediction value of the elements, resulting in an obvious loss peak in Figure 11.

In addition, this paper compares the segmentation results of other representative state-of-the-art methods on this paper's dataset, including PointNet++, KPConv, Point Transformer, and Swin3D-L. The results are shown in Table 4. The segmentation mIoU of all three classical networks, PointNet++, DGCNN, and KPConv, is lower than the improved DGCNN proposed in this paper. Point Transformer uses the self-attentive layer for the 3D point cloud, which has sequence-independent properties and is thus suitable for extracting point cloud features. This network achieves the highest segmentation mIoU in the dataset of this paper. Due to the significant differences in point cloud quality and segmentation element types between the railway and indoor datasets, Swin3D-L as a state-of-the-art pretraining network for indoor scenes does not perform well on the railway scene dataset in this paper. Finally, the segmentation accuracy of the improved DGCNN proposed in this paper is slightly lower than Point Transformer; but compared with the original DGCNN, the segmentation accuracy is significantly improved, which indicates that the PE-EdgeConv module enriches the local feature ordering information of the point cloud and effectively improves the network's recognition of the point cloud.

Overall, the improved DGCNN in this paper's dataset achieves point cloud segmentation accuracy comparable to that of state-of-the-art networks. Furthermore, the improved DGCNN has a significant advantage in model weight, which facilitates the deployment of point cloud segmentation applications.

3.4. Impact of Batch Size. Batch size (BS) is a major hyperparameter that has a significant impact on the performance of neural networks. Studies [37, 38] have shown that increasing the BS can significantly reduce the network training time and improve the accuracy and stability of gradient descent. However, the generalizability of the network will decrease as the BS increases excessively [39]. In order to balance the training efficiency and generalizability of the network, it is necessary to select the appropriate BS through an experiment. The experiment compares the segmentation results of the improved DGCNN with four different BS settings, and area 3 in scene (b) is selected as the test set and the remaining areas serving as the train set. To ensure that the network can converge within 600 epochs with different settings, the learning rate is also adjusted proportionally to the BS.

As shown in Figure 12, the results indicate that as BS increases, the training time spent on each epoch decreases in a certain linear relationship, but the decrease is weakened.

