Journal of Globe Scientific Reports (2024)

Journal of Globe Scientific Reports

journal homepage: www.journal-gsr.online

Paper Type: Original Article

Deep learning-based spatial vector point set partition model

Yulin Chen[1,#], Junwen Chen[2,#], Yuhong Xiao[3,#], CHENG SUNNY[4,&], Muxin Shi[2,&] , Jiahao Li[2,*]

1.University College London, London, United Kingdom

2.Shenzhen Aurora International Academy, Shenzhen, China

3.Shenzhen Senior High School Group International Division, Shenzhen, China

4.The Affiliated International School of Shenzhen University, Shenzhen, China

#Co-first author

&Co-second author

*Corresponding author

Abstract

In the era of big data, the scale and complexity of spatial data are continuously increasing, making the effective partitioning and classification of spatial vector point sets a critical and challenging problem. This study proposes a spatial vector point set partitioning model based on deep learning, which leverages techniques such as Convolutional Neural Networks (CNN) and PointNet. Through an end-to-end learning process, the model automatically extracts the intrinsic structures and patterns of the data. The method employs 3D-CNN and PointNet models to process point cloud data, achieving efficient and accurate partitioning results. The findings indicate that the model demonstrates stronger robustness and higher accuracy when handling large-scale, high-dimensional data, with a classification accuracy reaching 100%. In conclusion, the spatial vector point set partitioning model based on deep learning holds significant theoretical and practical value, offering more precise and reliable technical support for related fields.

Keywords: Deep learning, point set division, 3D-CNN, Point net, Point net

1 | Introduction

With the rapid development of information technology, we are now in a data-driven era. The application of various sensors and data acquisition devices leads to the increasing scale and complexity of spatial data, which is not only explosive in quantity, but also more diverse in type and structure. As an important form of spatial data, spatial vector point set has a wide range of applications, covering many fields such as geographic information system (GIS), remote sensing image analysis, robot path planning, computer vision and so on. These application fields put higher requirements for the processing and analysis of spatial data, especially in the division and classification of data. Traditional spatial data partitioning methods, such as K-mean clustering, hierarchical clustering, and density-based clustering methods (e. g., DBSCAN), perform well when handling small-scale data. However, with the increasing data scale and increasing data dimension, these traditional methods face problems such as low computational efficiency, long processing time and unstable clustering effect. Moreover, noise and outliers in high dimensional spatial data also have a significant impact on the effects of traditional clustering methods. The existence of these problems limits the application of traditional methods to large-scale, high-dimensional spatial data.

In this context, the rapid development of deep learning techniques provides new ideas for solving the problem of spatial vector point set division. Deep learning models, especially convolutional neural network (CNN) and autoencoder (Autoencoder), have achieved remarkable results in image classification, object detection and semantic segmentation. These models are able to automatically learn complex features of the data and perform efficient feature extraction. Applied deep learning techniques to the division of spatial data can show greater robustness and higher accuracy when processing large-scale, high-dimensional data. The spatial vector point set division model based on deep learning can not only make use of the powerful feature extraction ability of deep neural network, but also automatically learn the intrinsic structure and pattern of data through the end-to-end learning process. This approach is expected to overcome the limitations of traditional methods and achieve efficient division and accurate classification of complex spatial data. For example, in remote sensing image analysis, deep learning model can be trained to identify and classify different surface features; in robot path planning, the model can help the robot find the optimal path in complex environment. In addition, the flexibility and scalability of deep learning models also enable them to be customized and optimized according to the requirements of different application scenarios.

The research and development of the spatial vector point set division model based on deep learning has both important theoretical significance and extensive practical application value. By deeply exploring the potential of deep learning in spatial data processing, it can promote the development of spatial data analysis technology and provide innovative solutions for multiple application scenarios. This can not only improve the efficiency and accuracy of existing systems, but also explore new application areas, such as intelligent transportation systems, environmental monitoring, urban planning, etc. Deep learning has promising applications in the field of spatial vector point set division. As technology continues to advance and innovate, it is reasonable to believe that deep learning will play an increasingly important role in spatial data analysis in the future. For example, through deep learning model, urban traffic flow can be predicted more accurately and traffic signal control is optimized; in environmental monitoring, the distribution and diffusion of pollutants can be identified and monitored more effectively; in urban planning, land use planning and resource allocation can be conducted more scientifically. These applications can not only improve the intelligent level of urban management, but also provide residents with a higher quality of living environment. In short, the application of deep learning technology in spatial vector point set division can not only solve the challenges encountered by traditional methods in dealing with large-scale and high-dimensional data, but also provide more accurate and reliable technical support for applications in related fields. With the continuous development of technology, the application of deep learning in spatial data analysis will be more extensive, and its potential and value will be continuously explored and realized.

