

Article

# Processing Laser Point Cloud in Fully Mechanized Mining Face Based on DGCNN

Zhizhong Xing<sup>1</sup> , Shuanfeng Zhao<sup>1,\*</sup> , Wei Guo<sup>1</sup>, Xiaojun Guo<sup>2</sup> and Yuan Wang<sup>1</sup>

<sup>1</sup> College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; 17101016008@stu.xust.edu.cn (Z.X.); guow@xust.edu.cn (W.G.); wangy1212@xust.edu.cn (Y.W.)

<sup>2</sup> School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China; 202011000572@mail.scut.edu.cn

\* Correspondence: zsf@xust.edu.cn

**Abstract:** Point cloud data can accurately and intuitively reflect the spatial relationship between the coal wall and underground fully mechanized mining equipment. However, the indirect method of point cloud feature extraction based on deep neural networks will lose some of the spatial information of the point cloud, while the direct method will lose some of the local information of the point cloud. Therefore, we propose the use of dynamic graph convolution neural network (DGCNN) to extract the geometric features of the sphere in the point cloud of the fully mechanized mining face (FMMF) in order to obtain the position of the sphere (marker) in the point cloud of the FMMF, thus providing a direct basis for the subsequent transformation of the FMMF coordinates to the national geodetic coordinates with the sphere as the intermediate medium. Firstly, we completed the production of a diversity sphere point cloud (training set) and an FMMF point cloud (test set). Secondly, we further improved the DGCNN to enhance the effect of extracting the geometric features of the sphere in the FMMF. Finally, we compared the effect of the improved DGCNN with that of PointNet and PointNet++. The results show the correctness and feasibility of using DGCNN to extract the geometric features of point clouds in the FMMF and provide a new method for the feature extraction of point clouds in the FMMF. At the same time, the results provide a direct early guarantee for analyzing the point cloud data of the FMMF under the national geodetic coordinate system in the future. This can provide an effective basis for the straightening and inclining adjustment of scraper conveyors, and it is of great significance for the transparent, unmanned, and intelligent mining of the FMMF.

**Keywords:** point cloud; deep learning; feature extraction; fully mechanized mining face; machine learning; DGCNN; artificial intelligence; edge convolution



**Citation:** Xing, Z.; Zhao, S.; Guo, W.; Guo, X.; Wang, Y. Processing Laser Point Cloud in Fully Mechanized Mining Face Based on DGCNN. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 482. <https://doi.org/10.3390/ijgi10070482>

Academic Editors: Claudio Vanneschi and Matthew Eyre

Received: 18 May 2021

Accepted: 10 July 2021

Published: 13 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

At present, the processing technology of three-dimensional point cloud data is being applied more and more frequently in the mining field [1–3]. The national robotics engineering center of the United States can successfully draw a high-precision, three-dimensional map of underground roadways using the point cloud data obtained by a three-dimensional laser scanner and then propose an intelligent mining mode based on the three-dimensional map [4]. Using three-dimensional point cloud data to describe and draw the whole FMMF can accurately and intuitively reflect the spatial position relationship between the coal wall and fully mechanized mining equipment [5–8]. This can provide the direction information of a scraper conveyor so as to adjust the displacement of the hydraulic support in a timely manner. Therefore, we arranged markers in the FMMF and then found the markers in the point cloud of the FMMF (in this study, we arranged the sphere under hydraulic support in the FMMF and used the sphere as the marker). In subsequent research, we took the marker as the intermediate medium to transform the point cloud data of the FMMF to the national geodetic coordinate system for analysis. This is not only of great significance

for the straightening and inclining of the scraper conveyor and improving the mining efficiency and safety of coal mines and other mines, but it is also an important means to realize intelligent and unmanned mining in the mining field.

Compared with the two-dimensional image feature extraction method, the three-dimensional point cloud feature extraction method is a recent innovation. How to realize an efficient and robust point cloud geometric feature extraction algorithm has been a hot issue in this field in recent years [9–14]. Many scholars have extracted the geometric features of point clouds through traditional methods. For instance, Su et al. [15] proposed an adaptive Hilbert curve insertion algorithm with quasi linear time to improve the efficiency of point cloud triangulation. Liu et al. [16] proposed a new Delaunay triangulation algorithm for point clouds that divides the point cloud into triangular elements and derives inactive triangulation. Zhao et al. [17] proposed a method to quickly extract the features of point clouds based on gridding. This method first establishes a virtual model of the point cloud, then enhances the normal features of the point cloud and uses fast Fourier transform (FFT) to calculate the diffraction of point clouds on the grid. This method has the advantages of high speed and low memory consumption. Dey et al. [18] believed that the Voronoi cell of the point cloud adjacent to the point reflects the local geometric characteristics of the center point, and the Voronoi cell at the boundary feature of the point cloud is plate-shaped. In addition, the Voronoi cell at the intersection of the surface patches is spherical, and the Voronoi cell on the smooth surface is rod-shaped, so the point cloud features can be extracted from the geometric shape presented by the Voronoi cell.

However, the geometric feature extraction of point clouds based on triangulation is a time-consuming process and vulnerable to noise. Most importantly, the triangular patch cannot conform to the real surface topology of point clouds [19,20]. Although the method of gridding point clouds does not need triangulation, it is a simple and fast method. However, the grid generally contains a large amount of point cloud data, and as the grid is used as the basic unit to extract point cloud features, some feature points can be omitted [21,22]. Extracting point cloud features based on the Voronoi method requires the generation of a local Voronoi diagram, and the deviation of the eigenvalue ratio of different point cloud models is large. Therefore, it is difficult to set a threshold value for the general ratio in Voronoi diagrams [23].

In recent years, deep learning has become a hot research topic in the field of computer vision [24–28]. According to the data type of the deep neural network (DNN), the existing point cloud feature extraction methods can be divided into indirect methods and direct methods. Using an indirect method, Hu et al. [29] extracted the features of a LiDAR point cloud using convolutional neural network (CNN). Firstly, the neighborhood of each point was divided into several meshes so as to realize the projection of the three-dimensional point cloud onto a two-dimensional plane. Then, the maximum, minimum, and average elevations of all points in each grid were calculated as the eigenvalues of the three channels (red, green, blue) in order to generate a feature map for each point and train a CNN with a large number of labeled data sets. Zhao et al. [30] used a method that involved generating a multi-scale point cloud feature map and designing a multi-scale CNN to extract the features of the point cloud. After using multi-scale CNN to obtain the probability distribution of the point cloud, a decision tree was used to optimize the results, which further improved the accuracy of the network. Politz et al. [31] generated an elevation map by combining an airborne point cloud with a dense matching point cloud and then directly used U-Net to segment the elevation map under different parameter settings so as to explore the feature extraction performance of U-Net for these two types of point cloud. Qi et al. [32] of Stanford University proposed the deep neural network PointNet, which directly takes a three-dimensional point cloud as the input. PointNet can directly process the original point cloud data without preprocessing the point cloud data, as in other traditional deep neural network models (such as CNN). PointNet can achieve permutation invariance by processing each point independently. It uses multi-layer perceptron (MLP) to enhance the feature dimensions of each point, conducts max pooling as a symmetric function to fuse

the features of each point, and finally outputs the global feature vector to complete the recognition task.

However, the indirect methods of using deep neural networks to process point clouds all need to transform the point cloud first. This transformation will inevitably lose some of the spatial information of the point cloud, meaning that the deep neural network cannot learn the characteristics of the point cloud well and cannot distinguish different point clouds clearly, which will affect the classification, segmentation, and feature extraction of point clouds in different application environments [33]. Although PointNet can directly process the geometric features of the point cloud, it cannot capture the local information in the point cloud model because it only extracts the features of independent points, leading to the network having a poor ability to extract features from the point cloud, which also affects the generalization ability of the network [34]. DGCNN is based on the graph neural network (GNN) with the help of the concept of CNN in deep learning. It uses edge convolution to process the data on the graph structure and has seen broad application in geological engineering, environmental engineering, chemical engineering, etc. [35–37]. DGCNN does not need to convert the graph structure data into low-dimensional continuous space vector; rather, it can directly take the whole graph structure as an input (the graph structure in this study is the point cloud). In addition, the edge convolution operation in DGCNN can retain the local feature information of the point cloud data and ensure the exchange invariance and rotation invariance of its structure so as to extract the feature information of the point cloud data more efficiently and realize the corresponding feature expression. Due to the above reasons, we propose the use of DGCNN to extract the geometric features of point clouds in the FMMF.

In addition, by reviewing the existing reports in the literature, we found that no study has been conducted on the use of DGCNN to process the point clouds of the FMMF. Moreover, from the existing literature [38–40], we know that DGCNN can effectively learn the characteristics of point clouds of bridges and sites of cultural heritage so as to process these point clouds. However, whether the DGCNN built by the author's research group can be applied in the fields of mining engineering and geological engineering and how well it performs in extracting geometric features from the point clouds of the FMMF still need to be explored and studied. Therefore, in this research, the spheres in the point clouds of the FMMF are extracted by DGCNN. This provides the basis for the subsequent coordinate transformation of FMMF to geodetic coordinates using the sphere as the intermediate medium and then for the straightening and inclining adjustment of the scraper conveyor. At the same time, this is important for the intelligent and unmanned mining of the FMMF. Additionally, this study provides a reference for the application of DGCNN in energy- and geology-related fields.

The contributions of our work can be summarized as follows:

(1) In order to ensure that the DGCNN fully learns the characteristics of spherical point cloud and improve the robustness of the DGCNN, we added different amounts of noise to the point cloud of the complete sphere and incomplete sphere to produce a data set of a diverse sphere point cloud. Our results fill an existing research gap in that there is currently no data set in the depth neural network model for extracting the geometric features of point clouds.

(2) According to the characteristics of the point cloud, we propose a method to extract the geometric features of the point cloud based on DGCNN to obtain the position of the target sphere in the FMMF. This study provides a new method for the feature extraction of point clouds in the FMMF and provides the basis for the subsequent coordinate transformation of the FMMF to geodetic coordinates with the sphere as the intermediate medium as well as providing the basis for the straightening and inclining adjustment of a scraper conveyor.

(3) In DGCNN with a multi-layer edge convolution layer, the neighborhood information extracted by edge convolution can potentially represent a very long distance in the original space, but too deep an edge convolution layer will cause overfitting, which will

affect the feature extraction performance of the DGCNN. We define the best number of edge convolution layers for DGCNN to process the point cloud of the FMMF. Additionally, we use the Adam algorithm to improve the DGCNN to solve the problem that neural networks are likely to use the local optimal solution. The research results not only reduce the position error of the sphere, but also improve the adaptability, generalization ability, and practical value of the DGCNN model.

(4) We compared the effect of the improved DGCNN with PointNet and PointNet++ in extracting the geometric features of point clouds from the FMMF using the evaluation indexes, such as sphere position error. The results show that the improved DGCNN performs better in all types of evaluation. These results provide a direct early guarantee for the follow-up analysis of the point cloud data of the FMMF under the national geodetic coordinate system, which can provide an effective basis for the command and control of coal mine production. This is of great significance for the transparent, unmanned, and intelligent mining of the FMMF.

The remainder of the paper is structured as follows. Section 2 details the methods used in the work, including the method of producing a diversity sphere point cloud, the optimization algorithm of the DGCNN, the edge convolution method of the point cloud, and the method for finding markers in the point cloud of the FMMF. Section 3 details the experiment, including a description of the data set, the experimental details, the experimental results, and a discussion. Section 4 presents the conclusion.

## 2. Methods

This study aims to fill the research gap in the extraction of geometric features in the point cloud of the FMMF in the field of intelligent mining. In addition, the indirect method of point cloud feature extraction based on deep neural networks first requires the transformation of the point cloud, which will inevitably lose some of the spatial information of the point cloud. Additionally, the direct method loses some of the local information of the point cloud. In this paper, a method for the geometric feature extraction of point clouds in the FMMF based on DGCNN is proposed and the spherical point cloud is found in the point cloud of the FMMF, as shown in Figure 1. Edge convolution (EdgeConv) is analyzed and studied in detail in Section 2.3.

Sphere markers are made of rubber materials. Sphere markers are intermediate medium objects that transform the coordinates of the FMMF to the national geodetic coordinate system. These sphere markers were distributed in the head and tail of the FMMF, and there were three sphere markers in both the head and the tail. We only needed to find one sphere marker. The reason why we arranged so many sphere markers was to allow the DGCNN to find the sphere with the smallest position error. We collected all the point clouds of the FMMF using LiDAR on the inspection robot. Then, we used DGCNN to find the sphere, calculated the position error of each sphere, and finally displayed the sphere with the smallest position error.

As shown in Figure 1, we added noise to the single point cloud of the sphere and the point cloud of the incomplete sphere to form a training set for the point cloud so that the DGCNN could fully learn the characteristics of the point cloud of the sphere. The edge convolution layers in DGCNN were connected in turn, and the local and global features of the point cloud were obtained by max pooling. At the end of the DGCNN, multi-layer perceptron (MLP) was connected to classify the point clouds. We took the point cloud of the FMMF as the test set for the DGCNN. Through the trained DGCNN, we found the point cloud of the sphere in the point cloud of the FMMF. Finally, we framed the point cloud of the sphere and displayed the sphere position error beside it. In addition, the point clouds of the training set and testing set can be represented by a graph structure and form an  $N \times 3$  matrix where  $N$  is the number of point clouds and 3 is the three-dimensional coordinate of each point.