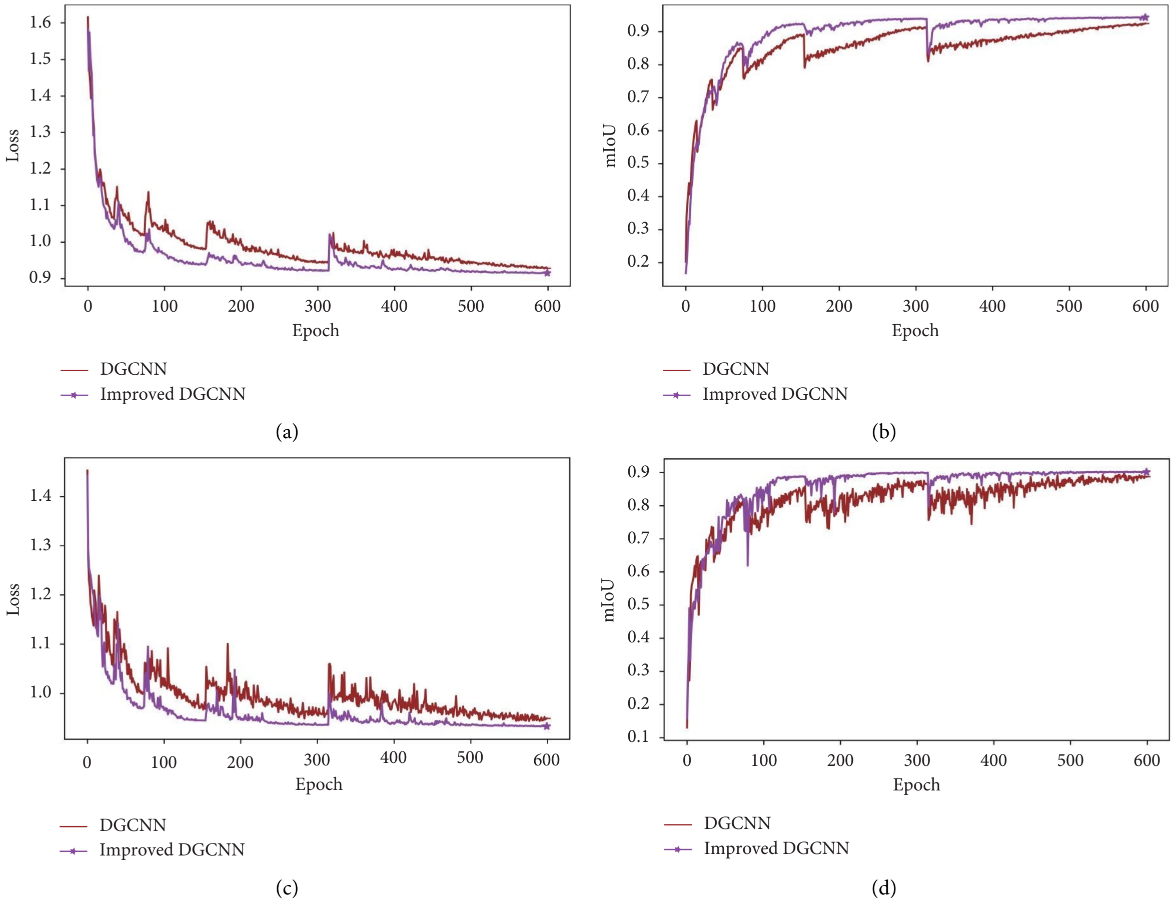


Figure 11: Training and testing results of both DGCNN and improved DGCNN. (a) Train loss; (b) train mIoU; (c) test loss; and (d) test mIoU.

Table 4: Segmentation results of different networks in scene (b).

Methods	Test mIoU (%)	Test bACC (%)	Parameters size (MB)
PointNet++ [23]	88.25	93.17	7.18
DGCNN [25]	88.96	93.67	2.45
KPConv [20]	89.36	94.04	14.10
Point transformer [35]	**90.92**	**95.53**	7.80
Swin3D-L [36]	83.87	89.68	60.75
Improved DGCNN	90.64	95.31	**2.12**

When BS is less than 32, the mIoU of training and testing increases as BS increases. When BS is greater than 32, increasing BS improves the training mIoU but decreases the testing mIoU, which indicates that the network enters an overfitting state. Taking into account the preceding analysis, it is beneficial for the network to set BS to 32 in order to achieve a balance between training efficiency and generalizability.

3.5. Cross-Validation of Individual Scenes. This section demonstrates the segmentation accuracy and generalizability of the proposed method in scenes (a), (b), and (c). Stratified K-fold cross-validation experiments are designed for different areas in each scene, as shown in Table 5. All types of elements in the three scenes achieve high segmentation accuracy, but there are significant differences. Among them, scene (a) has the highest segmentation accuracy, scene (b) follows, and scene (c) shows the lowest segmentation accuracy. This is linked to the predicted difficulty of the segmentation task for the three scenes and is closely related to the complexity of the scenes and the quality of the dataset. The point cloud segmentation results for each scene are shown in Figure 13.

The point cloud segmentation results for the three scenes are analyzed as follows:

(i) Scene (a)

In scene (a), the variety of structural types is relatively simple, and their arrangement is more regular, with high-quality point clouds. As a result, the segmentation accuracy of each structural element is