2 | Related Works

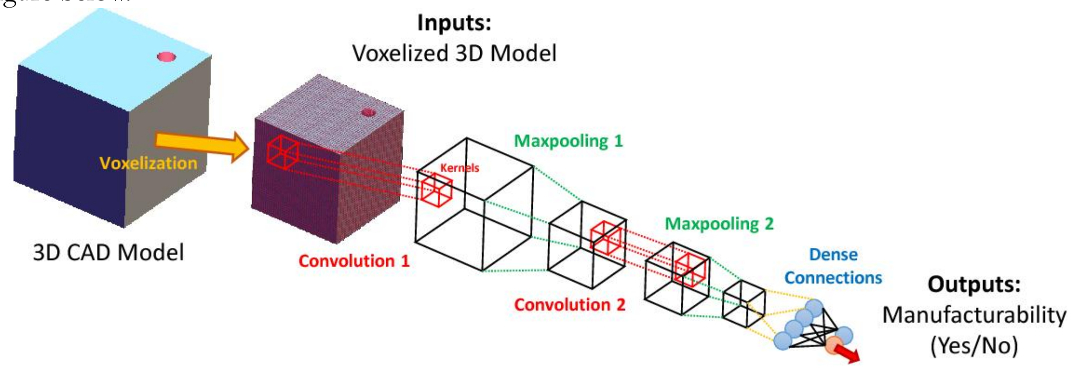
In the field of spatial vector point set division, deep learning methods have received extensive attention and research recently. Yu Xijun and Duan Yong [1] proposed a 3 D point cloud classification method based on PointCloudTransformer and optimized integration learning, which shows good performance in improving classification accuracy and efficiency. Aida, Zhang Xiaoyang, Xu Ce and other [2] summarized the new progress of large-scale point cloud deep learning semantic segmentation methods, and introduced the performance of various latest technologies and algorithms in practical application in detail. Jia Mingda, Yang Jinming, Meng Weiliang and other [3] discussed the environmental target detection technology of fusion point cloud and image, and proposed a multi-mode fusion method, which effectively improved the accuracy and reliability of target detection. Zhang Yi Yao [4] proposed a new target detection framework based on 3 D convolutional neural network to verify its superior performance in complex environment. In the field of point cloud object tracking, [5] studied an algorithm based on deep learning and proposed the method of multi-scale feature fusion, which significantly improves the tracking accuracy and robustness. Chen Shaojin [6] proposed a deep learning method based on geometric features for semantic segmentation of airborne point clouds, and verifies its superior performance in complex terrain through experiments. Lu Bin and Wang Zhiyuan [7] studied the regional growth point cloud segmentation method combining supervoxel and color information, and proposed a new regional growth algorithm, which effectively improves the accuracy and efficiency of segmentation. Zhang Dongdong, Guo Jie, Chen Yang [8] combined deep learning and integrated learning to propose a new point cloud object recognition method, showing its wide applicability and efficiency in different scenarios. Guo Dawei, Li Jinghao and Lu Jun [9] proposed the multi-scale deep learning method of three-dimensional point cloud registration, and verified its high efficiency and accuracy in various registration tasks through experiments. Jiang Peiqi, Wu Jie, Zhang Shirong and other [10] studied the point cloud segmentation and flood risk simulation method based on deep learning, and proposed a new segmentation and simulation framework, which showed good application prospects in flood risk assessment. Shu Jun et al. [11] proposed a multi-modal point cloud classification network based on the residual multi-layer perceptron (MLP), which enhances the classification performance by integrating multiple features. Their research provides new perspectives for processing multi-source point cloud data. Then, Liu Hui et al. [12] proposed a method based on lightweight PointNet network for the real-time identification of targets in the agricultural field, especially in the forest orchard spray operation. This work not only improves the real-time performance of the identification, but also reduces the demand for computational resources through a lightweight design. Liu Origin and other [13] further explored the field of point cloud semantic segmentation, and they proposed a network structure combining feature fusion and loss optimization, aiming to improve the segmentation accuracy. This study provides an effective solution for point cloud segmentation tasks by optimizing the loss function and feature fusion strategy. In terms of point cloud registration technology, Liang Jietao et al. [14] proposed a point cloud registration method based on coordinate geometry sampling. Their research improves the efficiency and accuracy of registration through innovative sampling techniques. Although the details of the literature are not currently available, this work has the important implications for the development of point cloud processing technologies based on the summary provided. Finally, Xu Jie et al. [15] conducted research on the classification and segmentation of tree point cloud in nurseries. They improved the classification and segmentation ability of tree point cloud data based on the improved PointNet + + model. This study has potential applications for forestry resource management and ecological environment monitoring.

The main work of this paper is to develop and evaluate a deep learning-based spatial vector point set division model, which uses 3D-CNN and Point net technologies to automatically extract the intrinsic structure and patterns of point cloud data, realizing efficient spatial data division and classification. The results show that the model has excellent robustness and accuracy in processing large-scale and high-dimensional data, and the classification accuracy reaches 100%, providing innovative technical support for GIS, remote sensing image analysis and other application fields, and demonstrating the broad application prospect of deep learning in spatial data processing.

3 | Theory and Method

3.1 | 3DCNN model

3DConvolutional neural networks (3 DCNN) are used to process three-dimensional spatial data, such as 3D images, videos, and point cloud data. Its basic operations include 3D convolution, activation function, pooling and full connection layer. The specific structure is shown in the figure below.



$X \in \mathbb{R}^{D\times H\times W\times C} DHWC$Let the input data be a 3 D matrix, where the depth, the height, the width, and the number of channels (for example, the number of channels in the RGB image is 3). The 3D convolutional layer slides over the input data through a 3 D convolution kernel (filter) for the convolution operation. Let the convolution kernel be, where is the depth, height, and width of the convolution kernel, the number of input channels, and the number of output channels.$K \in \mathbb{R}^{k_D\times k_H\times k_W\times C\times C'} k_D, k_H, k_W C C'$ In the 3D convolution, the convolution kernel slides in the depth, height, and width direction of the input data. For each position, each value of the convolution kernel was multiplied by the corresponding input data value and then summed to obtain an output value. This process is repeated throughout the input data to generate output feature maps. The convolution operation formula is as follows:

$$Y_{i,j,k,c'} = \sum_{c=1}^{C}\sum_{d=1}^{k_D}\sum_{h=1}^{k_H}\sum_{w=1}^{k_W} X_{i+d-1,j+h-1,k+w-1,c} \cdot K_{d,h,w,c,c'} + b_{c'}$$

