Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 25, Issues 5, pp. 3563–3571, DOI 10.12694/scpe.v25i5.3144

# THE CONSTRUCTION AND APPLICATION OF RESIDENTIAL BUILDING INFORMATION MODEL BASED ON DEEP LEARNING ALGORITHMS

SHUANG ZHAO* AND YU YANG†

**Abstract.** In order to explore the construction and application of building BIM models, the construction industry is actively exploring a method that can quickly reshape the 3D information model of existing buildings in the wave of digital twins and smart cities. Starting from the perspective of deep learning 3D object detection algorithms, the author starts with the generation of large-scale building datasets and the theory of point cloud deep learning, analyzes the input data types required for point cloud deep learning frameworks, and focuses on the creation process of 3D bounding boxes and 3D point clouds for various building components. The author compares different point cloud datasets with the same data structure and implements an object detection algorithm based on the ScanNet dataset, furthermore, a feasible technology route for automatic generation of BIM models from 3D point clouds based on deep learning is integrated. Through this technology route, the trained neural network can input unknown building 3D point clouds and output BIM model parameters.

**Key words:** Building dataset, 3D point cloud, Deep learning, BIM

**1. Introduction.** BIM (Building Information Model), also known as Building Information Model, contains the geometric information, performance, and functionality of all components in the model. It encompasses all information throughout the entire lifecycle of a building project into a single model [1]. In recent years, BIM has gradually been applied to the design and construction of modern buildings. At the same time, due to the development of new technologies, people have begun to study the parameterized information model of ancient buildings and apply it to the design and construction of some antique buildings.

With the transition of urban renewal from the "incremental era" to the "stock era", the relationship between data on built environments and corresponding human behavior data has become increasingly close. Big data demonstrates a people-oriented perspective, timely and real-time information, and fine resolution spatial dynamics. In the face of data research on built environments such as remote sensing images and street view images, after several years of image analysis research in the field of land planning, the semantic labels of high-resolution (VHR) remote sensing images assign a category task to each pixel in the image, including land use planning, infrastructure management, and urban expansion detection. The use of deep learning intervention has been widely adopted. With the deepening of deep learning technology research, street view images have gradually become an important data source for quantitative research on built environments due to their numerical characteristics and the accompanying geographical location information [2]. Spatial quantitative evaluation based on a humanistic perspective has become an important research direction, including the detection of street style features, environmental features, building materials and functions, semantic segmentation of building facade components, and the relationship between street scenery environment.

Illegal construction of structures is one of the important works in the law enforcement and supervision of natural resources. Illegal buildings not only affect urban planning and urban beauty, but also bring trouble to the management of state-owned land. All departments need to monitor and control the illegal construction information, and need a large number of people to obtain first-hand information with human and material resources. The traditional illegal construction information investigation is mostly conducted by relevant departments organizing field investigators to conduct field investigation and field comparison. If newly built or expanded buildings are found, relevant data should be transferred in the land management system in time to

*School of Architecture and Art Design, Hebei Academy of Fine Arts, Shijiazhuang, Hebei, 050800, China
†School of Architecture and Art Design, Hebei Academy of Fine Arts, Shijiazhuang, Hebei, 050800, China (Corresponding author, duoduoduo0620@163.com)

verify whether there is illegal construction. Although this method can obtain illegal construction information, it is time-consuming and laborious, and has low timeliness. With the continuous development of remote sensing technology and application, the use of remote sensing images for land resources and national conditions dynamic remote sensing monitoring has been a part of the annual law enforcement inspection in large and medium-sized cities. Law enforcement inspection based on remote sensing technology has good objectivity, but the manual identification of illegal land use and illegal construction is still the main method. In recent years, with the increasing maturity of artificial intelligence deep learning algorithms, such as using video images to carry out statistical analysis of road traffic flow, target recognition algorithm to carry out cargo ship operation, and semantic segmentation to carry out natural resources survey, etc.