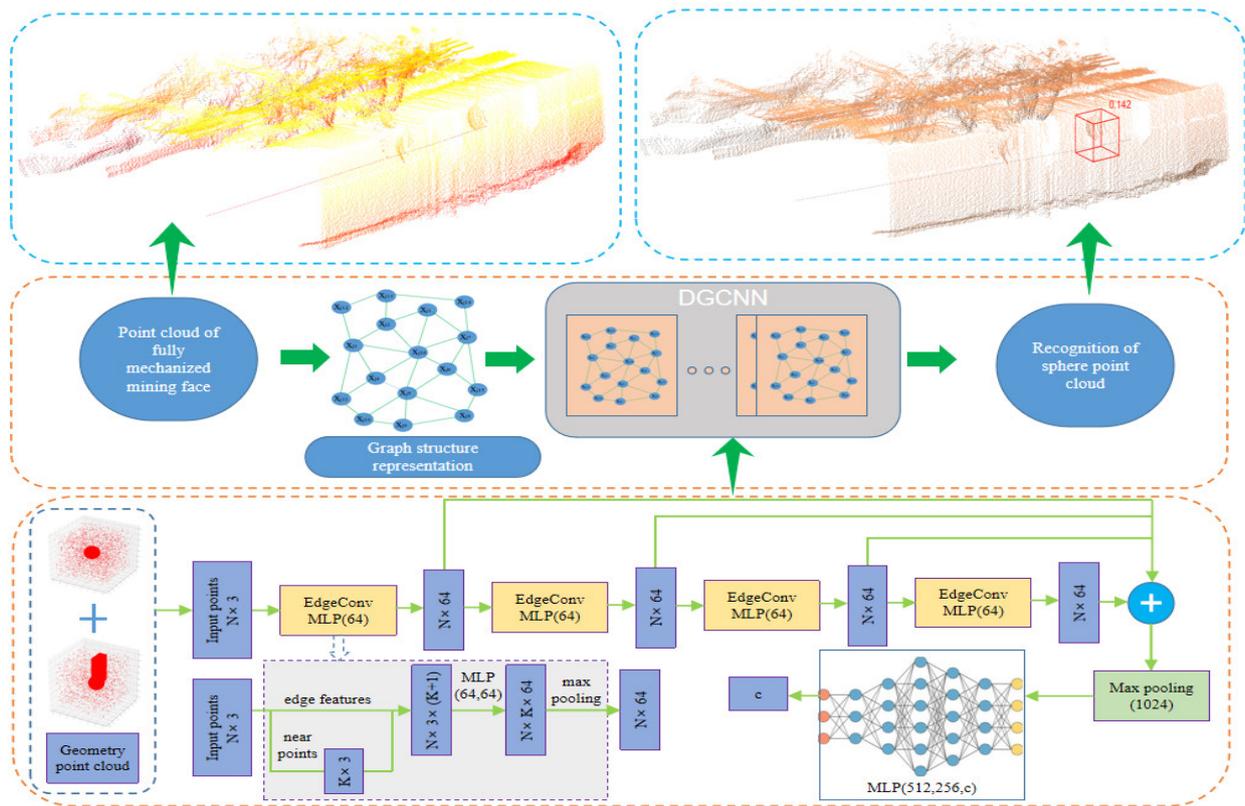


Figure 1. Running DGCNN on the point cloud of the FMMF (top) and the geometry point cloud training DGCNN (bottom).

### 2.1. Production Method of Diversity Sphere Point Cloud

The environment of the FMMF is complex, and the characteristics of the sphere point cloud are not obvious. In order to ensure that the DGCNN fully learned the characteristics of the spherical point cloud, in the point cloud of FMMF, DGCNN could still recognize the sphere when facing the complete spherical point cloud and incomplete spherical point cloud (more than half of the sphere and less than half of the sphere). Therefore, in MATLAB R2019a, the positions of the sphere and the cuboid were set to be random (Figure 2).

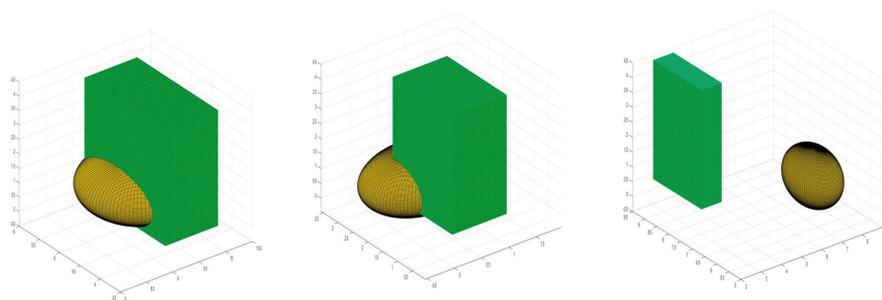
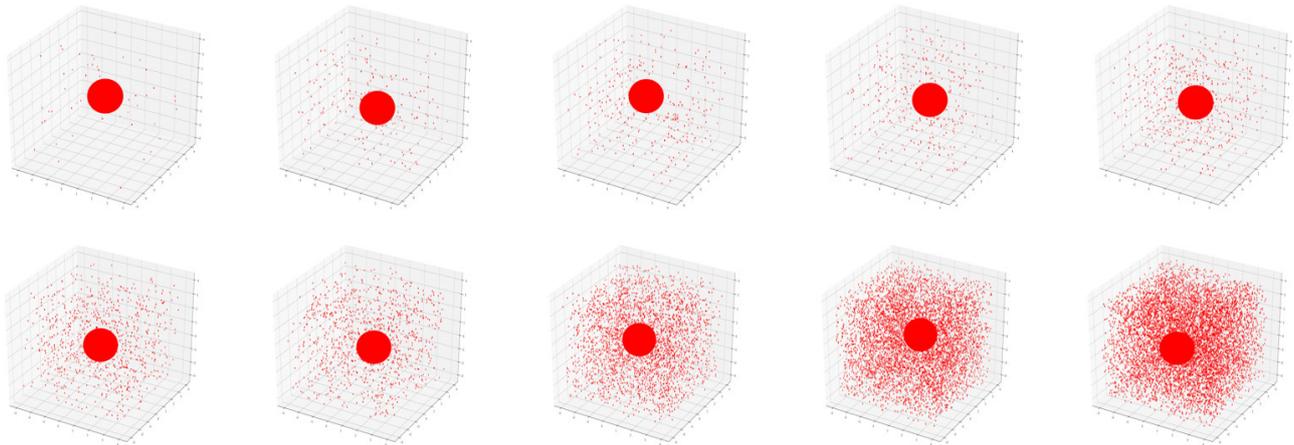
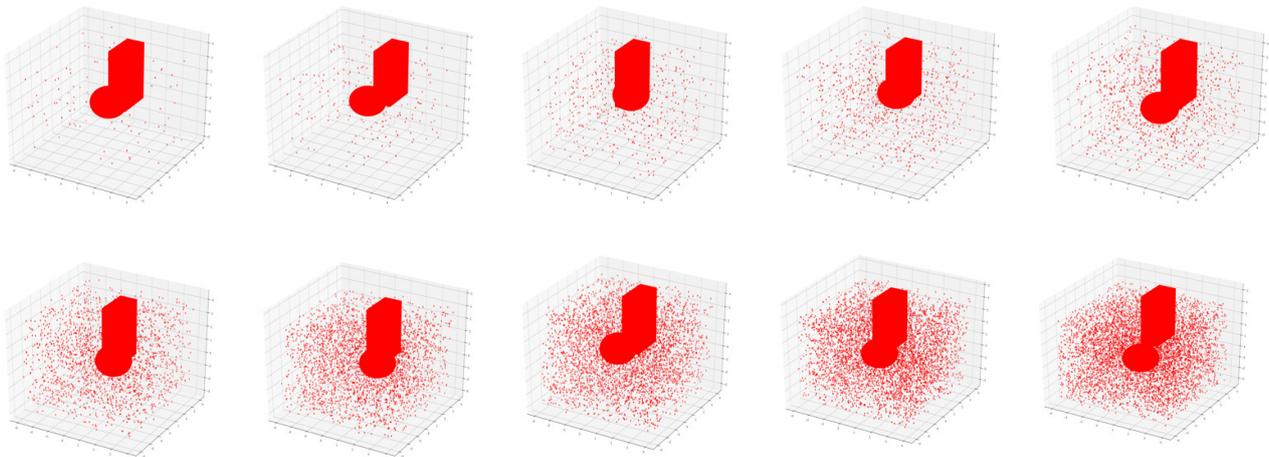


Figure 2. Spheres and combinations in different spatial positions.

As shown in Figure 2, the positions of the sphere and the cuboid were random in the specified space, and the cuboid formed different degrees of cover for the sphere. In addition, in order to increase the robustness of DGCNN, we added different numbers of noise points to the point cloud of a single sphere and the point cloud of a combination of a sphere and cuboid, as shown in Figures 3 and 4.



**Figure 3.** Point cloud of the sphere after adding different numbers of noise points.



**Figure 4.** Point cloud of the combination after adding different numbers of noise points.

As shown in Figures 3 and 4, as the number of noise points in the point cloud of the sphere and the point cloud of the combination formed by the sphere and cuboid increased gradually, the spatial position of the noise points changed randomly. Therefore, these noise points form different degrees of interference in the DGCNN recognition sphere.

## 2.2. Optimization Algorithm of DGCNN

### 2.2.1. Stochastic Gradient Descent

As the basic algorithm of the neural network optimization algorithm, the basic idea of Stochastic Gradient Descent (SGD) is to calculate the gradient on random and small-batch subsets; then, this is approximated to the real gradient on the whole data set [41]. As in Equations (1) and (2), SGD iteratively updates the weights with small batch samples at each step.

$$\omega_{t+1} = \omega_t + \Delta\omega_t, \quad (1)$$

$$\Delta\omega_t = \lambda \nabla_w E(\omega_t), \quad (2)$$

where  $\omega_{t+1}$  is the weight value at time  $t+1$ .  $\omega_t$  is the weight value at time  $t$ .  $\Delta\omega_t$  is the gradient operator at time  $t$ , that is, the weight update part of each iteration.  $\lambda$  is the learning rate of the SGD algorithm,  $E(\omega_t)$  is the loss function of weight  $\omega_t$  in the  $t$ -th iteration, and  $\nabla_w E(\omega_t)$  is the ladder degree of weight  $\omega$  in the loss function at time  $t$ .

The disadvantages of SGD are [42,43]:

- (1) For a non-convex error function, it is easy to fall into the local optimum.
- (2) It is difficult to select the appropriate learning rate for the SGD algorithm. If the learning rate is too low, the convergence speed may be very slow; if the learning rate is too large, the convergence will be hindered, which leads to the weight fluctuating near the optimal solution or even causing divergence.
- (3) SGD uses random, small batches of data to calculate the gradient, which is similar to the real gradient in the whole data set, thus reducing the calculation density, and this will cause gradient noise and variance.

### 2.2.2. Adaptive Moment Estimation

Adaptive moment estimation (Adam) is an optimization algorithm that can replace the traditional gradient descent process. It updates neural network weights iteratively based on training data [44]. The Adam algorithm is different from the traditional random gradient descent algorithm. The random gradient descent algorithm uses a single learning rate to update all weights, and the learning rate will not change during the training process. Adam designs independent, adaptive learning rates for different parameters by calculating the first-order moment estimation and second-order moment estimation of the gradient. The Adam algorithm combines the advantages of an adaptive gradient (AdaGrad) and root mean square prop (RMSProp). It has an excellent performance in solving unsteady and nonlinear problems.

In this study, we used the Adam algorithm to improve DGCNN in order to solve the problem that it is easy for the neural network model to fall into the local optimal solution as well as the problems of the poor convergence speed and learning speed. The iterative process of using Adam to update weights is shown in Figure 5 [45].

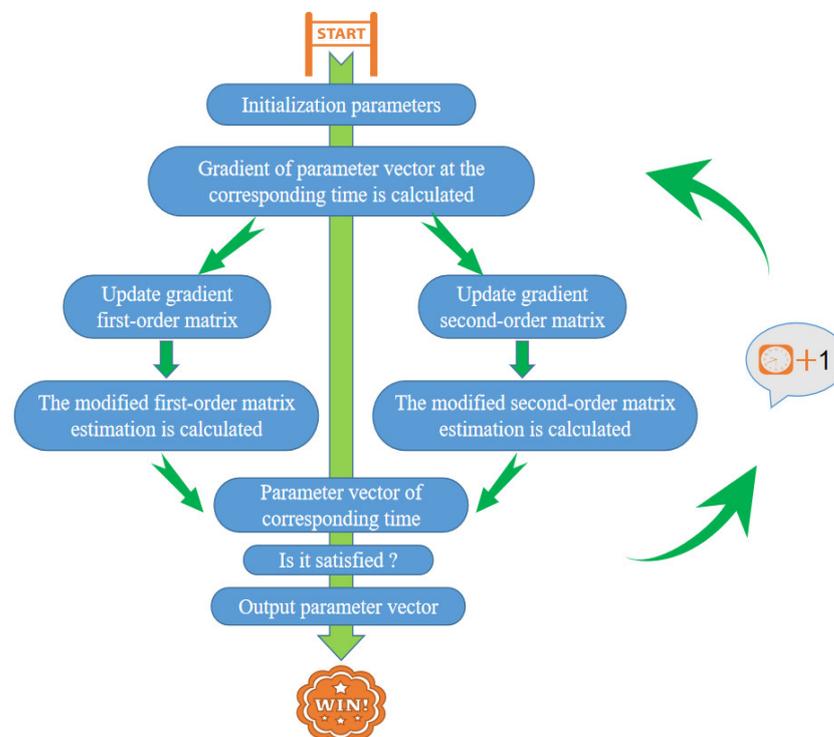


Figure 5. Basic flow of the Adam algorithm.