YAmong them, it is the convolution output, which is the bias.$b_{c'} X_{i+d-1,j+h-1,k+w-1,c}$: Value of the data in the first channel.$(i+d-1, j+h-1, k+w-1) c K_{d,h,w,c,c'} (d,h,w) c c'$: The weight between the convolution kernel in, position, the first input channel and the first output channel.$Y_{i,j,k,c'}$: Outputs the feature map in the position, the value of the first channel.$(i,j,k) c'$ The convolution operation can be seen as applying a 3 D convolution kernel to each local region of the input data and calculating the weighted sum at each position. The activation function is used to introduce non-linearity, enabling the network to learn more complex features. Common activation functions include the ReLU (Rectified Linear Unit). The ReLU activation function formula:

$$Y_{i,j,k,c'} = \max\left(0, Y_{i,j,k,c'}\right)$$

ReLU sets the negative value to 0 and preserves the positive values, increasing the nonlinear representation power of the model. The pooling layer is used to reduce the spatial dimension of the data while retaining important features. Common pooling operations include maximum pooling (max pooling) and average pooling (average pooling). The size of the pooling core is the maximum pooling operation formula:$p_D \times p_H \times p_W$。

$$Y_{i,j,k,c'} = \max_{d,h,w} X_{i+d-1,j+h-1,k+w-1,c'}$$

The pooling operation reduces the size of the data by taking the maximum or average within each pooling region while retaining the main features. The fully connected layer flatten the multi-dimensional feature graph into a one-dimensional vector and linearly transforms the weight matrix and bias. Full connection layer formula:

$$z = W \cdot x + b$$

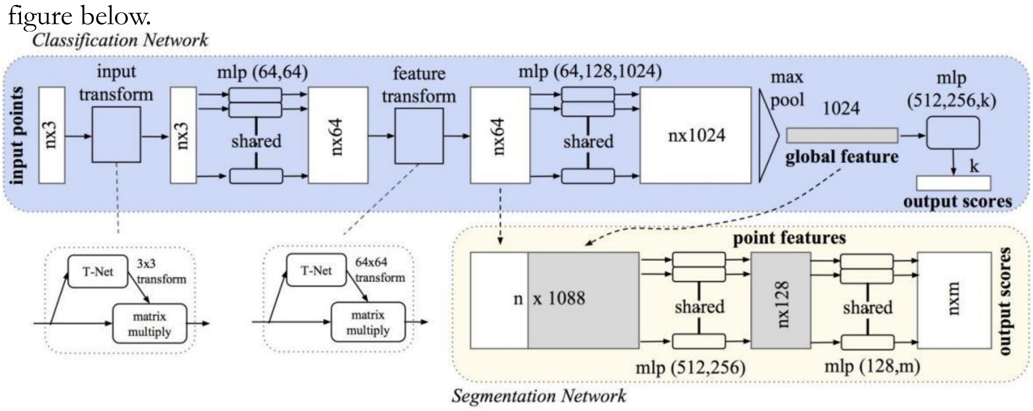
xWb Among them, is the input vector after leveling, is the weight matrix, is the bias. The fully connected layer enables further combination and abstraction of features through linear transformation and activation function. For the classification task, the last layer is usually the Softmax layer, converting the network output into class probabilities. Softmax Formula:

$$p_i = \frac{\exp(z_i)}{\sum_{j=1}^{N} \exp(z_j)}$$

$Np_i i$ Where, is the number of categories, is the probability of the first category. The Softmax layer transforms the output of the model into a probability distribution suitable for multi-category classification problems.3D convolutional neural networks effectively extract spatial features in 3 D data by performing convolution operations in 3 D space. Its main steps include 3D convolution, activation function, pooling, full connectivity, and Softmax layers. Through these steps, 3D CNN can process 3 D images, videos and point cloud data, realizing tasks such as classification, segmentation and detection. Detailed convolution and pooling operation formulas show how the 3 DCNN slides convolutional and pooling kernels on 3 D data to extract and aggregate features.

3.2 | PointNet model

PointNet Is a deep learning network structure for processing point cloud data, built by Charles R. As presented in 2017 by Qi et al. Its core idea is to directly feature extraction from each point in the point cloud, and then aggregate the features of all points through global pooling operation to achieve a unified description of the whole point cloud. The advantage of PointNet is its ability to handle irregular point cloud data without any complex data preprocessing, such as voxel or grid, which gives it high flexibility and efficiency in handling tasks such as 3D object recognition, classification and segmentation. Moreover, the design of PointNet allows it to capture the global structural information of point cloud data, which is essential for understanding complex 3D scenarios. Although PointNet may encounter performance bottlenecks when dealing with large-scale point clouds, its innovative network structure and processing methods lay the foundation for subsequent point cloud deep learning research. The specific structure is shown in the figure below.