Village building environment with the development of economic construction, the traditional residential buildings are increasingly suffer gradually eating, the style of the village is gradually alienation, fortunately, this problem has gradually get attention, but for the protection of traditional village development, how to evaluate the village residential architectural features, classified statistical management, is indeed a very necessary and difficult work. For the protection of traditional villages, an urgent need to the number of village residential buildings, style, building quality, building height, and other information for quantitative evaluation and analysis, and improve the village planning development management, currently in the practical village planning work, which also put forward specific requirements, but the present for village building information statistics mainly through the way of artificial field investigation, after multi-directional residential photos, artificial interpretation of its architectural characteristics. Such a way, on the one hand, is easy to be limited by the traffic, climate and terrain factors of the local villages, which brings inconvenience to data collection, but also increase a lot of research costs;On the other hand, due to the way of manual interpretation, it is bound to bring some uncertain changes due to the subjective factors such as the subject background, life experience, mood and emotion, and bring certain disturbance to the result of the definition of architectural style. The continuous emergence of new technologies and new data provides a rich data basis for more detailed spatial quality research. At the same time, applying intelligent technologies such as machine learning and edge computing to various industries is a concave solution that conforms to the development of The Times. Such a study first requires a data collection process to collect the required data that collects critical architectural imaging data, often relying on field investigations. Such a high level of labor-intensive and time-consuming work makes a large-scale assessment of architectural features extremely difficult. In this regard, the collection and integration of architectural landscape data in an effective way remains a challenge for current academic research.

Digital twin cities can make forward-looking predictions about people and things in the city to a certain extent, with the premise of reshaping high-precision and multi coupled Building Information Modeling (BIM). However, in the initial digital modeling process of BIM, very few existing buildings have complete and accurate information data. In the field of BIM model generation for existing buildings, current methods focus on preprocessing the 3D point cloud data of existing buildings, extracting point cloud features through prior knowledge and corresponding algorithms (such as Hough transform, RANSAC (RandomSampleConsensus), and then generating BIM models through certain human-computer interaction processing [3,4]. This involves a large amount of manual work, which is time-consuming and subjective and prone to errors, in addition, the generalization performance of 3D reconstruction methods based on prior knowledge is poor, and the point cloud features extracted by traditional algorithms cannot be optimized end-to-end to obtain the global optimal solution. In the process of generating BIM models from 3D point clouds, the segmentation and localization of point cloud data are the most critical steps, and automatic implementation of this step is often difficult. Going deep into the basics, the difficulty lies in the fact that BIM models are composed of various types of components, and point clouds are a whole. In the process of automatically creating BIM models, component identification, localization, and modeling need to be synchronized. With the development of deep learning theory, deep neural networks have been widely adopted in different industries and have achieved good performance in computer vision tasks. The author explores the combination of deep learning and the generation of existing building BIM models. Combining the input data required for training point cloud neural networks, a set of point cloud datasets (SYNBIM) and corresponding creation methods are proposed for conventional building components that can be applied to 3D object detection tasks, by comparing the data storage formats of the SYNBIM dataset and the Scan Net point cloud dataset proposed by the author, the input and output data of the point cloud