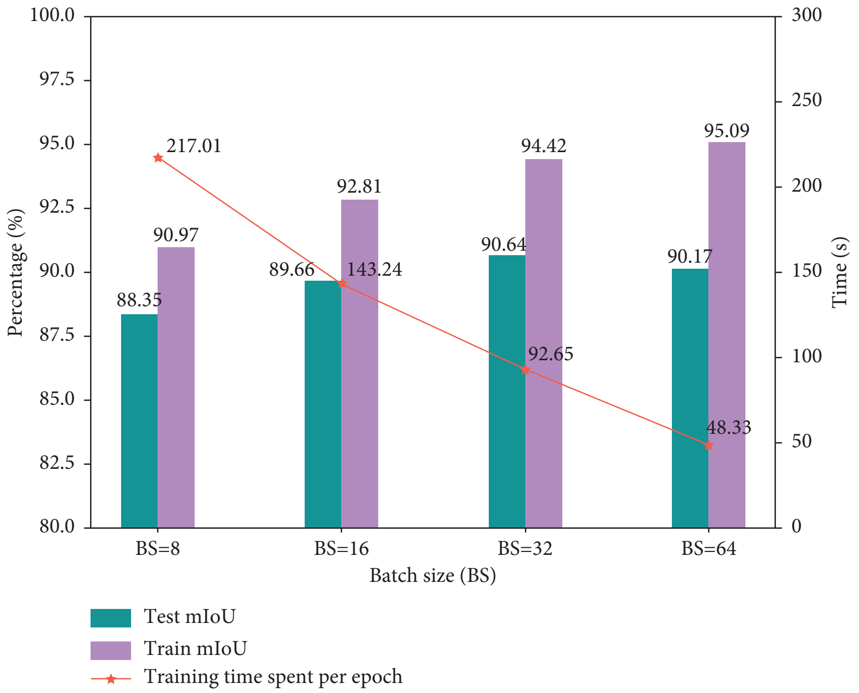


Figure 12: Point cloud segmentation results with four BS settings.

higher, and the mean IoU of individual elements in different areas reaches over 98%.

(ii) Scene (b)

In scene (b), the UAV was shot at a height of 40 meters, resulting in a lower point cloud density. The segmentation accuracy of each element is significantly lower compared to scene (a). The track beds in the bridge section are elevated, while the track beds in the roadbed section are aligned with the clusters. There are some differences between these two labeled track beds, resulting in a lower IoU value of the training network for the segmentation of the track beds. In addition, the rail is easily mistaken for track bed as it is closely connected to the track bed, resulting in unclear shape features.

(iii) Scene (c)

Due to the high complexity of scene (c) (see Section 2.1), the uniformity of the dataset is reduced and there are structural differences among the same elements. In addition to the structural differences in the roadbed mentioned in scene (b), the distribution of the masts is more complex, which further leads to large differences in segmentation accuracy between the elements. Although cable is similar in appearance to rail, their segmentation accuracy is maintained above 97% in all three scenes due to their unique positional characteristics as they are located above the track. Similarly, the guardrail is located on both sides of the rails and has a significant height difference, resulting in a higher segmentation accuracy.

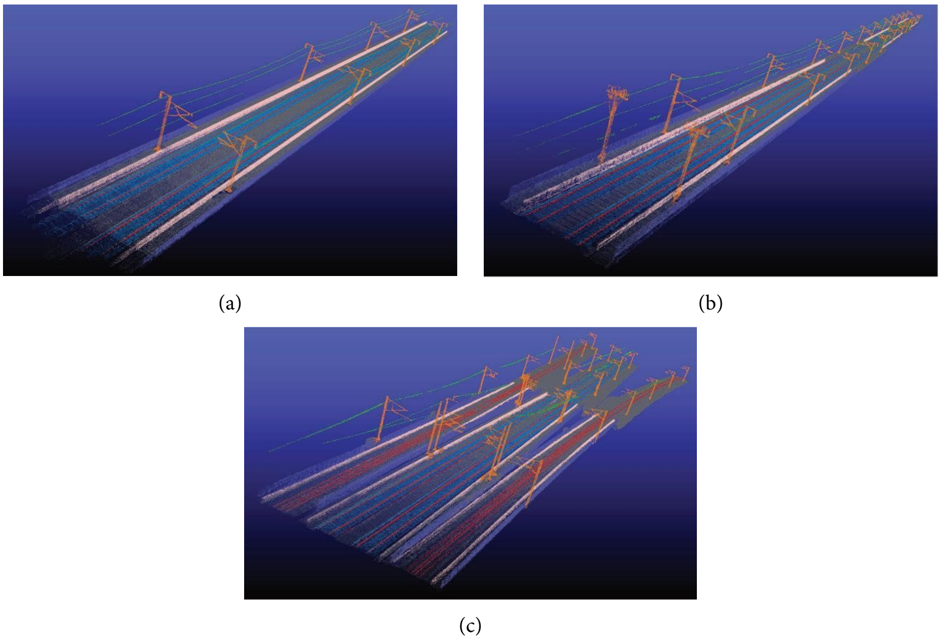
In conclusion, the point cloud segmentation mIoU of the proposed method can be maintained above 85% for all three scenes with various levels of difficulty, and the segmentation mIoU between different areas of a single scene is comparable, indicating good generalizability.