Let the input point cloud be, for each point. $P = \{p_1, p_2, \ldots, p_n\} p_i \in \mathbb{R}^3$ Then, to make the model insensitive to the arrangement of point clouds, the input point clouds are first geometrically transformed. The transformation is learned from a small neural network (TNet):

$$T(P) = T(\{p_1, p_2, \ldots, p_n\}) \in \mathbb{R}^{3\times3}$$

The input-transformed point cloud is:

$$p_i^{'} = T \cdot p_i$$

$p_i^{'}$For each transformed point, a multilayer perceptron (MLP) was applied to extract the features:

$$f_i = \mathrm{MLP}\left(p_i^{'}\right) = \sigma\left(W_2 \cdot \sigma\left(W_1 \cdot p_i^{'} + b_1\right) + b_2\right)$$

$W_1, W_2 b_1, b_2 \sigma$Where, is the weight matrix, is biased, is the activation function (such as ReLU). Use symmetric functions (such as max pooling) to aggregate features for each point into global features:

$$g = \text{MAX}\ (f_1, f_2, \ldots, f_n)$$

Where, MAX is the operation that takes the maximum value for each feature dimension. Further transform the global features to ensure that they are invariant to the rotation and translation transformations:

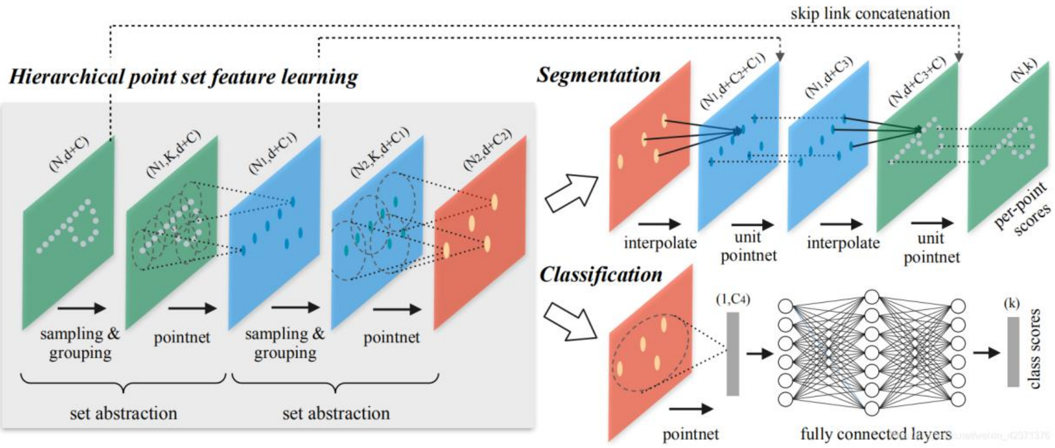
$$g' = T(g) \in \mathbb{R}^k$$

Classify the global features or segment the features at each point:

$$\text{Output} = \text{MLP}\left(g'\right)$$

3.3 | PointNet++ model

PointNet + + is an extension and improvement of the original PointNet architecture, proposed by Qi et al. in 2017. It aims to solve the problem of PointNet in processing local structural information, and enhance the capture of local convolution features by introducing the concept of graph convolutional network (GCN). The core idea of PointNet + + is to divide the point cloud data into multiple local regions, then perform local feature extraction within each local region, and finally combine these local features with global features to obtain a more comprehensive representation of the point cloud. PointNet + + achieves this through a strategy called "sampling + aggregation". First, it randomly samples a set of points as the central points of the region, and then, for each central one, finds its k-nearest-neighbor points to form a local region. Then, the feature of each local region is extracted using a shared MLP (multi-layer perceptron), a process called aggregation. Through multiple sampling and aggregation, PointNet + + is able to capture the local features of point clouds from different scales. Another important feature of PointNet + + is its ability to handle point cloud data with different densities, because its local feature extraction does not depend on a fixed grid structure. This flexibility allows PointNet + + to perform well in handling a variety of 3D shapes and scenes, whether in classification, segmentation, or detection tasks. Overall, PointNet + + provides a powerful and efficient method for deep learning analysis of point cloud data, and the specific structure is shown in the figure below.



$lP_l = \{p_1, p_2, \ldots, p_{n_l}\} P_{i,l}$ Let the point cloud data of the first layer be. For each local region, the PointNet was applied for feature extraction:

$$f_{i,l} = \text{PointNet}\ (P_{i,l})\ \forall i \in \{1,2,\ldots,n_l\}$$

$P_{i,l}$Specifically, for each local region, T-Net was first applied to transform the region:

$$T(P_{i,l}) = T(\{p_{i,1}, p_{i,2}, \ldots, p_{i,k}\}) \in \mathbb{R}^{3\times 3}$$

Transformed point cloud:

$$p'_{i,j} = T \cdot p_{i,j}$$

$p'_{i,j}$The MLP is then applied to each point to extract the features:

$$f_{i,j} = \text{MLP}\ \left(p'_{i,j}\right)$$

Then use the max pooling to aggregate the local features:

$$f_{i,l} = \mathrm{MAX}\left(f_{i,1}, f_{i,2}, \ldots, f_{i,k}\right)$$

The local features of each layer are aggregated to gradually form the global features.

$$g_l = \text{ Aggregate function } \left(f_{1,l}, f_{2,l}, \ldots, f_{n_l,l}\right)$$

Feature extraction and fusion at different scales to improve the ability of the model to capture features of different granularity.