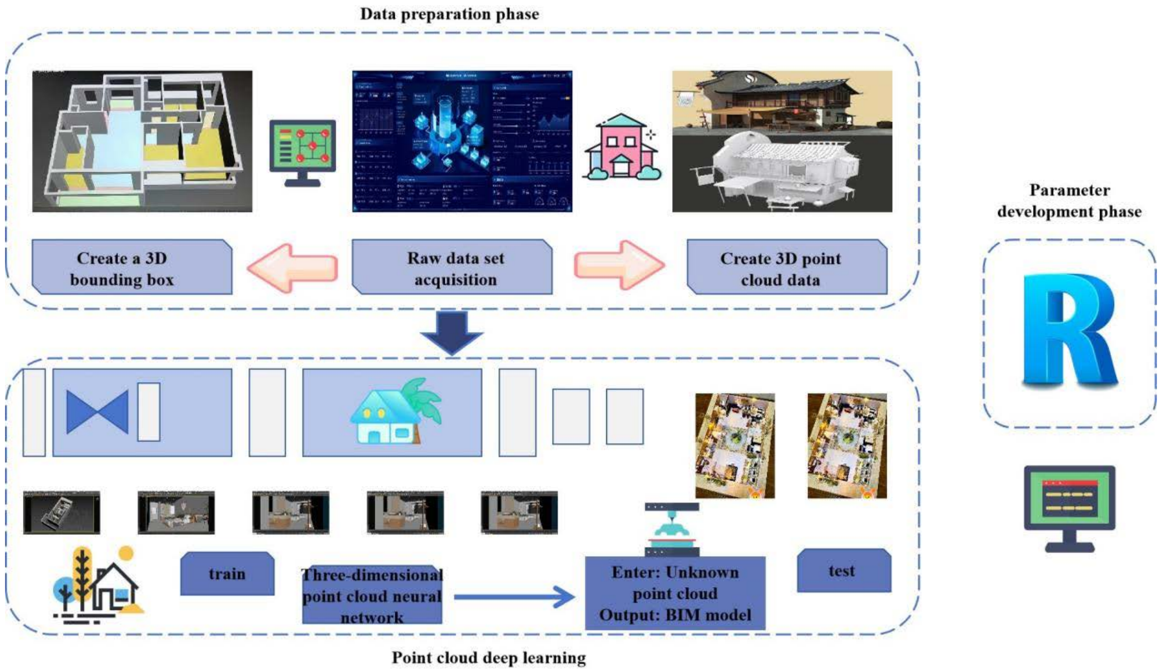


Fig. 2.1: Technical route of BIM model for automatic generation of 3 D point cloud based on deep learning

neural network framework during the testing phase were analyzed, and it was proved that the trained neural network can achieve "input unknown point clouds and output BIM model parameters" [5].

**2. Methods.**

**2.1. Technical roadmap.** The artificial intelligence algorithm is the most ideal method for reshaping BIM models in 3D and point clouds, but currently, the theory of point cloud processing and training data resources are not yet mature. The technical roadmap proposed by the author based on these two points is shown in Figure 2.1. In the data preparation stage, the SUNCG dataset 9, released by the Computer Vision and Robotics Research Group at Princeton University in 2017, was selected as the original dataset. This is a manually synthesized large-scale 3D scene virtual dataset with dense volume annotations. However, the original dataset is not mature enough for 3D bounding box annotations of conventional components in the field of civil engineering, such as walls, doors, windows, etc. In order to achieve the task of deep learning 3D object detection, the author selected five types of building components, namely "walls", "doors", "windows", "floors", and "ceilings", as the research objects. With the help of the Open3D 3D semantic library, each type of component was processed separately and corresponding hypotheses were proposed during the processing process. Finally, instance level 3D bounding boxes were generated as the labeled information for supervised learning. Then, combined with the camera pose estimation method, multiple indoor observation points are generated and stereo cameras are placed to capture multiple depth maps and two-dimensional images. These images with depth information are then synthesized into point cloud data, which can be used as input to the neural network. In terms of point cloud deep learning, by combining the input and output data of the point cloud neural network framework Votenet, and analogizing the data storage structures of the SYNBIM dataset and ScanNet dataset, a trained Votenet model is used to test the ScanNet point cloud scene, and the 3D object detection result is the BIM model [6]. For the five types of conventional building components, the data output by neural networks is well-established, and the parameter information of the components can be obtained through relevant geometric operations. In theory, corresponding BIM models can be created through Revit secondary development programming.

**2.2. Obtaining the original dataset.** The original dataset provides 45622 sets of 3D virtual scenes containing multi story buildings. Each 3D scene is saved in a house.json file. Based on the content of the JSON file, select a hierarchical processing method to generate the target 3D bounding boxes for each scene, and use separate processing for five different objects. With the help of Open3D, an open-source library that supports 3D data processing, and its advantage of customizable operation functions, the original dataset can be read and operated on.