3.6. Testing on the Fusion Dataset. To verify the generalizability of the dataset, models trained on a single-scene dataset were applied to the segmentation tasks of the other two scenes. As shown in Table 6, due to the significant differences between the three scenes, the segmentation accuracy of the models trained on a single scene is low when applied to the other scenes. Furthermore, the model's segmentation accuracy is the lowest when trained on scene (a); when trained on scene (b) or scene (c), the accuracies are similar. This indirectly indicates that the positions and shapes of the elements in the different areas of the scene (a) are more uniform, which does not sufficiently train the model's generalizability. In contrast, the quantities and shapes of elements in the different areas of scenes (b) and (c) have some variations, which help improve the network's generalizability.

The network trained with a single scene dataset has low segmentation accuracy for point clouds from other scenes, as shown in Table 6. In order to address this issue, we designed a multiscene fusion dataset test experiment in which fifty percent of the point clouds from scenes (a), (b), and (c) are removed for multiple combination tests. The results demonstrate that the merged dataset contains more elements and data features, which can be utilized to train the network's generalizability in various scenes.

Table 5: Results of segmentation of three scenes in different areas.

Scenes	Test area	IoU (%)							mIoU (%)	bACC (%)
		Cable	Cluster	Guardrail	Mast	Protective wall	Rail	Track bed		
(a)	1	99.46	99.56	99.92	97.88	99.08	97.76	99.21	98.98	99.51
	2	99.61	99.71	99.96	98.43	99.51	98.37	99.32	99.27	99.68
	3	99.40	99.82	99.99	97.86	99.60	98.53	99.52	99.24	99.68
	4	99.10	99.58	99.95	97.90	99.42	99.10	99.45	99.21	99.63
	Mean	99.39	99.66	99.88	98.02	99.40	98.44	99.38	99.18	99.63
(b)	1	98.74	90.27	97.80	91.07	92.92	83.80	86.77	91.62	96.73
	2	98.44	88.31	—	92.97	—	82.83	87.63	90.04	95.15
	3	98.78	87.84	97.12	90.20	91.70	82.05	86.82	90.64	95.31
	4	98.56	88.49	—	92.42	—	84.80	85.86	90.03	95.07
	5	98.77	89.63	98.62	90.41	91.96	83.84	87.18	91.49	96.66
	Mean	98.66	88.91	97.85	91.41	92.19	83.46	86.85	90.76	95.78
(c)	1	97.50	84.86	—	89.45	—	74.29	—	86.53	92.11
	2	96.95	83.45	—	88.23	—	75.45	82.20	85.26	91.05
	3	97.04	83.39	—	88.91	—	75.84	—	86.30	91.77
	4	97.84	82.84	94.46	89.59	85.99	72.54	—	87.21	92.64
	5	96.38	81.55	93.80	86.76	83.95	73.53	82.64	85.52	91.26
	6	—	83.70	96.95	88.70	78.84	72.94	—	84.23	90.87
	Mean	97.14	83.30	95.07	88.61	82.93	74.10	82.42	85.84	91.62


(a)

(b)

(c)

Figure 13: Semantic segmentation results for three scenes. (a) Segmentation result of scene (a); (b) segmentation result of scene (b); and (c) segmentation result of scene (c).

4. Discussion

In this section, the proposed method will be discussed in terms of accuracy, efficiency, and generalizability with experimental results.

4.1. Accuracy and Efficiency. To improve the accuracy of the network's segmentation, we focused on three aspects. First, in order to improve the quality of the point cloud, we adopted a large-scale global point cloud denoising strategy. Also, the secondary denoising is performed by increasing the number of points in order to increase the point cloud density and shape contour. In general, as the number of points increases, the contour of the point cloud becomes more distinct. However, after reaching a certain level (about 30% increase), it becomes difficult to improve the segmentation accuracy of the point cloud and instead increases the computational burden of point cloud block sampling. Second, considering that the original DGCNN lacks the use of edge feature similarity ordering information, we proposed PE-EdgeConv, which stores edge feature ordering information by referencing the position encoding in [28]. This allows the network to further learn the depth space distribution features of the point cloud. Finally, we adjusted the

Table 6: Segmentation tests of point cloud scenes with different data sets.