$$g = \text{ Aggregate function } (g_1, g_2, \ldots, g_L)$$

Classified or segment the fused features:

$$\text{Output} = \mathrm{MLP}(g)$$

Suppose we have a point cloud of data containing points, and the coordinates of each point are.Pnp_i Then the input transform:

$$p_i' = T \cdot p_i$$

Feature extraction was then performed:

$$f_i = \sigma\left(W_2 \cdot \sigma\left(W_1 \cdot p_i' + b_1\right) + b_2\right)$$

lowed by global feature aggregation:

$$g = \mathrm{MAX} \quad (f_1, f_2, \ldots, f_n)$$

Then perform the output transform:

$$g' = T(g)$$

Finally, make the classification / segmentation:

$$\text{Output} = \mathrm{MLP}\left(g'\right)$$

4 | Results and Discussions

4.1 | Data set visualization

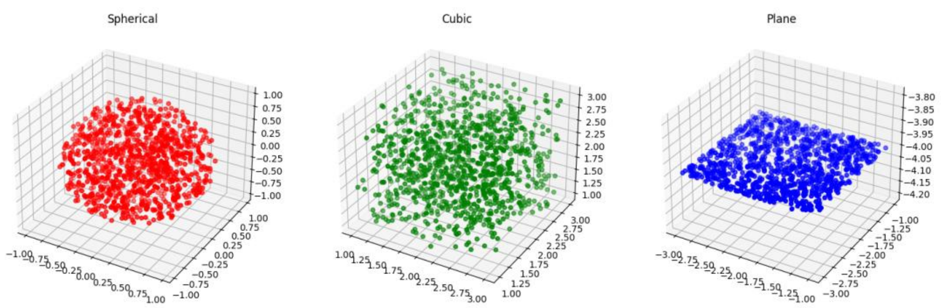
First, this paper constructs different types of spatial point cloud data, including spherical, cubic, and flat point clouds. These point cloud data are used to simulate the 3 D structures in practical applications. For each type of data, we generated point clouds using specific mathematical methods:

1. Spherical point cloud: points that are evenly distributed on the sphere. Using the spherical coordinate system, the points on the sphere were converted to Cartesian coordinates.

2. Cube point cloud: points that are evenly distributed within the cube. The coordinates of the randomly generated points are within the boundaries of the cube.

3. Planar point clouds: points that are evenly distributed on a plane. Compliance to the plane equation by generating a random point and adjusting its position.

Subsequently, we used the 3D mapping function in the `matplotlib` library to visualize these point cloud data. Each type of point cloud data was assigned a different color to differentiate in three dimensions, as shown in the figure below.



4.2 | The voxel of the data

The voxization of data is a technique for converting scattered point cloud data into a regular 3 D grid structure. This approach first requires the creation of a grid of voxels with a fixed size, consisting of many small cubes, each called a voxel. The initialization of the voxel mesh is usually a full zero array, where each voxel represents a volume cell in the grid, and the initial values of all voxels are set to zero. During voxelization, each point in a point cloud first needs to map its coordinates within the size range of the voxel mesh through a normalization operation. The normalized coordinates were then converted to an integer index in the grid, ensuring that each point could be accurately positioned to the corresponding voxel.

Next, each point is mapped to the corresponding position in the voxel grid by walking through all the points in the point cloud, and the count value for that location is updated. This process involves rounding the coordinates of each point or other forms of quantification to ensure that they can exactly correspond to the integer coordinates in the grid. As the traversal proceeds, the voxel grid is gradually populated, and the value of each voxel reflects the number of its internal points, thus forming a voxelized representation of the point cloud data. This representation not only provides a regular geometry for point cloud data, but also enables subsequent 3D-CNN models to process the data more efficiently, as these models often require regular input formats for convolutional operations and other deep learning tasks.

The advantage of voxelization technology lies in its ability to transform the irregularities and disorder of point cloud data into a regular grid structure, which makes the data more suitable for the processing of deep learning models. The 3D-CNN model enables this structured data for feature extraction and pattern recognition. By applying the convolution operation on the voxelized data, the model can learn the spatial features that capture both local and global information in the point cloud data. Furthermore, voxelization also helps to reduce the storage requirements of the data, as it converts continuous point clouds into discrete voxels, each of which only needs to store limited information. While improving the computational efficiency, it also provides a new perspective for the analysis and understanding of point cloud data.

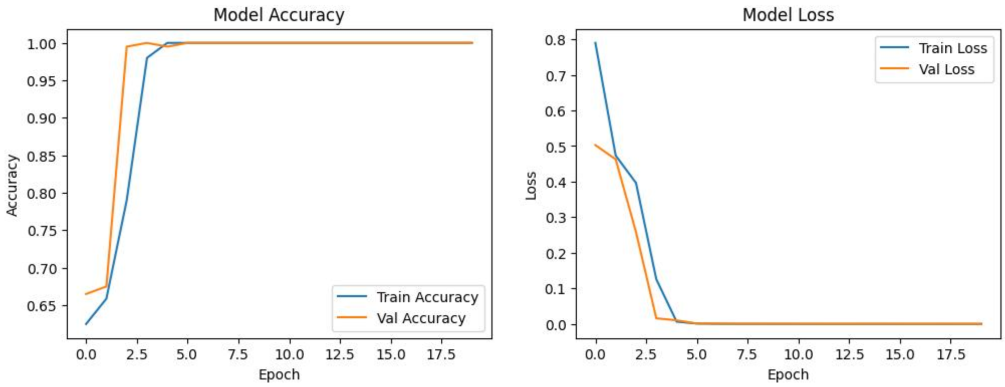
4.3 | 3 Point-set partitioning of the D-CNN model

The constructed 3D convolutional neural network (3D-CNN) first accepts as input the voxelized point cloud data, which is organized into a three-dimensional grid of fixed size. The preliminary convolutional layer of the network uses 32 convolution kernels of 3x3x3 to extract the basic spatial features in the input data and introduce a nonlinearity through the ReLU activation function. Subsequently, a maximum pooling layer of 2x2x2 was used to reduce the spatial dimension of the feature map and to reduce the computational complexity while retaining important feature information. The following convolution layer further extracts the deeper features using 64 convolution kernels of the same size. Apply the maximum pooling layer again to further reduce the spatial dimension. Subsequently, the network continues to process the data with 128 convolutional kernels, capturing more complex spatial relationships and further squeezing the feature graph through the maximum pooling layer. The leveling layer levels the multi-dimensional feature graph into a one-dimensional array for input to the fully connected layer. The fully connected layer combines these extracted features and is further processed through 512 neurons, and finally the output layer generates a probability distribution for each class using the Softmax activation function. This network structure allows the model to efficiently extract and learn spatial features from the voxel grid data, ultimately achieving efficient classification prediction, and the specific network structure is shown in the table below.