**2.3. Deep learning principles.**

*(1) The fully convolutional neural network.* Full convolutional neural network (fully convolutional networks, FCN) uses convolutional neural network to realize the transformation from image pixel to pixel category. It can realize the classification of pixel level and solve the problem of semantic level image segmentation. Therefore, it is more suitable for the extraction of building map spots.The traditional convolutional neural network (convolutionalneural net-works, CNN) usually squthe original two-dimensional matrix into one-dimensional at the last fully connected layer, losing spatial information, and finally trains to output a label. The building change extraction task is not only to know the types of objects contained in the image, but also to divide the location of different objects.Compared with other CNN, the difference is to remove the original CNN last fully connected layer, using deconvolution layer of the last convolution layer feature map sampling, make it back to the input image of the same size, the last pixel calculation classification loss, so can produce a prediction for each pixel, retain the spatial information in the original input image. Since high-resolution remote sensing images have a lot of detailed information, the underlying low-resolution semantic features directly classify the images, which will produce obvious errors. Therefore, FCN upsamples the results of different pooling layers, forms the optimization output, and then combines them with the downsampling to obtain the final classification results.

*(2) Add deep learning features.* In the sample training, the deep learning algorithm usually carries out a lot of iterative calculation based on the texture and spectral information of the sample, but there will be false identification for the structures with different shapes and different uses. On the basis of the identification and extraction of building information by deep learning algorithm, plus the features of NDVI and SAV, it can effectively improve the differentiation degree of woodland, bare land, vegetation and building, and then improve the extraction accuracy of building changes. Both indices are calculated as follows:

$$N = \frac{B_{NIR} - B_R}{B_{NIR} + B_R} \tag{2.1}$$

$$S = \frac{B_{NIR} - B_R}{B_{NIR} + B_R + L}(1 + L) \tag{2.2}$$

where N indicates the vegetation index NDVI;BNIR represents the near-infrared band band;BR red light band; S is the regulated soil brightness index SAVI; L is the soil regulation coefficient, the value range is 0∼1, L=0 means the vegetation coverage is 0, L=1 indicates the soil background effect is 0. The texture features, spectral features and the above two exponential features of the image are combined into multi-source index, and the optimal training accuracy is obtained by the continuous iterative calculation of the image model within the sample range.

CNN can be applied in scene classification and image classification. LeNet is one of the earliest CNN structures, which is mainly used in the character classification problem. Since the convolution operation is used in the program, not only the features of the picture can be extracted, but also the convolution operation maintains the spatial relationship between the pixels. In the CNN, a filter is used as a feature extractor, and the matrix obtained by convolution is called a "feature graph". When selecting a specific CNN, the image characteristics of the target object, such as the difference between rural and urban buildings, and the situation of coarse-grained buildings.Because the real world classification problems are non-linear, and convolution operation is linear operation, so when using CNN to solve, must use a nonlinear function such as ReLU (or other nonlinear functions, such as Tanh and Sigmoid,) to add the results of the nonlinear properties, then adopt the form of downsampling, extract the features after the ReLU processing, or extract the element average or extract the maximum value, so as to reduce the dimension of the feature graph while maintaining the important information

Table 3.1: Part of Wall Ground Truth Data Styles

| xc | yc | zc | x_size | y_ size | z_size | yaw |
|---|---|---|---|---|---|---|
| $7.46\times10^{-2}$ | $2.69\times10^{0}$ | 1.4 | 5.26 | $7.86\times10^{-2}$ | 2.73 | $1.58/\times10^{0}$ |
| $9.27\times10^{0}$ | $1.38\times10^{1}$ | 1.39 | 5.04 | $1.03\times10^{-1}$ | 2.73 | $1.58/\times10^{0}$ |
| $1.51\times10^{1}$ | $1.54\times10^{1}$ | 1.38 | 2.15 | $1.03\times10^{-1}$ | 2.78 | $1.58/\times10^{0}$ |
| $4.68\times10^{0}$ | $1.27\times10^{1}$ | 1.35 | 2.71 | $9.36\times10^{-2}$ | 2.73 | $1.58/\times10^{0}$ |
| $1.73\times10^{1}$ | $1.13\times10^{1}$ | 1.35 | 4.06 | $1.06\times10^{-1}$ | 2.7 | $1.75/\times10^{-3}$ |
| $9.3\times10^{0}$ | $2.68\times10^{0}$ | 1.37 | 4.9 | $1.04\times10^{-1}$ | 2.73 | $1.58/\times10^{0}$ |
| $1.38\times10^{0}$ | $1.63\times10^{1}$ | 1.39 | 2.41 | $9.56\times10^{-2}$ | 2.81 | $3.45/\times10^{-3}$ |
| $3.62\times10^{0}$ | $5.27\times10^{0}$ | 1.37 | 6.61 | $9.86\times10^{-2}$ | 2.71 | $3.15/\times10^{0}$ |
| ... | ... | ... | ... | ... | ... | ... |

of the picture. Finally, the fully connected layer (multilayer perceptron) is combined together and classified through the whole layer using a softmax activation function. To is a vector with a value of 0-1 to judge picture classification by the probability value.