In the Adam algorithm,  $g$  is the gradient of random objective function  $f$ . After setting the parameters in the first step, at step  $t$ , the values of the first-order moment estimation  $h_t$  and the second-order moment estimation  $r_t$  are calculated using Equations (3) and (4).

$$h_t = \beta_1 h_{t-1} + (1 - \beta_1) g_t, \quad (3)$$

$$r_t = \beta_2 r_{t-1} + (1 - \beta_2) g_t^2, \quad (4)$$

In Equations (3) and (4),  $\beta_1$  is the exponential decay rate of the first-order moment estimation, and  $\beta_2$  is the exponential decay rate of the second-order moment estimation.  $g_1, g_2, \dots, g_t$  represent the gradient of the time step sequence where the gradient obeys  $g_t \sim p(g_t)$ . We initialize the second-order moment estimation  $r_0 = 0$ . By iterating Equation (4), we can obtain the function of the gradient and decay rate in all time steps, as shown in Equation (5).

$$r_t = \sum_{i=1}^t \beta_2^{t-i} g_i^2 - \beta_2 \sum_{i=1}^t \beta_2^{t-i} g_i^2, \quad (5)$$

We take both sides of Equation (5) as expectations at the same time to obtain Equation (6):

$$E[r_t] = E \left[ \sum_{i=1}^t \beta_2^{t-i} g_i^2 - \beta_2 \sum_{i=1}^t \beta_2^{t-i} g_i^2 \right] = E[g_t^2] (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} + \delta = E[g_t^2] (1 - \beta_2^t) + \delta, \quad (6)$$

If the second-order moment  $E[g_t^2]$  is static, then  $\delta = 0$ . In other cases,  $\delta$  has a small value. In both cases, only  $(1 - \beta_2^t)$  items are left. Therefore, when the initial time or decay rate is small,  $r_t$  will be biased to the zero vector, which needs to be divided by the  $(1 - \beta_2^t)$  term to correct the initialization deviation. After the second-order moment estimation is modified by time  $t$ , Equation (7) is obtained.

$$\hat{r}_t = \frac{r_t}{1 - \beta_2^t}, \quad (7)$$

After correction, the final weight update is obtained, as shown in Equation (8).

$$\omega_{t+1} = \omega_t - \frac{\hat{h}_t \eta}{\delta + \sqrt{\hat{r}_t}}, \quad (8)$$

In Equation (8),  $\omega_{t+1}$  is the weight at time  $t+1$ ,  $\omega_t$  is the weight at time  $t$ ,  $\hat{h}_t$  is the result of modifying the first-order moment estimation at time  $t$ ,  $\hat{r}_t$  is the result of modifying the second-order moment estimation at time  $t$ , and  $\eta$  is the learning rate set by the network.

According to existing research [46], when the above two optimization algorithms encounter a saddle model, the SGD algorithm cannot escape from the saddle bottom (the result of the SGD algorithm is the local minimum), as shown in Figure 6.

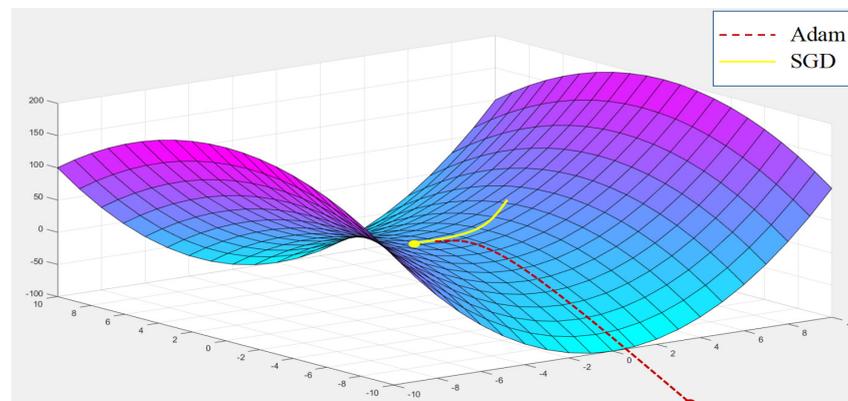


Figure 6. Effects of Adam and SGD on the saddle model.

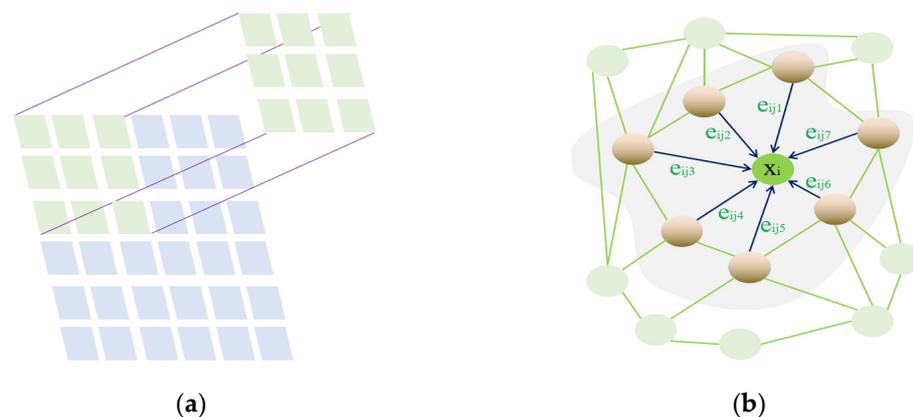
It can be seen from the saddle diagram in Figure 6 that the corresponding result of the SGD algorithm is stuck in the local minimum value of the saddle model. Therefore,

the performance of the optimization algorithm SGD in the DGCNN is improved by Adam in this study. This study focuses on the analysis of the effect of DGCNN in extracting the corresponding geometric features of the point cloud in FMMF using Adam and SGD in order to verify the correctness of the improved optimization algorithm.

### 2.3. Edge Convolution Method of Point Cloud

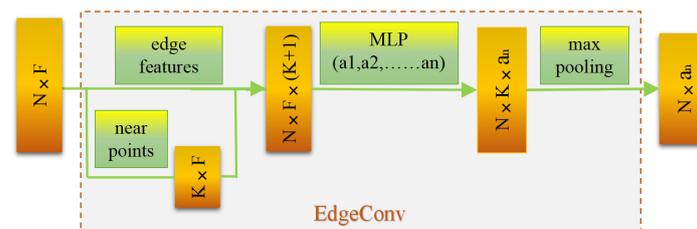
Edge convolution in DGCNN uses the connecting edge between two interconnected nodes to represent the feature information synthesis of two interconnected nodes, then aggregates the feature information of multiple nodes interconnected with the central node through a series of nonlinear transformations so as to express the local features of the central node [47].

In edge convolution,  $k$  nearest points are selected for the center point by the  $k$ -nearest neighbor (KNN) graph algorithm, and each point is regarded as the center point of the graph structure in turn for edge relation calculation, feature concatenation, and MLP calculation. When edge convolution is carried out layer by layer, each layer will produce a new graph structure and output a new feature space. The operation diagram of conventional convolution and edge convolution is shown in Figure 7.



**Figure 7.** Conventional convolution and edge convolution. (a) Schematic diagram of a conventional convolution operation on Euclidean structured data and (b) schematic diagram of edge convolution operation on point cloud data.

As shown in Figure 7, compared with the traditional convolution on Euclidean structured data, edge convolution uses  $k$ -nearest neighbors to define the  $k$  points nearest to the center point as the adjacent region of the center point. Edge convolution first extracts the edge features between the center point and the adjacent points, then convolutes the edge features. Therefore, after the edge convolution operation, each central node contains its own characteristic information and the characteristic information of  $k$ -adjacent nodes. The framework of edge convolution is shown in Figure 8.



**Figure 8.** Frame graph of the edge convolution.

As shown in Figure 8,  $N$  is the total number of points in the point cloud,  $f$  is the dimension information of each point,  $(a_1, a_2, \dots, a_n)$  in MLP is the input dimension information

and the output dimension information of each layer, and  $K$  is the number of adjacent nodes in the KNN graph algorithm.

By down-sampling the target point cloud, we obtained an  $F$ -dimensional point cloud with  $n$  points,  $X = \{x_1, x_2, \dots, x_n\} \subseteq R^F$ . In this study,  $f$  is the coordinate information of each central point ( $F = 3$ ). In the neural network structure, the neurons of each layer operate on the output of the previous layer, so dimension  $f$  is used to represent the characteristic dimension of the neuron input. Then, a directed graph  $G = (v, \varepsilon)$  is set to represent the local structure of the point cloud according to the number of points in the point cloud in which  $v = \{1, 2, \dots, n\}$  and  $\varepsilon \subseteq v \times v$  represent the vertices and edges of the directed graph, respectively. On the graph structure, the edge between the center point and the adjacent point can be expressed using Equation (9) [48]:

$$e_{ij} = h_{\Theta}(x_i, x_j - x_i), \quad (9)$$

In Equation (9),  $x_i$  is the center point,  $x_j$  is the point adjacent to the center point,  $h_{\Theta} : R^F \times R^F \rightarrow R^{F'}$ , and  $\Theta$  is the learnable parameter. Therefore,  $h_{\Theta}$  is a nonlinear activation function composed of the parameter  $\Theta$ . The MLP is used to calculate the features of all the associated edges and the features of the center so as to obtain the high-dimensional features of the graph structure. The output of the center point on the graph structure can be expressed using Equation (10):

$$x'_i = \text{pooling}_{j:(i,j) \in \varepsilon} h_{\Theta}(x_i, x_j - x_i), \quad (10)$$

In this study, DGCNN was used to process the point cloud data of FMMF. After stacking a large number of network layers, the traditional deep neural network model achieved remarkable results in many problems due to its powerful representation ability. In the multi-layer edge convolution layer DGCNN, the neighborhood information extracted by edge convolution can potentially represent the far distant area in the original cloud space. With the increase in the number of edge convolution layers, the receptive field becomes larger and larger, and the features of the central node aggregate more information of the nodes. The edge convolution in DGCNN includes aggregating the features of neighbor nodes. When the edge convolution layers are stacked in multiple layers, the features between the nodes are too smooth and lack discrimination, which affects the network performance. Therefore, this work analyzes the influence of different edge convolution layers on the performance of the DGCNN in Section 3.1.

#### 2.4. Searching Method for Markers in Point Cloud of FMMF

The point clouds of the FMMF analyzed in this study were collected from 43,101 mining face of Yujialiang coal mine in the northeast of Shenmu City, Shaanxi Province, China (Figure 9). Yujialiang coal mine is a large-scale export coal base of the China National Energy Group. At the same time, Yujialiang coal mine is one of the ten million ton mines of Shendong coal group. Yujialiang coal mine covers an area of 56.33 km<sup>2</sup>, with geological reserves of 504 million tons and recoverable reserves of 355 million tons. The coal quality in the mine field is excellent, with the characteristics of ultra-low sulfur, ultra-low phosphorus, ultra-low ash, and high calorific value [49].

The length of 43,101 mining face is 351.4 m, the advancing length is 1809.4 m, the thickness of the coal seam is 1.0~1.7 m, the average thickness is 1.5 m, the dip angle is 1~3°, the unit weight is 1.30 t/m<sup>3</sup>, and the mining height is 1.4 m [50]. We installed an expensive underground inspection robot on the special track outside the scraper conveyor in 43,101 mining face, and the point cloud of the whole FMMF was obtained by the LiDAR function on the inspection robot (Figure 10).

We used MATLAB R2019a to read the point cloud of FMMF collected by LiDAR. For intuitive observation, we intercepted part of the point cloud for visualization, as shown in Figure 11. The blue line in the point cloud is the track of the LiDAR.

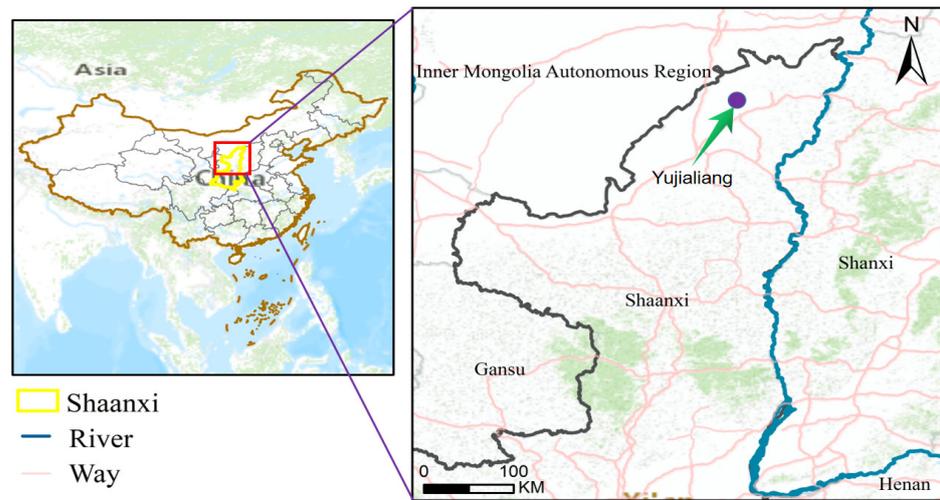


Figure 9. Location of Yujialiang coal mine.