Layer (type)	Output Shape	Param
conv3d_4 (Conv3D)	(None, 30, 30, 30, 32)	896
max_pooling3d_3 (MaxPooling3D)	(None, 15, 15, 15, 32)	0
conv3d_5 (Conv3D)	(None, 13, 13, 13, 64)	55,360
max_pooling3d_4 (MaxPooling3D)	(None, 6, 6, 6, 64)	0

conv3d_6 (Conv3D)	(None, 4,4,4,128)	221,312
max_pooling3d_5 (MaxPooling3D)	(None, 2, 2, 2, 128)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524,800
dense_3 (Dense)	(None, 10)	5,130

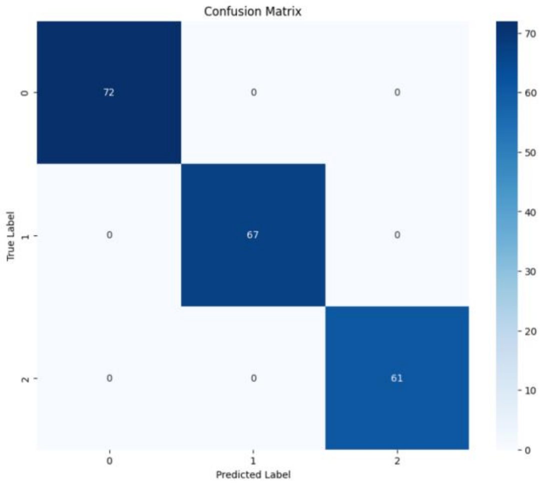
In the model training and evaluation section, we first compiled the constructed 3D-CNN model. The compilation process includes selection optimizer, loss function and evaluation metrics. In doing so, we used the Adam optimizer, an adaptive learning rate optimizer that effectively adjusts the learning rate to speed up convergence. The loss function selects `sparse _ categorical _ crossentropy`, suitable for the multi-category classification task, computing the error between the model predictions and the true labels. The evaluation metric selected `accuracy` to measure the accuracy of the model during training and validation. During the model training phase, we used the voxelized data to train the model, setting a certain number of training rounds (epochs) and batch size (batch_size). During training, the model gradually adjusts its internal weights to minimize the value of the loss function and improve the classification accuracy. To track the performance during training, we recorded the accuracy and loss values of each round and plot these metrics varying with training rounds. In this way, the learning progress and performance changes of the model during the training process can be intuitively observed, as shown in the figure below.



In the model evaluation phase, we evaluated the trained model using the test set. By calculating and displaying the classification report of the model, we can obtain the precision, recall, and F1 score for each category, as shown in the following table.

performance index	Accuracy	Precision	Recall	F1 Score
numeric value	1	1	1	1

In addition, we generated a confusion matrix to visualize the classification effect of the model on individual categories. The confusion matrix shows the match between the model prediction results and the actual labels, helping to analyze the classification performance of the model and identify possible misclassification patterns. These evaluation results provide a comprehensive understanding of the model performance, which will help us to further optimize and adjust the model.



Confusion Matrix (Confusion Matrix) is a performance evaluation tool commonly used in classification problems, especially in supervised learning. It is a table for visualizing the differences between the model prediction results and the actual labels. The confusion matrix usually contains the following four main elements:

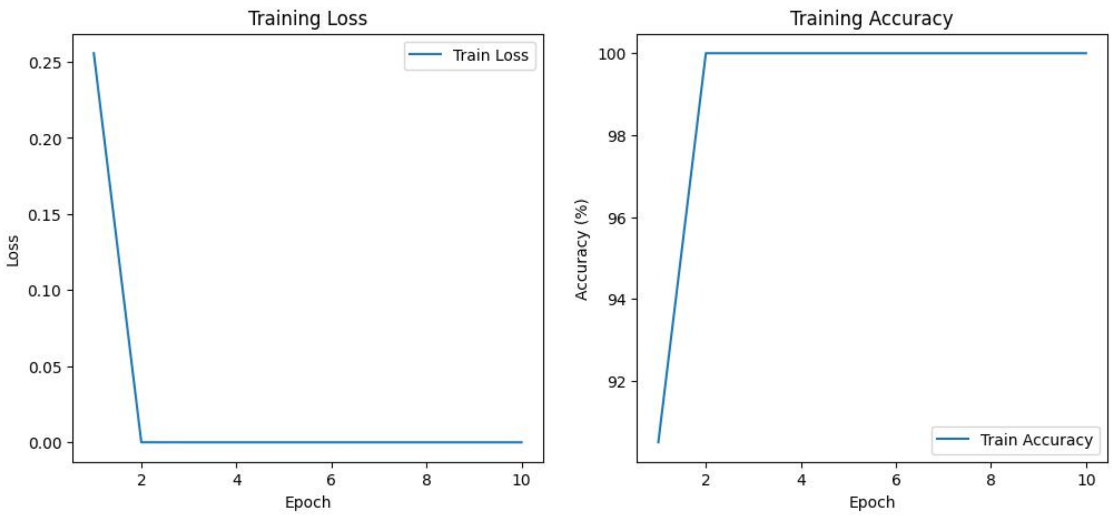
- True example (True Positives, TP): the number of samples that the model correctly predicts as positive classes. For example, in dichotomous problems, this means that the model correctly identifies samples that are actually positive as positive.
- False positive case (False Positives, FP): number of negative samples that the model incorrectly predicts as positive. This is also called a type I error or a type I error in dichotomy problems.
- True negative cases (True Negatives, TN): the number of samples that the model has correctly predicted to be negative. In the dichotomous problem, this means that the model correctly identifies samples that actually are as negative.
- False negative cases (False Negatives, FN): number of positive samples that the model falsely predicts as negative. This is also called type II error or type II error in dichotomy problems.