In recent years, deep learning methods, especially CNN performance in various computer vision tasks have gone beyond traditional methods, such as its excellent research results in target detection, semantics and image segmentation. The method of labeling image pixels with labels is based on the semantics in the image, that is, the algorithm will exist in the image, such as cars, trees or buildings as semantics from the extraction of the whole image, and each semantics. Moreover, in the field of computer vision, there is a large amount of research on the various modules used in convolutional neural networks that utilize the concept of ”per object classification”. These modules, such as convolution and pyramid pooling, improve the algorithmic performance for semantic segmentation tasks. In recent years, with the significant improvement of chip processing power (such as GPU units), the significant reduction of the cost of computing hardware, and the significant progress of machine learning algorithms, deep learning has made rapid progress in the field of image recognition, thus greatly improving the processing power of computers.

## 3. Results and Analysis.

**3.1. Creating 3D bounding box walls.** The manifestation of walls in the field of multi story and multi room indoor scene research is very complex. In order to prevent the calibration requirements of model bounding boxes in 3D object detection, a set of paradigms is needed to define the representation of wall objects, therefore, the author proposes the following assumptions when defining wall objects: (1) Shortest wall assumption: When a long wall and a short wall intersect in the same floor space, the long wall is split into two walls at the intersection; (2) One wall corresponds uniquely to a 3D bounding box. The bounding boxes extracted directly from the geometric information of the walls in the original data through programming, where different walls are marked with different colored bounding boxes. However, based on the above assumptions, there are several issues: Firstly, multiple long walls are not broken when encountering short walls; Secondly, there are multiple instances where the same wall is repeatedly calibrated by multiple bounding boxes, resulting in severe box overlap; Finally, there is no height difference between different bounding boxes, which may hide walls with repeated calibration.

By calling the open3d library functions through programming and customizing relevant functions to read and operate on the raw data, a set of bounding boxes is generated. The single-layer building walls are processed to meet the above assumptions.

The data of the wall bounding box is shown in Table 3.1, where a single row of data represents information about a bounding box, and seven parameters represent the coordinates of the three center points of the box in the overall coordinate system, the length, width, height of the box, and the deflection angle along the height direction. These generate the calibration parameters for the wall, and the other four components are processed and have the same data structure as the wall bounding box [7].

**3.2. Creating 3D bounding boxes - doors and windows.** During the process of processing raw data, doors and windows encountered the problems shown in Table 3.2 and Figure 3.1. Through investigation, it

Table 3.2: Initial Boundary Box Problems and Solutions for Doors and Windows

| Initial problem | Initial error rate/% | Solution | Final accuracy rate/% |
|---|---|---|---|
| Thickness greater than wall | 23.48 | Detect the wall where it is located | 99.18 |
| Detaching from the overall building | 9.7 | Recalibrate the detection&delete it | 99.8 |
| Repeated calibration | 19.87 | Merge | 98.18 |

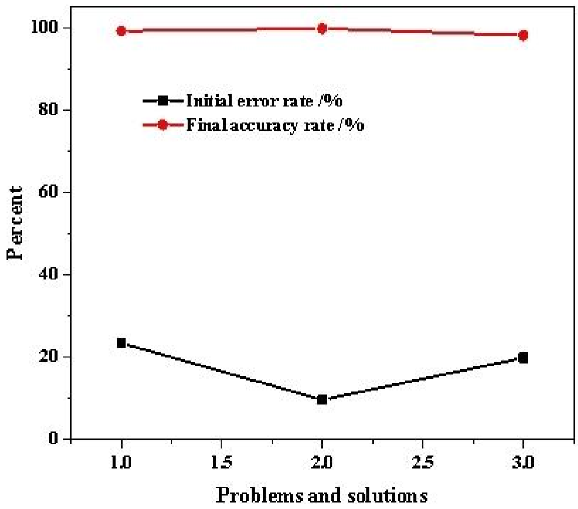


Fig. 3.1: Error rate and final accuracy rate of initial bounding boxes for doors and windows

was found that some buildings had small doors and partially open windows [8]. According to the minimum bounding box theory, the thickness of the bounding box obtained by directly extracting data from partially open doors and windows exceeded the thickness of the wall.