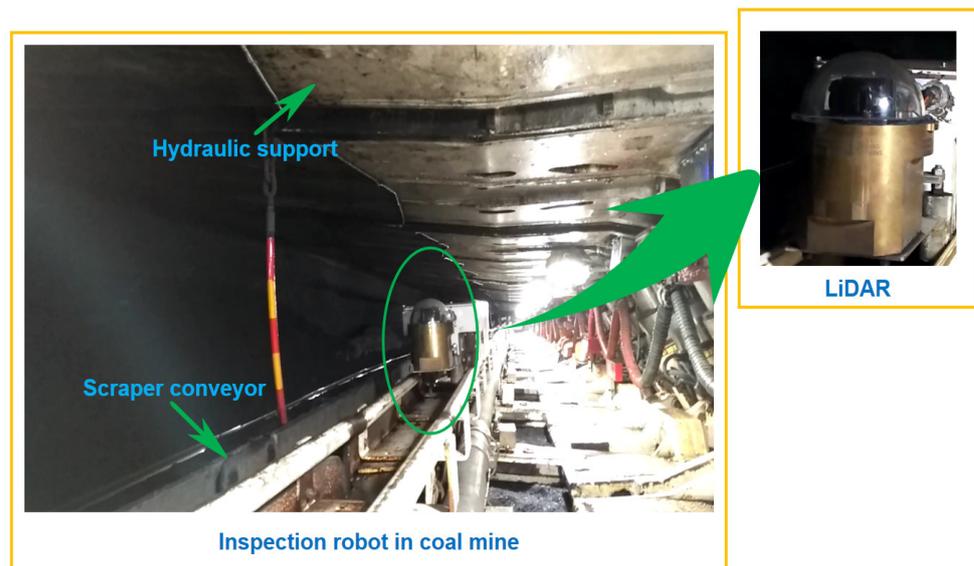


Figure 10. Inspection robot used in Yujialiang coal mine.

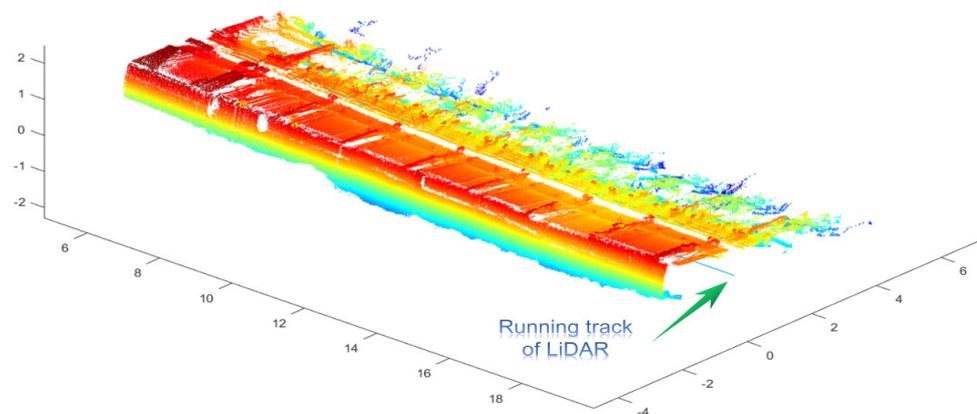
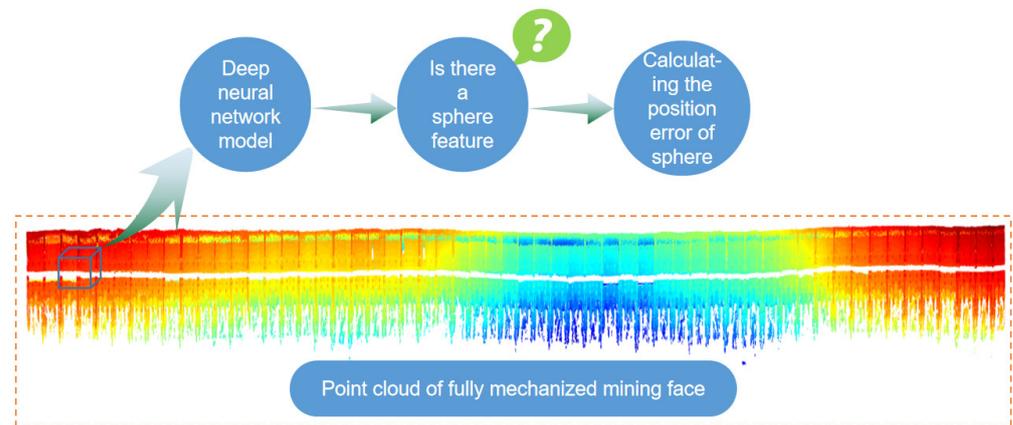


Figure 11. Point cloud of FMME.

On the point cloud of FMME, the sliding frame scanning method was used to find the point cloud of the sphere, as shown in Figure 12.



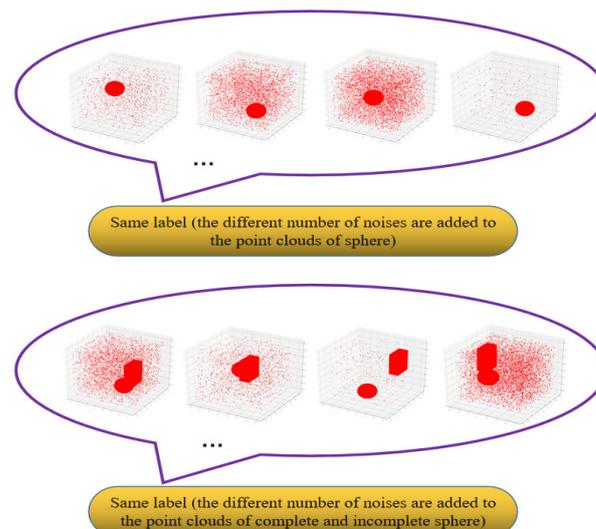
**Figure 12.** Searching for a spherical point cloud on the point cloud of FMMF.

As can be seen from Figure 12, the lower part of the figure is the point cloud of the FMMF collected by the LiDAR on the inspection robot. In this study, we used the depth neural network model to traverse and intercept the point cloud of the FMMF in turn in the way of the sliding window, then judged whether the point cloud intercepted had the characteristics of a sphere. If the feature of sphere exists in the current point cloud, the position error between the sphere identified by the depth neural network and the real sphere can be calculated. Finally, we took the sphere with the least position error as the recognition result of the depth neural network and visualized it.

### 3. Experiment and Results

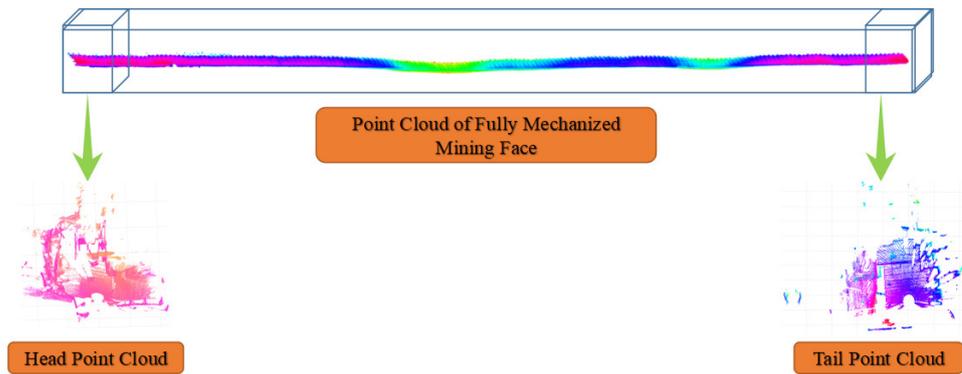
#### 3.1. Data Set Description

We labeled the complete sphere and incomplete sphere point clouds after adding the different levels of noise and completed the production of the point cloud training set according to the method described in Section 2.1. Part of the point cloud training set is shown in Figure 13.



**Figure 13.** Some point cloud data sets.

In Section 2.4, we obtained the point cloud of the FMMF. We sought to find the point cloud of the sphere in the point cloud of the FMMF through the trained DGCNN. The real position of the sphere was mainly distributed in the head and tail of the point cloud of the FMMF. The position of the point cloud of the head and tail is shown in Figure 14.



**Figure 14.** Position of the point cloud of the head and tail.

As shown in Figure 14, the top of the figure is the main view of the point cloud of the whole FMMF in the Y direction. The lower part of the figure is the top view of the head and tail point clouds of the FMMF in the Z direction, which occupies a smaller space than the whole FMMF. When the shearer reaches the head or tail of the FMMF, it adjusts the driving direction to mine again.

### 3.2. Experimental Details

We propose an improved DGCNN to identify the sphere in the point cloud of the FMMF. The experiment was carried out on a workstation with an Intel Core-I7 9700 CPU @ 4.70 GHz with an NVIDIA GTX 2080 Super GPU and 16 GB of video memory. The software configuration included PyTorch 1.7.1, Python 3.7.9, and Tensorflow 2.3.0 to build the structure of DGCNN. We set the batch size as 8, the epochs as 100, the K value as 20, and the dropout rate as 0.5 in the KNN graph algorithm. The activation function of the edge convolution layer and the MLP layer used ReLU. The loss function was cross entropy loss function [38], and its mathematical expression is shown in Equation (11).

$$loss = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)], \quad (11)$$

In Equation (11),  $x$  is the sample,  $y$  is the actual tag,  $a$  is the predicted output, and  $n$  is the total number of samples.

Precision rate (precision), recall rate (recall), F1 score (F1), and position error are used to evaluate the effect of the neural network. The mathematical definition of precision is shown in Equation (12).

$$Precision = \frac{TP}{TP + FP'} \quad (12)$$

$TP$  is the number of samples whose real tags belong to positive class and whose predicted value is also positive class, and  $FP$  is the number of samples whose real tags belong to negative class but whose predicted value is positive class. The mathematical definition of recall is shown in Equation (13).

$$Recall = \frac{TP}{TP + FN'} \quad (13)$$

$FN$  is the number of samples whose real tags belong to positive class but whose predicted value is negative class. The mathematical definition of F1 is shown in Equation (14).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall'} \quad (14)$$

In the point cloud of FMMF, the true center position of the sphere point cloud can be obtained by the MATLAB R2019a software. In this paper, the position error of spherical point cloud in FMMF detected by DGCNN is evaluated by comparing the detected value with the actual value. The formula for calculating the position errors of different spheres is shown in Equation (15).

$$E_i = \sqrt{(x_{i2} - x_{i1})^2 + (y_{i2} - y_{i1})^2 + (z_{i2} - z_{i1})^2}, \quad (15)$$

In Equation (15),  $i$  is the number of spheres;  $x_{i1}, y_{i1}, z_{i1}$  are the actual spherical center coordinate value when the number of spheres in the point cloud is  $i$ ; and  $x_{i2}, y_{i2}, z_{i2}$  are the detected spherical center coordinate value when the number of spheres processed by DGCNN is  $i$ . We obtained and displayed the result of minimum error of the sphere position.

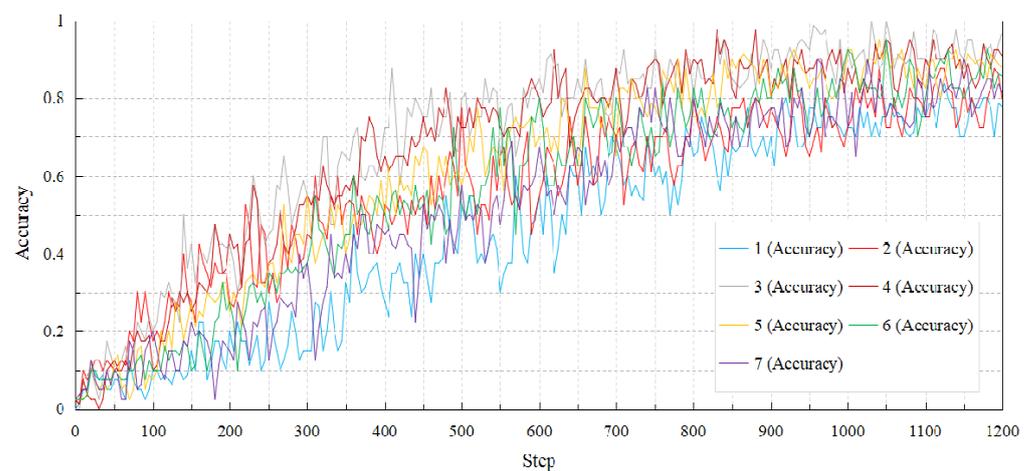
Accuracy, precision, recall, and F1 are dimensionless quantities. Accuracy means the proportion of the correct predicted results in the total samples. Precision refers to the prediction results and indicates the proportion of actual positive samples in the predicted positive samples. Recall refers to the original sample and indicates the proportion of predicted positive samples in the actual positive samples. F1 is the harmonic mean of precision and recall. Therefore, there is no unit for accuracy, loss, precision, recall, or F1. Loss is used to estimate the degree of inconsistency between the predicted value and the real value of the model, so loss has no unit. The unit of the position error of the sphere is m.