The confusion matrix can see that the 3D-CNN network constructed in this paper can complete the division of point cloud datasets in space.

4.4 | Point net Model point set division

The constructed PointNet model consists of three main parts: input transformation network (input transform network), feature transformation network (feature transform network), and classification network (classification network). First, the input transform network processes the input point cloud data and transforms it into a space more suitable for subsequent processing. This network consists of a series of convolutional layers and maximum pooling layers used to extract local features and maximum pooling layers to aggregate global features. Next is the feature transformation network, this part of the network is similar to the input transformation network structure, but used to transform the feature space. The output of the feature transformation network is a feature transformation matrix used to transform the local features of the input to make it more discriminative in subsequent processing. Finally, the classification network, which classifies the transformed features through a series of convolution layers, maximum pooling layers, and fully connected layers. The convolutional layer was used to further extract the features, and the fully connected layer was used for the final classification decision. The entire network structure utilizes max pooling operations to aggregate local features into global features to enable efficient classification of point cloud data. The key to the design of the network structure lies in the transformation network of input and feature, which can effectively deal with the disorder and transformation invariance problems of point cloud data.

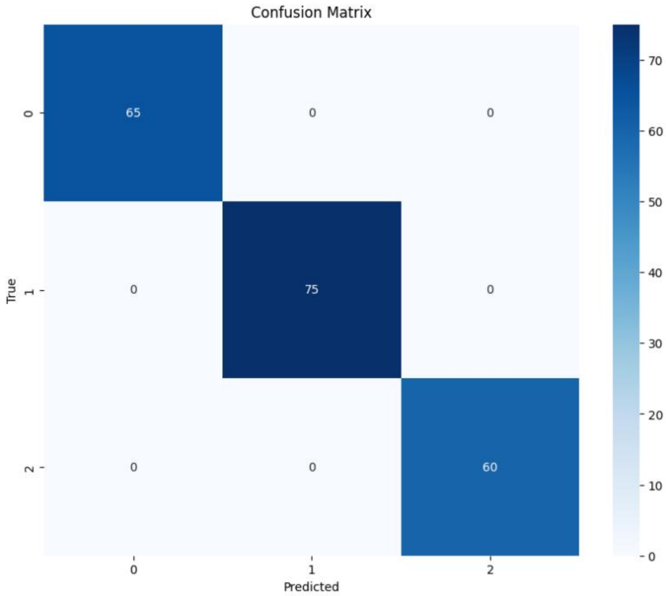
During the model training process, the PointNet model first uses the cross-entropy loss function (CrossEntropyLoss) to measure the difference between the prediction results and the true labels. The optimizer uses the Adam optimizer to minimize the loss function by adjusting the model parameters. For each training cycle (epoch), the model enters into the training mode and trathrough the training dataset. For each batch of data, the gradient was first zeroed, and then the input data was propagated forward through the PointNet model to obtain the predicted results. The loss is then calculated, and the model parameters are updated by backpropagation. At the end of each epoch, the mean loss and accuracy were calculated. During training, the average loss and accuracy of each epoch were recorded to facilitate subsequent analysis and visualization. The average loss reflects the error of the model on the training data, and the accuracy rate represents the classification performance of the model on the training data. After training, the model performance was evaluated and the training effect was visually demonstrated by calculating the confusion matrix and drawing the loss and accuracy curves. The confusion matrix shows the correct versus wrong situation of the model classification, helping to understand how the model performs on different categories. The loss curve and the accuracy curve show the change trend of the performance of the model in the training process, and help to judge whether the model converges and whether there is overfitting or underfitting phenomenon. The results are shown in the figure below.



It can be seen from the figure that the model has converged in the second generation, and the classification accuracy has reached 100%. Meanwhile, the following table also gives the final classification performance index of the model.

performance index	Accuracy	Precision	Recall	F1 Score
numeric value	1	1	1	1

It can be seen from the above classification performance index that Point net also completes the division of the point cloud data set in space, but its division speed is much faster than CNN. The following figure also gives the confusion matrix of classification.



5 | Conclusion

This study successfully constructed and evaluated two models: 3D-CNN and PointNet. The proposal of these two models aims to meet the rapid growth of the scale and complexity of spatial data in the big data era, as well as the challenges faced by traditional spatial data division methods when dealing with large-scale, high-dimensional data. The 3D-CNN model uses voxelization techniques to transform continuous point cloud data into discrete 3 D grids, thus providing structured data input to the convolutional neural network. Through multi-layer convolution and pooling operations, 3D-CNN is able to effectively extract spatial features of point cloud data and achieve high accuracy predictions in classification tasks. This process not only improves the efficiency of data processing, but also enhances the ability of the model to express the spatial features through the powerful feature learning ability of deep learning. The PointNet model uses an innovative method to deal with the disorder and transformation invariance problems of point cloud data. Through the collaborative work of the input transformation network, the feature transformation network and the classification network, PointNet can directly process the raw point cloud data without prior complex preprocessing steps. This end-to-end processing simplifies the architecture of the model while maintaining sensitivity to the complex structure of point cloud data. The cross-entropy loss function and the Adam optimizer were used for both models during model training. The cross-entropy loss function provides the standard for the model to measure the difference between the prediction and the actual labels, while the Adam optimizer accelerates the optimization process of the model parameters through adaptive learning rate adjustment. Through the back propagation algorithm, the model is able to continuously learn and adjust the internal parameters to minimize the loss function and improve the classification accuracy.