**3.3. Creating 3D bounding boxes - Floors and ceilings.** The author defines the following rules when creating board class bounding boxes in the program: (1) Board objects cannot cover two or more rooms; (2) For multi story buildings, the boundary boxes of two floors or two ceilings cannot overlap, but overlapping is allowed between the two types of boxes of floors and ceilings; (3) The edges of at least three edges of the floor and ceiling bounding boxes are in contact with the edges of the wall.

**3.4. Creating 3D bounding boxes - entire building .** For single story buildings, create 3D bounding boxes for each target instance using the above methods to provide marker information for deep learning 3D object detection training tasks; For multi story buildings, the author first separates the information of each layer separately during the programming process of reading and manipulating the raw data, then sequentially processes and obtains the 3D bounding boxes of each layer's target instances. Finally, the processed multi story data is combined together to complete the processed multi story 3D bounding boxes [9].

**3.5. Building Datasets - Creating 3D Point Clouds.** In current practical applications, the digital acquisition based on 3D laser point clouds, which uses laser scanners to scan indoor and outdoor buildings in 3D, is a common method to obtain building point cloud data [10]. However, LiDAR has problems such as high cost, mirror black holes, and low lifespan, the method of generating depth images through stereo vision cameras and synthesizing point clouds is the future development direction. Depth images, also known as distance images, refer to images that use the distance values from the image collector to each point in the scene as pixel values. he author takes a long-term perspective and chooses to place a stereo vision camera at the virtual observation

Table 3.3: Dataset generation process and partial result presentation

| Classification | subclass | 3D Scene #1 Layer | 3D Scene #2 Layer | 3D Scene #3 Layer |
|---|---|---|---|---|
| Input | #Number of Layers/# Number of Depth Maps/#Number of Point Clouds/ | 1/1596/ 3041.407 | 2/768/ 1472.243 | 3 /549 / 2685.806 |
| Number | Number of rooms/area/m2 | 10/180.32 | 6/142.38 | 21 /278.52 |
| | Boundary Box # Wall/# Door/# Window | 28/9/3 | 11+9 /2+1 /1+4 | 23+23+26 /4+4+4 /4+4+3 |
| | Boundary Box # Floor/# Ceiling | 10/9 | 3+2/2+2 | 6+6+6 /3+3+3 |
| | Downsampling/Original Point Cloud/k | 29987/ 16849.562 | 18785/ 12492.263 | 36429/ 19443.005 |
| Running time/s | All bounding boxes | 12.45 | 10.6 | 17.9 |
| | Depth image+ color rendering | 654.76+ 99.46 | 445.67+ 67.42 | 953.16+ 137.67 |
| | 3D point cloud | 1843.13 | 1403.78 | 1930.19 |

points contained inside each building in the SUNCG dataset and generate corresponding depth images. The obtained depth images are 16 bit PNG format files, which can be used to generate 3D point clouds through relevant registration and registration algorithms.

Stereoscopic vision cameras need to capture objects from different observation points, and the depth map information obtained from different stations has their own independent coordinate systems [11]. After unifying the coordinate system, data fusion needs to be carried out. The basic formulas involved are:

$$\begin{cases} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + R_z(\alpha)R_z(\beta)R_z(\gamma) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ R_z(\alpha)R_z(\beta)R_z(\gamma) = \begin{bmatrix} cos\alpha, -sin\alpha, 0 \\ sin\alpha, cos\alpha, 0 \\ 1, 1, 1 \end{bmatrix} \\ \begin{bmatrix} cos\beta, 0, -sin\beta \\ 0, 1, 0 \\ -sin\beta, 0, cos\beta \end{bmatrix} \begin{bmatrix} 1, 0, 0 \\ 0, cos\gamma, sin\gamma \\ 0, -sin\gamma, cos\gamma \end{bmatrix} \end{cases} \tag{3.1}$$