### 3.3. Experimental Results and Discussion

#### 3.3.1. Performance Comparison of DGCNN with Different Edge Convolution Layers

We adjusted the number of edge convolution layers in the DGCNN. The accuracy rate (accuracy) and loss of the DGCNN at different steps when using different numbers of edge convolution layers are shown in Figures 15 and 16.

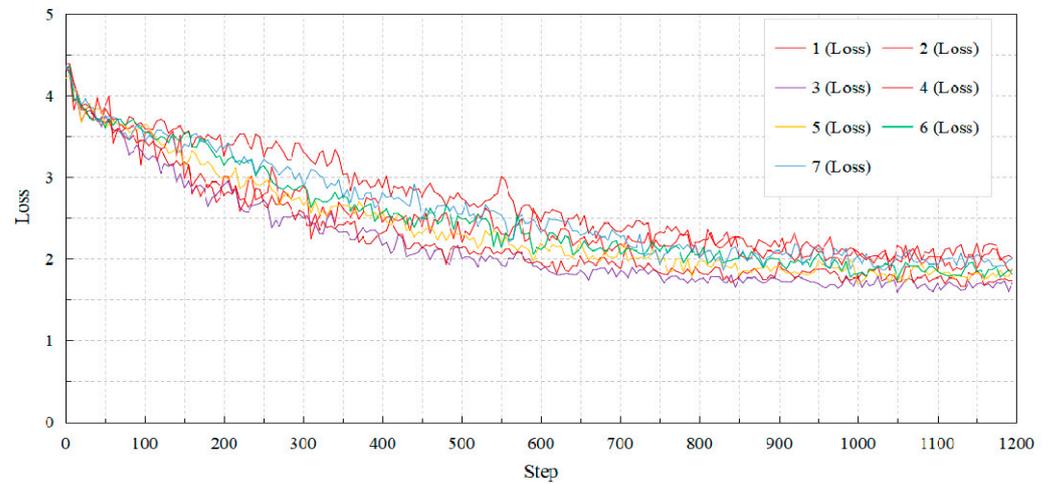
It can be seen that when the number of edge convolution layers is three, the accuracy of the DGCNN is the largest, and the loss of the DGCNN is the smallest. For intuitive and accurate analysis and research, the precision, recall, and F1 score of DGCNN when using different numbers of edge convolution layers are shown in Figure 17.



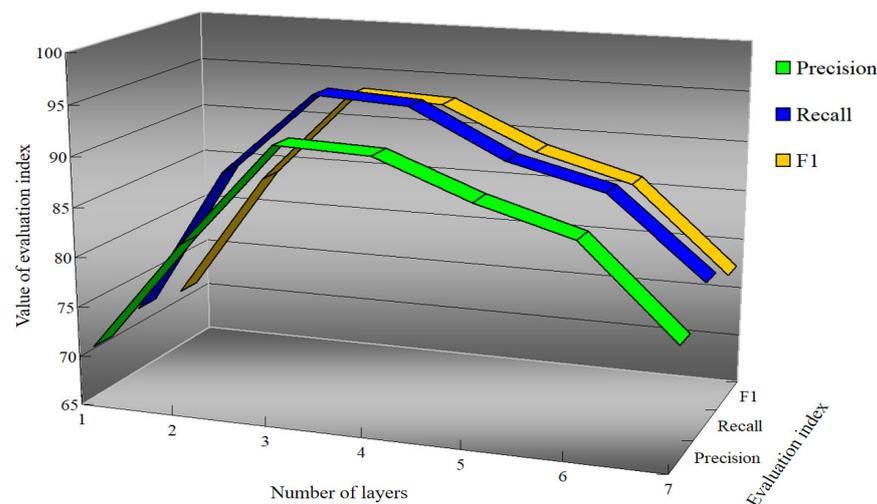
**Figure 15.** Accuracy of the different numbers of edge convolution layers.

It can be seen from Figure 17 that with the increase in the number of edge convolution layers, the precision, recall, and F1 show the phenomenon of first rising and then falling. Among them, when the number of edge convolution layers is three, the precision, recall, and F1 obtain their maximum values of 0.904, 0.937, and 0.920, respectively. This is because the edge convolution in DGCNN includes the operation of aggregating neighbor nodes. With the increase in the number of edge convolution layers, the center point can potentially represent the node information at a long distance and then fully and effectively describe

the local features. However, when there are too many edge convolution layers, and each node contains too much information from other nodes, the discrimination of each node is not obvious and lacks discrimination, and this then affects the network performance.



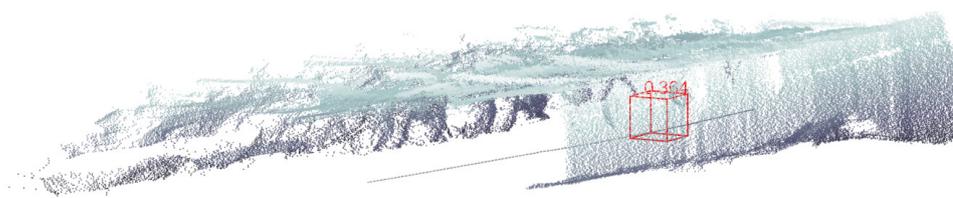
**Figure 16.** Loss of the different numbers of edge convolution layers.



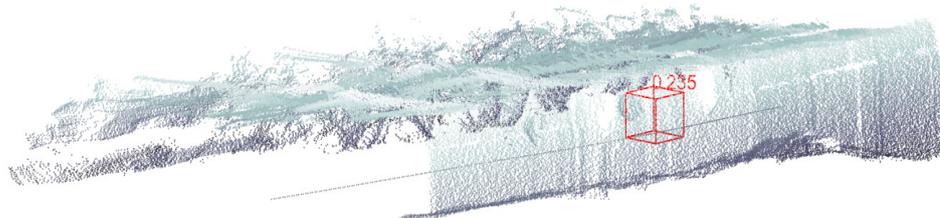
**Figure 17.** Value of the evaluation index when using different numbers of edge convolution layers.

In view of the black box problem of deep neural networks, the existing broken line chart, histogram, and other forms cannot be better imaged for the vivid evaluation of DGCNN to identify the effect of spherical point cloud in FMMF. We visualized the results of the point cloud of FMMF processed by DGCNN (the length of the whole FMMF is too long; in order to facilitate intuitive observation, we visualized the point cloud of partial FMMF where the sphere was located). Figure 18 shows the recognition effect of the point cloud of the sphere under different numbers of edge convolution layers.

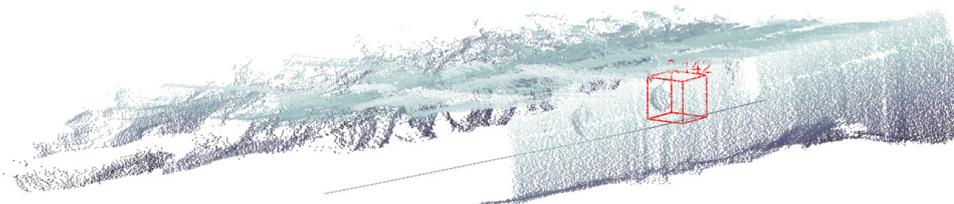
The number next to the box in Figure 18 represents the position error between the sphere recognized by DGCNN and the real sphere. The number of edge convolution layers of DGCNN increased from one layer to seven layers, and the position errors were 0.343, 0.364, 0.235, 0.142, 0.161, 0.193, and 0.286 m, respectively. It can be seen that with the increase in the number of edge convolution layers, the position error of the sphere first decreased and then increased. When the number of edge convolution layers was three, the position error of the sphere was the smallest, and the performance of the DGCNN was the best. These findings are consistent with the conclusion of the best number of edge convolution layers of DGCNN according to precision, recall, and F1.



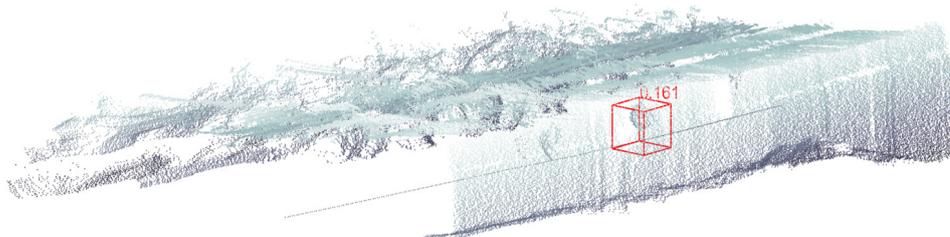
(a)



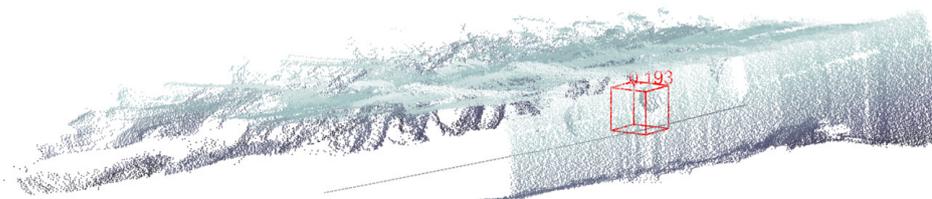
(b)



(c)

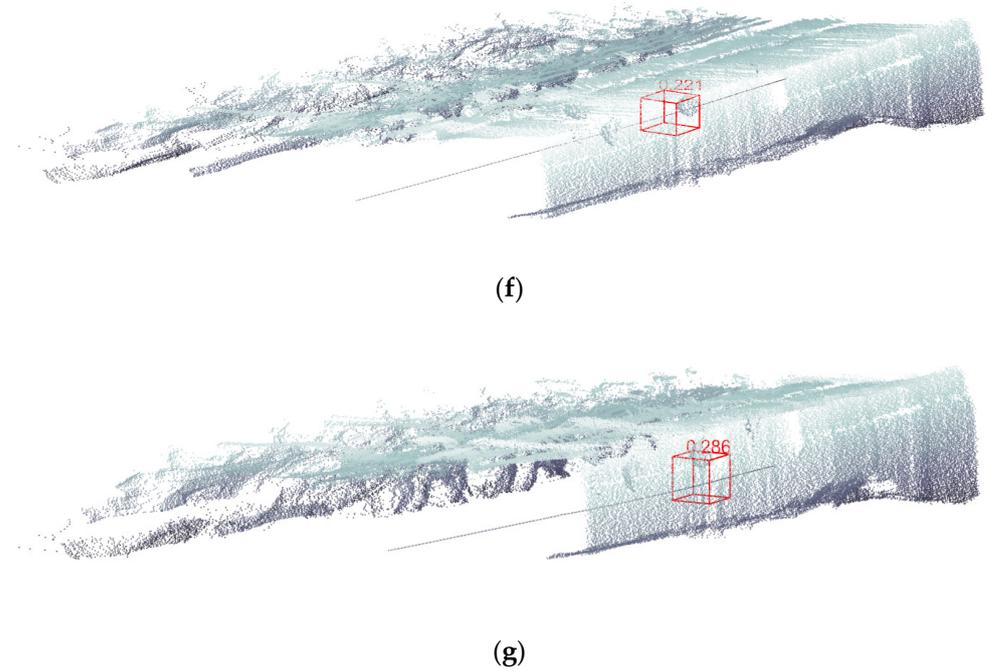


(d)



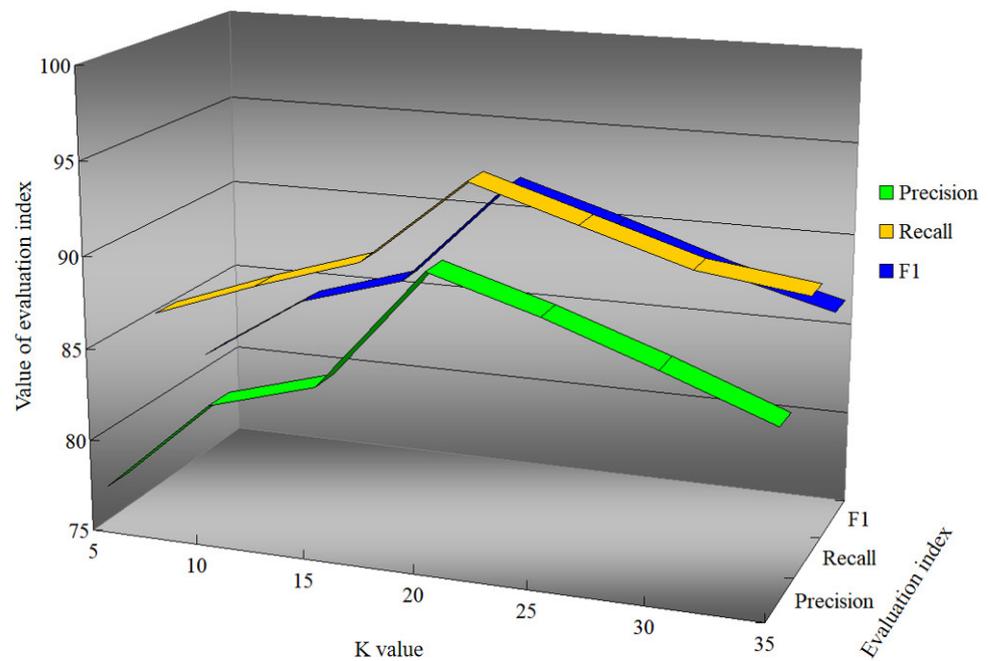
(e)

Figure 18. Cont.