Training scenes	Test scene	mIoU (%)	bACC (%)
(a)	(b)	25.40	42.15
	(c)	22.63	36.78
(b)	(a)	58.54	65.26
	(c)	42.99	49.65
(c)	(a)	62.49	71.35
	(b)	46.24	52.42
(a) and (b)	(c)	56.32	67.24
(a) and (c)	(b)	68.44	76.91
(b) and (c)	(a)	83.21	90.31
(a), (b), and (c)	(a)	98.65	99.03
	(b)	91.20	95.34
	(c)	77.87	86.06

size and orientation of the input point cloud blocks according to theoretical and practical application scenes and reduce the number of neurons in the decoding layer of the network. This effectively improves the efficiency of network training. It should be noted that the larger the point cloud block is, the number of sampling points needs to be increased proportionally to avoid losing point cloud accuracy. However, as the number of points increases, the memory consumption for computing and storing the k-nearest neighbor set grows rapidly [40].

4.2. Generalizability. The proposed method exhibits similar point cloud segmentation accuracy across various areas within the same scene, demonstrating generalizability. Therefore, when dealing with similar sections of the same route, it is possible to consider using datasets from some areas for training and then segmenting the remaining areas. This method can reduce the data and time costs of network training and enhance the feasibility of point cloud segmentation in practical railway bridge applications. However, for scenes of different routes, the scene elements normally differ significantly, and the segmentation accuracy is relatively lower when the trained model is directly applied (see Table 6). We believe that the low accuracy is due to the insufficient richness of the model's training set for other scenes. To verify this hypothesis, we constructed a simple fusion dataset to train the network, and the results demonstrate a significant increase in the segmentation accuracy of the different scenes. In order to improve the generalizability of the network in different scenes, it will be necessary for future applications to collect point cloud data from a variety of scenes as a training set for the network. In addition, the incorporation of multisource data, such as images and LiDAR [41], can be considered to improve segmentation precision.

5. Conclusions

This paper proposes a point cloud segmentation method based on improved DGCNN. The method is used to accomplish the complete process of automatically segmenting the relevant elements of railway bridges from the point clouds synthesized from multiview images of UAV. The relevant elements include cable, mast, rail, track bed, protective wall, guardrail, and cluster. The proposed method is improved in several aspects such as point cloud quality, network architecture, data input, and training details, which effectively improves the accuracy and generalizability of the network for point cloud segmentation.

The proposed method achieves 99.18%, 90.76% and 85.84% segmentation mIoU in three different complexity scenes from low to high. The segmented elements clearly display the structural composition and location information of the railway bridge, and the segmented results can be used as a building information model (BIM) basis for the railway digital twin, providing a feasible solution for related railway applications including asset management, geometric quality inspection, and construction progress tracking.

The point cloud segmentation method proposed in this paper shows good generalizability for different railway bridge scenes; however, the time-consuming and laborious acquisition and manual labeling of the required training sets make it difficult to be rapidly deployed for actual inspections. Physics-based virtual models are urgently needed to be developed for synthesizing standardized training sets of railway scenes with richer elements, driving more efficient deployment of deep learning methods. In addition, transforming supervised DGCNN to semisupervised or even unsupervised methods is also an effective improvement direction for reducing the cost of manual labeling.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. U1734208), the National Natural Science Foundation of China (Grant no. 52178442), and the Hong Kong Polytechnic University's Postdoc Matching Fund Scheme (Grant no. 1-W29R).

References

[1] M. Bassier and M. Vergauwen, "Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data," *Automation in Construction*, vol. 120, Article ID 103338, 2020.

[2] K. Mirzaei, M. Arashpour, E. Asadi, H. Masoumi, A. Mahdiyar, and V. Gonzalez, "End-to-end point cloud-based segmentation of building members for automating dimensional quality control," *Advanced Engineering Informatics*, vol. 55, Article ID 101878, 2023.

[3] V. Kasireddy and B. Akinci, "Assessing the impact of 3D point neighborhood size selection on unsupervised spall classification with 3D bridge point clouds," *Advanced Engineering Informatics*, vol. 52, Article ID 101624, 2022.

[4] R. Maalek, D. D. Lichti, and J. Y. Ruwanpura, "Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction," *Remote Sensing*, vol. 11, no. 9, p. 1102, 2019.

[5] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: a review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, 2020.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.

[8] J. Hernández and B. Marcotegui, "Filtering of artifacts and pavement segmentation from mobile lidar data," in *Proceedings of the ISPRS Workshop Laserscanning 2009*, Paris, France, September 2009.