The evaluation results show that the 3D-CNN model performs well in both classification accuracy and processing speed, thanks to its efficient convolution and pooling operations, as well as the deep exploration of spatial features. However, the PointNet model shows excellent classification performance with its end-to-end processing power and its robustness to the disorder of point cloud data, and its classification accuracy also reaches 100%. This result not only validates the potential of the PointNet model for processing point cloud data, but also shows its reliability in practical applications. By analyzing the confusion matrix and performance indicators, the results further verify the validity and robustness of the proposed method. The confusion matrix clearly shows the classification effect of the model on each category, while the performance metrics comprehensively evaluate the classification performance of the model from multiple dimensions including precision, recall rate and F1 score. These evaluation methods provide an important reference basis for the optimization and improvement of the model. In conclusion, the deep learning-based spatial vector point set division model proposed in this study shows important value at both the theoretical and practical levels. These models can not only improve the efficiency and accuracy of spatial data processing, but also bring new technical ideas and solutions to the field of spatial data analysis through the application of deep learning technology. With the continuous development and optimization of technologies, these models are expected to play a greater role in intelligent transportation systems, environmental monitoring, urban planning and many other fields, promoting the innovation and development of related technologies.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Xi Jun, Duan Yong. 3D point cloud classification based on PointCloudTransformer and optimized Ensemble learning [J/OL]. Journal of electronic measurement and instrument, 1-12 [2024-07-20]. HTTP: / / http://kns.cnki.net/kcms/detail/11.2488.TN.20240624.1433.019.html.

[2] IDA, Zhang Xiaoyang, Xu Ce, et al. New advances in semantic segmentation methods based on large-scale point cloud deep learning [J]. Advances in Laser and Optoelectronics, 2019,61(12):43-60. (in Chinese)

[3] Jia Mingda, Yang Jinming, Meng Weiliang, et al. Research progress of fusion point cloud and image environmental target detection [J]. Journal of Image and Graphics, 2019,29(06):1765-1784. (in Chinese)

[4] Zhang Y Y. Research on point cloud object detection method based on 3D Convolutional neural network [D]. The northern industrial university, 2024. DOI: 10.26926 /, dc nki. Gbfgu. 2024.000121.

[5] Liu Zhaodi. The point cloud target tracking algorithm based on depth of learning research [D]. The northern industrial university, 2024. The DOI: 10.26926 /, dc nki. Gbfgu. 2024.000587.

[6] Chen Shaojin. Research on semantic segmentation of airborne point cloud in deep learning based on geometric features [D]. Disaster prevention institute of science and technology, 2024. DOI: 10.27899 /, dc nki. GFZKJ. 2024.000015.

[7] Lu Bin, Wang Zhiyuan. Combination of super voxel and color information region growing point cloud segmentation method [J]. Computer engineering and design, 2024, (5) : 1482-1489. The DOI: 10.16208 / j.i ssn1000-7024.2024.05.027.

[8] Zhang Dongdong, Guo Jie, Chen Yang. Research on point cloud object recognition based on deep learning and ensemble Learning [J]. Computer and Digital Engineering, 2019,52(03):761-767. (in Chinese)

[9] Guo Dawei, Li Jinghao, Army. 3 d point cloud registration method of multi-scale deep learning [J/OL]. Journal of intelligent systems, 1-10 [2024-07-20]. http://kns.cnki.net/kcms/detail/23.1538.TP.20240313.1451.003.html.

[10] Jiang Peige, Wu Jie, Zhang Sheirong, et al. Segmentation of point cloud based on deep learning and flood risk simulation [J]. Advances in water science, 2024, 35 (01) : 62-73. The DOI: 10.14042 / j.carol carroll nki 32.1309.2024.01.006.

[11] Shu Jun, Li Yiyang, Yang Li, et al. A multi-modal point cloud classification network based on residual MLP [J]. Journal of Chongqing University of Technology (Natural Science), 2019,38(06):242-249. (in Chinese)
[12] Liu Hui, Du Zhipeng, Yang Feng, et al. Real-time target identification method for forest and orchard spray operation based on lightweight PointNet network [J]. Transactions of the Chinese Society of Agricultural Engineering, 2018,40(08):144-151.
[13] Liu Qixin, Lu Jinzheng, Huang Bingsen. Optimization based on feature fusion and loss of semantic segmentation of point cloud network [J]. Computer technology and development, 2024, (5) : 66-72. The DOI: 10.20165 / j.carol carroll nki ISSN1673-629 - x., 2024.0042.
[14] Liang Jietao, Luo Bing, Fu Lanhui, et al. Based on coordinate geometry sampling point cloud registration method [J/OL]. Computer application, 1-11 [2024-07-22]. http://kns.cnki.net/kcms/detail/51.1307.TP.20240423.1134.002.html.
[15] Xu Jie, Liu Hui, Shen Yue, et al. Tree point cloud classification and segmentation in nursery based on improved Pointnet ++ model [J]. Chinese Journal of Lasers, 2019,51(08):193-203.