**Figure 18.** Effect of using different numbers of edge convolution layers: (a) 1 layer, (b) 2 layers, (c) 3 layers, (d) 4 layers, (e) 5 layers, (f) 6 layers, and (g) 7 layers.

In the edge convolution,  $K$  edge features around the center point were constructed to represent the relationship between the center point and adjacent points. We adjusted the  $K$  value inside the edge convolution layer. The precision, recall, and F1 of the DGCNN when using different  $K$  values are shown in Figure 19.



**Figure 19.** Values of the evaluation index when using different  $K$  values.

It can be seen from Figure 19 that different  $K$  values have an impact on the effect of DGCNN, while the precision, recall, and F1 show a phenomenon of first rising and then falling. Among them, when  $K$  was 20, the precision, recall, and F1 obtained their maximum

value. This is because when the K value was too small, there were few adjacent points around the center point (each adjacent point and the center point were represented by an edge feature), and the edge convolution in DGCNN could not learn the local features of the point cloud well. However, when the K value was too large, edge convolution learned the features of the point cloud in a large neighborhood, which made the overall model simple and fall into learning the local details of the point cloud.

In addition, we obtained the recognition effect of DGCNN on the point cloud of the sphere in the FMMF under different K values, as shown in Figure 20.

As can be seen from Figure 20, when the K value was 20, the position error of the sphere was the smallest, which indicates that the performance of the model changes when the K value is different. This, then, has an impact on the effect of the DGCNN processing point cloud. Additionally, when  $K = 20$ , the effect of DGCNN searching for the point cloud of the sphere in the FMMF is the best.

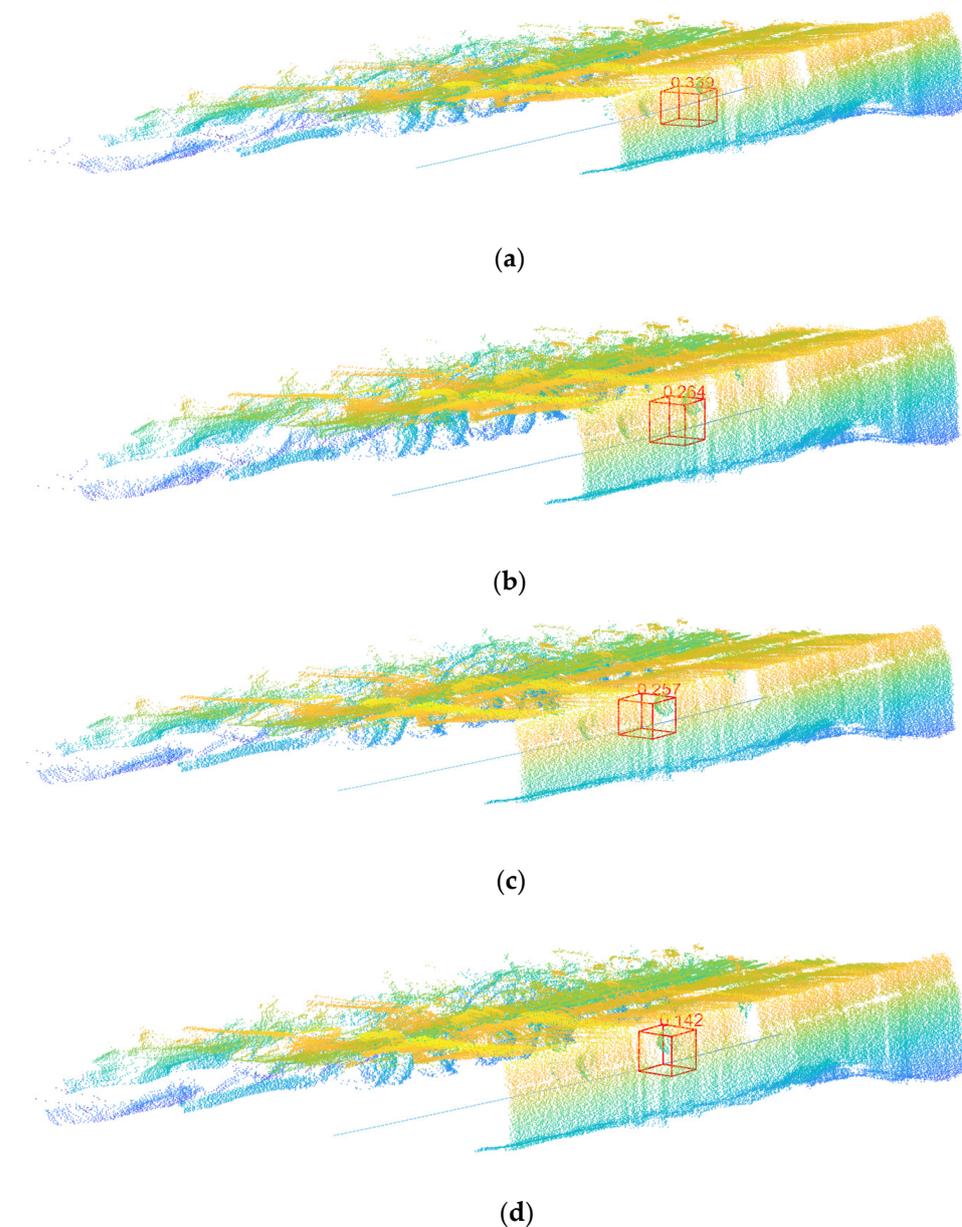
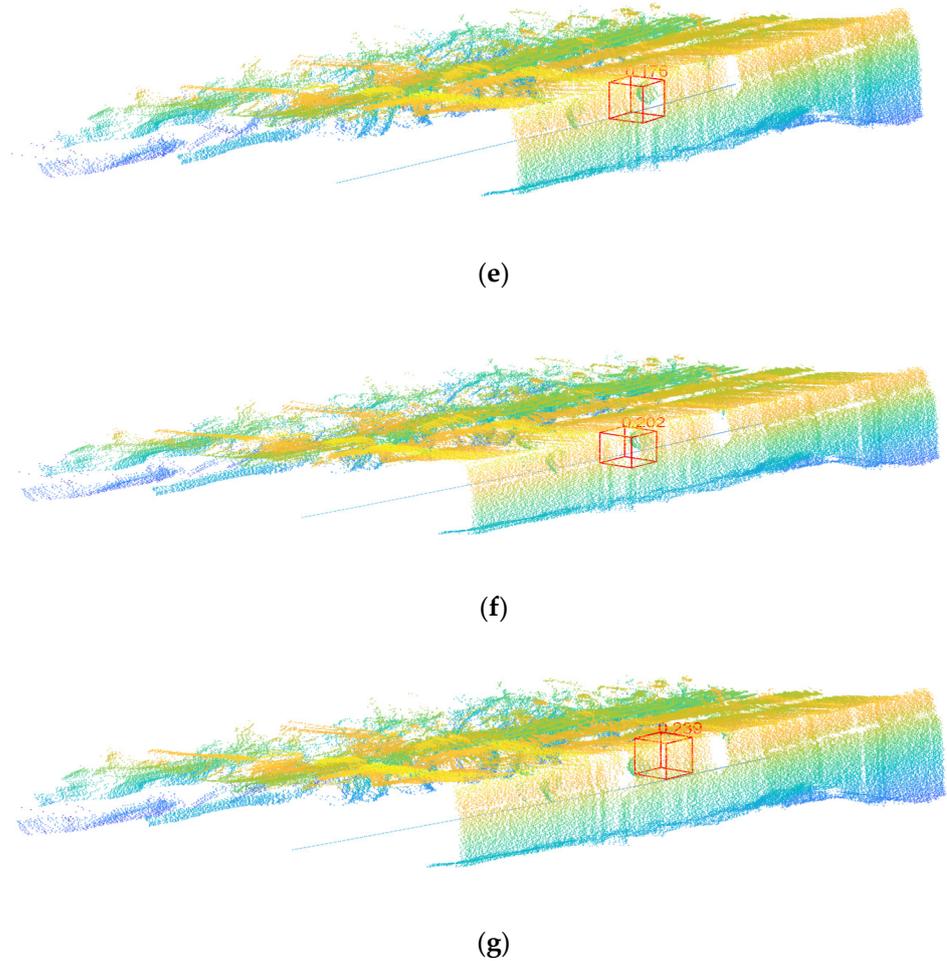


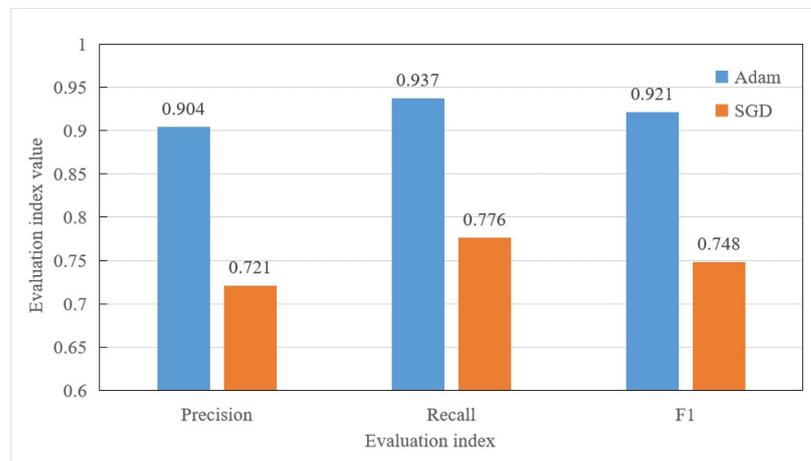
Figure 20. Cont.



**Figure 20.** Effect of different K values: (a) K is 5, (b) K is 10, (c) K is 15, (d) K is 20, (e) K is 25, (f) K is 30, and (g) K is 35.

### 3.3.2. Performance Comparison of DGCNN with Different Optimization Algorithms

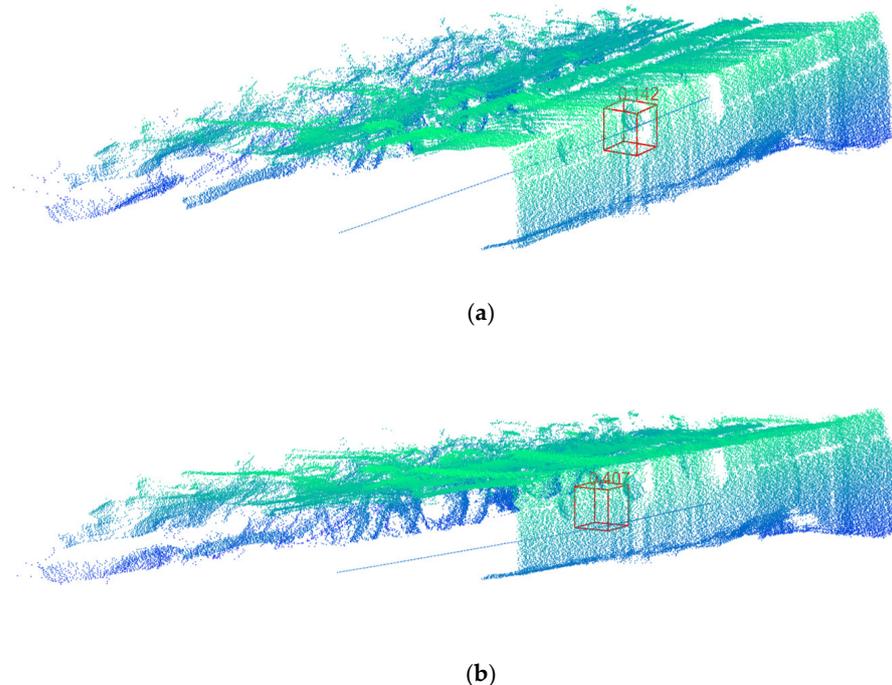
After determining the number of edge convolution layers of DGCNN, the precision, recall, and F1 of DGCNN under different optimization algorithms were analyzed, as shown in Figure 21.



**Figure 21.** Evaluation index values of DGCNN when using different optimization algorithms.

As can be seen from Figure 21, the precision, recall, and F1 values of DGCNN when using the Adam optimization algorithm were higher than those of the SGD optimization algorithm, and the differences were 0.183, 0.161, and 0.173, respectively. This is because the learning rates of all parameters in the SGD optimization algorithm are the same, which is unreasonable because some parameters do not need to change frequently, while some parameters need to learn and improve frequently. The Adam optimization algorithm combines the momentum algorithm and the RMSProp algorithm and uses the first- and second-order moment estimation of the gradient to dynamically adjust the learning rate of each parameter. The Adam optimization algorithm not only calculates the adaptive learning rate for each parameter based on the first-order moment estimation to improve the performance on sparse gradient but also remains well-adapted to the learning rate for each parameter based on the second-order moment estimation of the weight gradient.