[9] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical Pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[10] M. Ariyachandra and I. Brilakis, "Digital twinning of railway overhead line equipment from airborne lidar data," in *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, Kitakyushu, Japan, July 2020.

[11] A. Karunathilake, R. Honma, and Y. Niina, "Self-organized model fitting method for railway structures monitoring using lidar point cloud," *Remote Sensing*, vol. 12, no. 22, p. 3702, 2020.

[12] D. Lamas, M. Soilán, J. Grandío, and B. Riveiro, "Automatic point cloud semantic segmentation of complex railway environments," *Remote Sensing*, vol. 13, no. 12, p. 2332, 2021.

[13] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.

[14] W. Hu, W. Wang, C. Ai et al., "Machine vision-based surface crack analysis for transportation infrastructure," *Automation in Construction*, vol. 132, Article ID 103973, 2021.

[15] W. Wang, W. Hu, W. Wang et al., "Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network," *Automation in Construction*, vol. 124, Article ID 103484, 2021.

[16] Z. Wang, G. Yu, P. Chen, B. Zhou, and S. Yang, "FarNet: an attention-aggregation network for long-range rail track point cloud segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13118–13126, 2022.

[17] J. Huang and S. You, "Point cloud labeling using 3D convolutional neural network," in *Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, Cancun, Mexico, January 2016.

[18] M. Soilán, A. Nóvoa, A. Sánchez-Rodríguez, B. Riveiro, and P. Arias, "Semantic segmentation of point clouds with pointnet and kpconv architectures applied to railway tunnels," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2-2020, pp. 281–288, 2020.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Piscataway, NJ, USA, January 2017.

[20] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: flexible and deformable convolution for point clouds," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Seoul, Korea (South), December 2019.

[21] A. Sánchez-Rodríguez, B. Riveiro, M. Soilán, and L. M. González-deSantos, "Automated detection and decomposition of railway tunnels from mobile laser scanning datasets," *Automation in Construction*, vol. 96, pp. 171–179, 2018.

[22] J. Grandio, B. Riveiro, M. Soilán, and P. Arias, "Point cloud semantic segmentation of complex railway environments using deep learning," *Automation in Construction*, vol. 141, Article ID 104425, 2022.

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc, New York, NY, USA, 2017.

[24] S. Luo and W. Hu, "Score-based point cloud denoising: 10," 2021, https://arxiv.org/abs/2107.10981.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[26] E. Widyaningrum, Q. Bai, M. K. Fajari, and R. C. Lindenbergh, "Airborne laser scanning point cloud classification using the DGCNN deep learning method," *Remote Sensing*, vol. 13, no. 5, p. 859, 2021.

[27] R. Pierdicca, M. Paolanti, F. Matrone et al., "Point cloud semantic segmentation using a deep learning framework for cultural heritage," *Remote Sensing*, vol. 12, no. 6, p. 1005, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[29] A. Paszke, S. Gross, F. Massa et al., "Pytorch: an imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] I. Loshchilov and F. Hutter, "Sgdr: stochastic gradient descent with warm restarts," 2016, https://arxiv.org/abs/1608.03983.

[31] A. X. Chang, T. Funkhouser, L. Guibas et al., "ShapeNet: an information-rich 3D model repository," 2015, https://arxiv.org/abs/1512.03012.

[32] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," 2017, https://arxiv.org/abs/1702.01105.

[33] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[34] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, August 2010.

[35] M. H. Guo, J. X. Cai, Z. N. Liu, T. J. Mu, R. R. Martin, and S. M. Hu, "PCT: point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[36] Y. Q. Yang, Y. X. Guo, J. Y. Xiong et al., "Swin3D: a pretrained transformer backbone for 3D indoor scene understanding," 2023, https://arxiv.org/abs/2304.06906.

[37] S. L. Smith, P. J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," 2017, https://arxiv.org/abs/1711.00489.

[38] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," 2018, https://arxiv.org/abs/1803.09820.

[39] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc, New York, NY, USA, 2017.

[40] N. Bhatia, "Survey of nearest neighbor techniques," 2010, https://arxiv.org/abs/1007.0085.

[41] Y. Geng, F. Pan, L. Jia et al., "UAV-LiDAR-Based measuring framework for height and stagger of high-speed railway contact wire," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7587–7600, 2022.