We ran DGCNN with different optimization algorithms on the point cloud of FMMF. Similarly, because the length of the FMMF was too large, and the number of point clouds was particularly large, in order to facilitate intuitive observation we had to visualize the local point clouds of the actual FMMF. The recognition effect of the spherical point cloud on the point cloud of FMMF is shown in Figure 22.



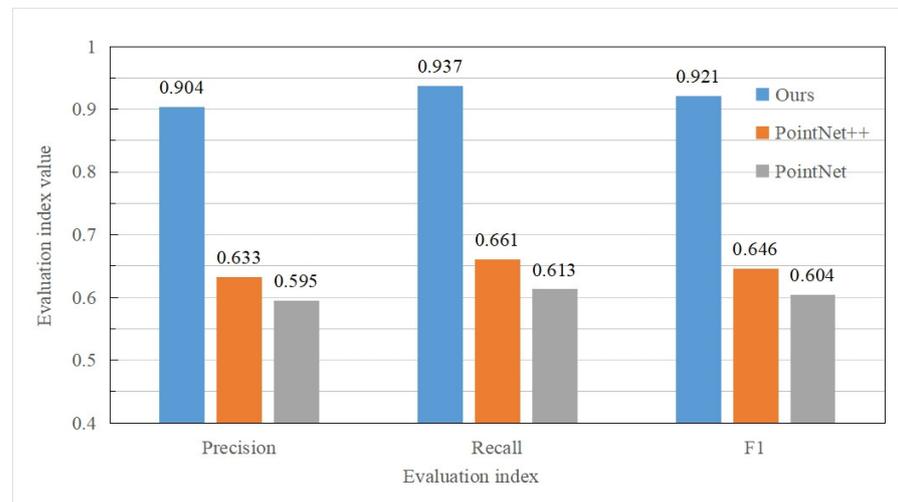
**Figure 22.** Recognition effect of the point cloud of the sphere based on different optimization algorithms: (a) the effect of the Adam optimization algorithm and (b) the effect of the SGD optimization algorithm.

As can be seen from Figure 22, the position error of DGCNN using the Adam optimization algorithm was lower than that of the SGD optimization algorithm, and the difference was 0.265 m. Regarding the effect of sphere recognition in the point cloud of FMMF, the effect of the Adam optimization algorithm is better than that of the SGD optimization algorithm. Therefore, we improved the DGCNN running on the point cloud of FMMF to three layers of the edge convolution layer; the optimization algorithm used was Adam.

### 3.3.3. Performance Comparison of Different Neural Networks

The essential difference between DGCNN and PointNet is that DGCNN innovatively designs a critical module called edge convolution, which enables DGCNN to learn the local features of point clouds. PointNet is equivalent to DGCNN without edge convolution [47].

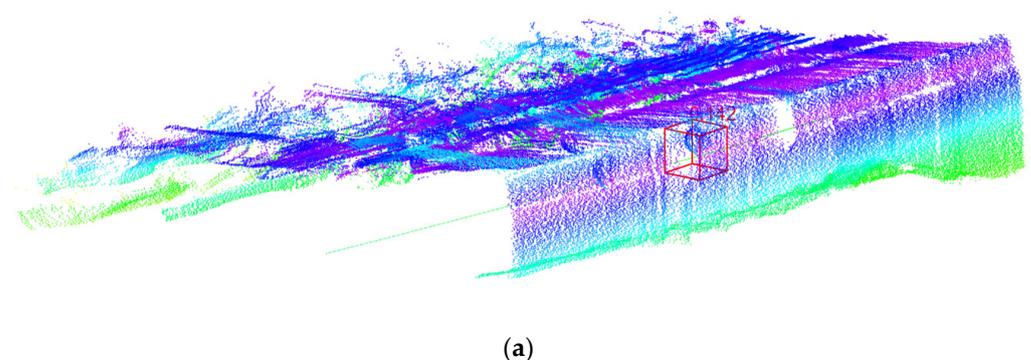
Therefore, we compared the improved DGCNN with PointNet and PointNet+. This is not only the result comparison of different depth neural network models but also the result of comparison to an ablation study of DGCNN without edge convolution. The precision, recall rate, and F1 of the three methods are shown in Figure 23.



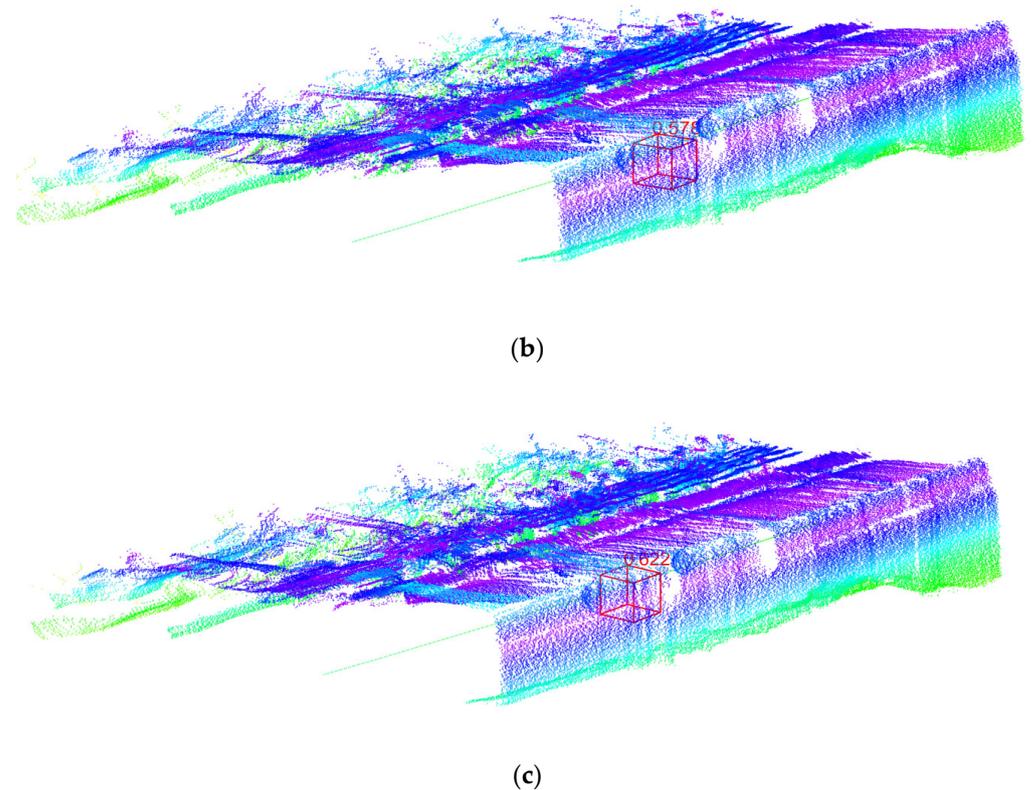
**Figure 23.** Comparison of the effects of different neural networks.

As can be seen from Figure 23, the precision, recall, and F1 obtained by our improved DGCNN are higher than those obtained by PointNet and PointNet++, and the maximum differences of precision, recall, and F1 were 0.309, 0.324, and 0.317, respectively. This is because PointNet and PointNet++ treat each data point in the point cloud as an isolated existence and process each point in the point cloud independently, ignoring the geometric information between the points, meaning that they lose the local features of the point cloud. DGCNN constructs the features of the local neighborhood of the point cloud using edge convolution, which not only considers the features of the current point but also considers the features of the K nearest points of the current point. This means that DGCNN can extract the features of the local neighborhood from the local graph composed of K points. Therefore, DGCNN improves the shortcoming of PointNet and PointNet++ lacking local information of the point cloud in the process of feature extraction.

In order to facilitate intuitive observation, we visualized the local point cloud of FMMF. The recognition effect of our improved DGCNN, PointNet, and PointNet++ on the point cloud of sphere in FMMF is shown in Figure 24.



**Figure 24.** Cont.



**Figure 24.** Recognition effect of the point cloud of the sphere based on different neural networks: (a) ours, (b) PointNet, and (c) PointNet++.

As shown in Figure 24, the position error obtained by our improved DGCNN is lower than that obtained by PointNet and PointNet++. The difference between DGCNN and PointNet is 0.580 m, and the difference between DGCNN and PointNet++ is 0.436 m. This shows the correctness and feasibility of using the improved DGCNN to identify the spherical point cloud in the point cloud of FMMF. The research results improve the adaptability, generalization ability, and practical value of the DGCNN model and provide a reference for the application of DGCNN in energy- and geology-related fields.

In addition, the contribution of the critical step edge convolution in DGCNN is as follows: firstly, the critical step edge convolution can extract the local features of the point cloud, which solves the defect of the depth neural network in this aspect. Secondly, DGCNN can obtain an ideal point cloud processing effect by stacking appropriate edge convolution layers. In addition, edge convolution can not only extract the local features of point cloud but also maintain the invariance of the point cloud arrangement. Additionally, because of the existence of an edge convolution layer, the DGCNN model can better learn the information of the point cloud by dynamically updating the graph structure between layers.

Although our improved DGCNN model is better than PointNet and PointNet++ in dealing with the point cloud of FMMF, the point cloud itself has the characteristics of dispersion and non-uniformity, and the FMMF is dusty and low-visibility. In the training set, both the point cloud of the complete sphere and the point cloud of the incomplete sphere were generated pertinently, and our proposed method was also targeted to deal with the point cloud of FMMF. Therefore, the current method has the defect that it cannot process cross-domain tasks (for example, references [51,52]). In the future, we will perform further research to improve the current method so that it has the ability to process cross-domain tasks.

#### 4. Conclusions

In order to obtain the position of the sphere (marker) in the point cloud of the FMMF, this study provides the basis for the subsequent coordinates transformation of the FMMF to geodetic coordinates using the sphere as the intermediate medium, then provides the basis for the straightening and inclining adjustment of the scraper conveyor. In addition, for the indirect method of the point cloud feature extraction of deep neural networks, the point cloud needs to be transformed first, which will inevitably lose some of the spatial information of the point cloud, while the direct method is unable to capture the local information of the point cloud. Therefore, we propose a method to extract the geometric features of the sphere in the point cloud of FMMF using DGCNN. Firstly, in order to increase the robustness of DGCNN, we added different levels of noise to the point clouds of complete and incomplete spheres in order to complete the production of point cloud data sets. At the same time, we took the point cloud of FMMF collected by LiDAR as the test set. The results fill the research gap concerning using a depth neural network model that has no data set to extract the characteristics of a spherical point cloud in the point cloud of FMMF. Secondly, edge convolution is used to aggregate the feature information of adjacent nodes to the central node. In DGCNN with a multi-layer edge convolution layer, the neighborhood information extracted by edge convolution can potentially represent a very long distance in the original space, but too deep an edge convolution layer will cause overfitting, which will affect the performance of the neural network. Therefore, we analyzed the effect of the DGCNN in extracting geometric features from the point cloud of FMMF when using different numbers of edge convolution layers and improved the optimization function of DGCNN. The results show that with the increase in the number of edge convolution layers, the precision, recall, and F1 show a trend of first rising and then falling, while the position error shows a trend of first falling and then rising. When the number of edge convolution layers is three, the precision, recall, and F1 are the highest, and the position error is the lowest; these are 0.9043, 0.9369, 0.9203, and 0.142, respectively. In addition, the effect of DGCNN when using the Adam optimization algorithm is better than that of the SGD optimization algorithm. Finally, we compared the effect of using the improved DGCNN with PointNet and PointNet++ to identify the point cloud of a sphere in the point cloud of FMMF. The results show that the improved DGCNN is better than PointNet and PointNet++. This shows the feasibility of using DGCNN to extract the features of a sphere in the point cloud of FMMF.

Therefore, this study provides a new method for the feature extraction of a sphere in the point cloud of FMMF and provides the foundation for the subsequent coordinate transformation of FMMF to the national geodetic coordinates when using a sphere as the intermediate medium, thus providing a basis for the straightening and inclining adjustment of scraper conveyor. Additionally, this can provide an effective basis for the command and control of coal mine production and provide a reference for the application of DGCNN in energy- and geology-related fields. At the same time, the results provide a direct early guarantee for analyzing the point cloud data of FMMF under the national geodetic coordinate system in the future, which is not only conducive to improving the mining efficiency and safety of the coal seam, but it is also of great significance for the transparent, unmanned, and intelligent mining of FMMF.

**Author Contributions:** Conceptualization, Zhizhong Xing, Shuanfeng Zhao and Wei Guo; methodology, Zhizhong Xing and Shuanfeng Zhao; software, Zhizhong Xing, Shuanfeng Zhao and Xiaojun Guo; validation, Zhizhong Xing, Shuanfeng Zhao and Wei Guo; formal analysis, Zhizhong Xing, Shuanfeng Zhao and Yuan Wang; investigation, Zhizhong Xing, Shuanfeng Zhao and Wei Guo; resources, Zhizhong Xing, Shuanfeng Zhao and Wei Guo; data curation, Zhizhong Xing and Shuanfeng Zhao; writing—original draft preparation, Zhizhong Xing and Shuanfeng Zhao; writing—review and editing, Zhizhong Xing, Shuanfeng Zhao and Wei Guo; supervision, Zhizhong Xing and Shuanfeng Zhao. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the National Key R&D Program of China (Grant No. 2017YFC0804310); the Key R&D Projects of Shaanxi Province (Grant No. 2020ZDLGY04-05); and the Key R&D Projects of Shaanxi Province (Grant No. 2020ZDLGY04-06).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

**Acknowledgments:** The authors thank the editor for the editing assistance. Lastly, the authors would like to thank the reviewers for their valuable comments and suggestions on an earlier version of our manuscript.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest.

## References

- Zheng, X.; He, X.; Yang, X.; Ma, H.; Yu, Z.; Ren, G.; Li, J.; Zhang, H.; Zhang, J. Terrain Point Cloud Assisted GB-InSAR Slope and Pavement Deformation Differentiate Method in an Open-Pit Mine. *Sensors* **2020**, *20*, 2337. [[CrossRef](#)]
- Ilija, L.; Loupasakis, C.; Tsangaratos, P. Land subsidence phenomena investigated by spatiotemporal analysis of groundwater resources, remote sensing techniques, and random forest method: The case of Western Thessaly, Greece. *Environ. Monit. Assess.* **2018**, *190*, 623. [[CrossRef](#)]
- Tziachris, P.; Aschonitis, V.; Chatzistathis, T.; Papadopoulou, M.; Doukas, I.D. Comparing Machine Learning Models and Hybrid Geostatistical Methods Using Environmental and Soil Covariates for Soil pH Prediction. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 276. [[CrossRef](#)]
- Fekete, S.; Diederichs, M.; Lato, M. Geotechnical and operational applications for 3-dimensional laser scanning in drill and blast tunnels. *Tunn. Undergr. Space Technol.* **2010**, *25*, 614–628. [[CrossRef](#)]
- Yang, X.; Huang, Y.; Zhang, Q. Automatic Stockpile Extraction and Measurement Using three-dimensional Point Cloud and Multi-Scale Directional Curvature. *Remote Sens.* **2020**, *12*, 960. [[CrossRef](#)]
- Ignjatović Stupar, D.; Rošer, J.; Vulić, M. Investigation of Unmanned Aerial Vehicles-Based Photogrammetry for Large Mine Subsidence Monitoring. *Minerals* **2020**, *10*, 196. [[CrossRef](#)]
- Pal, A.; Rošer, J.; Vulić, M. Surface Subsidence Prognosis above an Underground Longwall Excavation and Based on three-dimensional Point Cloud Analysis. *Minerals* **2020**, *10*, 82. [[CrossRef](#)]
- Leśniak, A.; Śledź, E.; Mirek, K. Detailed Recognition of Seismogenic Structures Activated during Underground Coal Mining: A Case Study from Bobrek Mine, Poland. *Energies* **2020**, *13*, 4622. [[CrossRef](#)]
- Tachella, J.; Altmann, Y.; Mellado, N.; McCarthy, A.; Tobin, R.; Buller, G.; Tournet, J.; McLaughlin, S. Real-time three-dimensional reconstruction from single-photon lidar data using plug-and-play point cloud denoisers. *Nat. Commun.* **2019**, *10*, 1–6. [[CrossRef](#)]
- Alsadik, B. Ideal Angular Orientation of Selected 64-Channel Multi Beam Lidars for Mobile Mapping Systems. *Remote Sens.* **2020**, *12*, 510. [[CrossRef](#)]
- Kim, H.-S.; Sun, C.-G.; Kim, M.; Cho, H.-I.; Lee, M.-G. GIS-Based Optimum Geospatial Characterization for Seismic Site Effect Assessment in an Inland Urban Area, South Korea. *Appl. Sci.* **2020**, *10*, 7443. [[CrossRef](#)]
- Cabrera-Barona, P. Influence of Urban Multi-Criteria Deprivation and Spatial Accessibility to Healthcare on Self-Reported Health. *Urban Sci.* **2017**, *1*, 11. [[CrossRef](#)]
- Zięba-Kulawik, K.; Skoczylas, K.; Mustafa, A.; Wężyk, P.; Gerber, P.; Teller, J.; Omrani, H. Spatiotemporal Changes in three-dimensional Building Density with LiDAR and GEOBIA: A City-Level Analysis. *Remote Sens.* **2020**, *12*, 3668. [[CrossRef](#)]
- Prokop, M.; Shaikh, S.A.; Kim, K.-S. Low Overlapping Point Cloud Registration Using Line Features Detection. *Remote Sens.* **2020**, *12*, 61. [[CrossRef](#)]
- Su, T.; Wang, W.; Lv, Z.; Wu, W.; Li, X. FRapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve. *Comput. Graph.* **2016**, *54*, 65–74. [[CrossRef](#)]
- Liu, N.; Lin, B.; Lv, G.; Zhu, A.; Zhou, L. A Delaunay triangulation algorithm based on dual-spatial data organization. PFG—Journal of Photogrammetry. *Remote Sens. Geoinf. Sci.* **2019**, *87*, 19–31.
- Zhao, Y.; Shi, C.; Kwon, K.; Piao, Y.; Piao, M.; Kim, N. Fast calculation method of computer-generated hologram using a depth camera with point cloud gridding. *Opt. Commun.* **2018**, *411*, 166–169. [[CrossRef](#)]
- Dey, T.; Wang, L. Voronoi-based feature curves extraction for sampled singular surfaces. *Comput. Graph.* **2013**, *37*, 659–668. [[CrossRef](#)]
- Shi, P.; Ye, Q.; Zeng, L. A Novel Indoor Structure Extraction Based on Dense Point Cloud. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 660. [[CrossRef](#)]
- Tong, G.; Li, Y.; Zhang, W.; Chen, D.; Zhang, Z.; Yang, J.; Zhang, J. Point Set Multi-Level Aggregation Feature Extraction Based on Multi-Scale Max Pooling and LDA for Point Cloud Classification. *Remote Sens.* **2019**, *11*, 2846. [[CrossRef](#)]

21. Zhou, T.; Popescu, S.; Malambo, L.; Zhao, K.; Krause, K. From LiDAR Waveforms to Hyper Point Clouds: A Novel Data Product to Characterize Vegetation Structure. *Remote Sens.* **2018**, *10*, 1949. [[CrossRef](#)]
22. Li, K.; Shao, J.; Guo, D. A Multi-Feature Search Window Method for Road Boundary Detection Based on LIDAR Data. *Sensors* **2019**, *19*, 1551. [[CrossRef](#)] [[PubMed](#)]
23. Xu, Z.; Zhang, Z.; Zhong, R.; Chen, D.; Sun, T.; Deng, X.; Li, Z.; Qin, C.-Z. Content-Sensitive Multilevel Point Cluster Construction for ALS Point Cloud Classification. *Remote Sens.* **2019**, *11*, 342. [[CrossRef](#)]
24. Qian, Z.; Liu, X.; Tao, F.; Zhou, T. Identification of Urban Functional Areas by Coupling Satellite Images and Taxi GPS Trajectories. *Remote Sens.* **2020**, *12*, 2449. [[CrossRef](#)]
25. Huang, T.; Zhao, S.; Geng, L.; Xu, Q. Unsupervised Monocular Depth Estimation Based on Residual Neural Network of Coarse–Refined Feature Extractions for Drone. *Electronics* **2019**, *8*, 1179. [[CrossRef](#)]
26. Petroșanu, D.-M.; Căruțașu, G.; Căruțașu, N.L.; Pirjan, A. A Review of the Recent Developments in Integrating Machine Learning Models with Sensor Devices in the Smart Buildings Sector with a View to Attaining Enhanced Sensing, Energy Efficiency, and Optimal Building Management. *Energies* **2019**, *12*, 4745. [[CrossRef](#)]
27. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Review: Deep Learning on three-dimensional Point Clouds. *Remote Sens.* **2020**, *12*, 1729. [[CrossRef](#)]
28. Pastucha, E.; Puniach, E.; Ścisłowicz, A.; Ćwiakała, P.; Niewiem, W.; Wiącek, P. 3D Reconstruction of Power Lines Using UAV Images to Monitor Corridor Clearance. *Remote Sens.* **2020**, *12*, 3698. [[CrossRef](#)]
29. Hu, X.; Yuan, Y. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. *Remote Sens.* **2016**, *8*, 730. [[CrossRef](#)]
30. Zhao, R.; Pang, M.; Wang, J. Classifying airborne LiDAR point clouds via deep features learned by a multi-scale convolutional neural network. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 960–979. [[CrossRef](#)]
31. Politz, F.; Sester, M. Exploring ALS and DIM Data for Semantic Segmentation Using CNNs. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. ISPRS Arch.* **2018**, *42*, 347–354. [[CrossRef](#)]
32. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for three-dimensional Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.
33. Young, M.; Pretty, C.; Agostinho, S.; Green, R.; Chen, X. Loss of Significance and Its Effect on Point Normal Orientation and Cloud Registration. *Remote Sens.* **2019**, *11*, 1329. [[CrossRef](#)]
34. Mirsu, R.; Simion, G.; Căleanu, C.D.; Pop-Calimanu, I.M. A PointNet-Based Solution for three-dimensional Hand Gesture Recognition. *Sensors* **2020**, *20*, 3226. [[CrossRef](#)] [[PubMed](#)]
35. Gamal, A.; Wibisono, A.; Wicaksono, S.B.; Abyan, M.A.; Hamid, N.; Wisesa, H.A.; Jatmiko, W.; Ardhianto, R. Automatic LIDAR building segmentation based on DGCNN and euclidean clustering. *J. Big Data* **2020**, *7*, 1–18. [[CrossRef](#)]
36. Zhang, J.; Hu, X.; Dai, H.; Qu, S. DEM Extraction from ALS Point Clouds in Forest Areas via Graph Convolution Network. *Remote Sens.* **2020**, *12*, 178. [[CrossRef](#)]
37. Lee, J.; Chung, J.; Cho, M.; Timilsina, S.; Sohn, K.; Kim, J.; Sohn, K. Deep-Learning Technique To Convert a Crude Piezoresistive Carbon Nanotube-Ecoflex Composite Sheet into a Smart, Portable, Disposable, and Extremely Flexible Keypad. *ACS Appl. Mater. Interfaces* **2018**, *10*, 20862–20868. [[CrossRef](#)]
38. Kim, H.; Kim, C. Deep-Learning-Based Classification of Point Clouds for Bridge Inspection. *Remote Sens.* **2020**, *12*, 3757. [[CrossRef](#)]
39. Morbidoni, C.; Pierdicca, R.; Paolanti, M.; Quattrini, R.; Mammoli, R. Learning from Synthetic Point Cloud Data for Historical Buildings Semantic Segmentation. *ACM J. Comput. Cult. Herit.* **2020**, *13*, 1–16. [[CrossRef](#)]
40. Pierdicca, R.; Paolanti, M.; Matrone, F.; Martini, M.; Morbidoni, C.; Malinverni, E.S.; Frontoni, E.; Lingua, A.M. Point Cloud Semantic Segmentation Using a Deep Learning Framework for Cultural Heritage. *Remote Sens.* **2020**, *12*, 1005. [[CrossRef](#)]
41. Belkina, A.C.; Ciccolella, C.O.; Anno, R.; Halpert, R.; Spidlen, J.; Snyder-Cappione, J.E. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nat. Commun.* **2019**, *10*, 1–12. [[CrossRef](#)] [[PubMed](#)]
42. Zhao, S.; Han, G.; Zhao, Q.; Wei, P. Prediction of Driver’s Attention Points Based on Attention Model. *Appl. Sci.* **2020**, *10*, 1083. [[CrossRef](#)]
43. Cortiñas-Lorenzo, B.; Pérez-González, F. Adam and the Ants: On the Influence of the Optimization Algorithm on the Detectability of DNN Watermarks. *Entropy* **2020**, *22*, 1379. [[CrossRef](#)] [[PubMed](#)]
44. Bala, P.C.; Eisenreich, B.R.; Yoo, S.B.M.; Hayden, B.Y.; Park, H.S.; Zimmermann, J. Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio. *Nat. Commun.* **2020**, *11*, 1–12. [[CrossRef](#)] [[PubMed](#)]
45. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
47. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph crn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [[CrossRef](#)]
48. He, C.; Li, S.; Xiong, D.; Fang, P.; Liao, M. Remote Sensing Image Semantic Segmentation Based on Edge Information Guidance. *Remote Sens.* **2020**, *12*, 1501. [[CrossRef](#)]
49. Guo, Y.; Chen, G.; Mo, R.; Wang, M.; Bao, Y. Benefit Evaluation of Water and Soil Conservation Measures in Shendong Based on Particle Swarm Optimization and the Analytic Hierarchy Process. *Water* **2020**, *12*, 1955. [[CrossRef](#)]

- 
50. Ji, X.; Song, D.; Zhao, H.; Li, Y.; He, K. Experimental Analysis of Pore and Permeability Characteristics of Coal by Low-Field NMR. *Appl. Sci.* **2018**, *8*, 1374. [[CrossRef](#)]
  51. Benjdira, B.; Bazi, Y.; Koubaa, A.; Ouni, K. Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images. *Remote Sens.* **2019**, *11*, 1369. [[CrossRef](#)]
  52. Li, Y.; Shi, T.; Zhang, Y.; Chen, W.; Wang, Z.; Li, H. Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 20–33. [[CrossRef](#)]