In the formula: (X, Y, Z) represents the coordinates after point cloud registration; (x, y, z) are the original coordinates of the point cloud; $\Delta x, \Delta y, \Delta z$ is the translation parameter; $\alpha, \beta, \gamma$ is the rotation parameter[12]. By creating bounding box marker information and point cloud data, the dataset required for training the point cloud neural network was obtained, consisting of 45622 sets of large-scale 3D scene data. The hardware information and related calculation time used in the data generation process are shown in Table 3.3 [13].

**3.6. Point Cloud Neural Network.** At present, the application of deep learning to input 3D point cloud data into neural networks is still an open problem, and there are mainly three methods used: (1) Projecting point cloud data onto a 2D plane, which does not directly process 3D point cloud data, but first projects the point cloud to certain specific perspectives before processing, such as MUCNNL; (2) Dividing point cloud data into voxels with spatial dependencies, this approach involves segmenting three-dimensional space, introducing spatial dependencies into point cloud data, and then using methods such as 3D convolution for processing. However, the computational complexity of 3D convolution is relatively high, such as PointCNN; (3) Directly processing the original point cloud, this method generally involves a multi-step 3D object detection algorithm.

Table 3.4: Comparison of Dataset Properties

| Data set | Scale | Data sources | Object scope | Data in | Output data | Using Lingcheng |
|---|---|---|---|---|---|---|
| ScanNet | 1513 | Real scan | Indoor scenes | 3D point cloud | 3D bounding box | Computer Vision |
| Hours (recalibration) | 45622 | synthetic | Indoor scenes | 3D point cloud | 3D bounding box | Architectural reshaping |

Table 3.5: Geometric parameters of target components

| Component category | Geometrical parameter |
|---|---|
| Wall | Wall bottom elevation, wall top elevation, wall centerline starting and ending wall thickness |
| Doors and windows | Placement coordinates, door bottom elevation, and door top elevation |
| Floors and ceilings | Outline boundary and plate thickness |

In order to deal with the perspective invariance of each target in the point cloud data and obtain accurate 3D box regression, the algorithm needs to perform multiple coordinate transformations, such as Votenet [14]. The reason for the above three input methods is that: (1) Point cloud data is a collection that is not sensitive to the order of data, and the model processing point cloud data needs to maintain invariance to different arrangements of data; (2) The target represented by point cloud data should have invariance to certain rigid transformations, such as rotation and translation. This network does not rely on any RGB information and only relies on simple point cloud geometric information [15,16]. The entire network framework is divided into two parts: The first part uses a point cloud feature extraction network to process the original point cloud data and generate seed points and growth points; The second part aims to achieve classification and localization by training the extracted growth point clusters. It can directly input point cloud data and output target 3D bounding boxes in 3D object detection tasks, achieving good results in the benchmark challenge of the real scanning dataset ScanNet. The comparison of some key parameters between the ScanNet dataset and the SYNBIM dataset created by the author is shown in Table 3.4 [17].

**3.7. Parametric modeling.** From the perspective of data coherence, parameterized modeling of BIM components only requires extracting model parameters to achieve model creation. Therefore, in response to the author's selection of component categories, the required parameters for creating a BIM model are summarized and listed in Table 3.5 [18].

For these five types of building components, the data output by neural networks is well-established, and the parameter information of the components is obtained through corresponding algorithm operations. Then, through Revit secondary development, corresponding BIM models can be created [19,20].

**4. Conclusion.** The author explores the combination of deep learning and existing building BIM model generation based on 3D object detection technology, and integrates a feasible technology roadmap for automatic generation of BIM models from 3D point clouds based on deep learning. The following important issues are discussed: Generate target category labeling information from virtual BIM models; Generate a 3D point cloud model from a virtual BIM model; The experimental analysis of the data input and output of the point cloud neural network proves that its output data foundation is complete, and a BIM model can be generated through simple parameter modeling. The implementation method of a deep learning based 3D point cloud generation BIM model proposed by the author unifies the steps of point cloud data preprocessing, component recognition, segmentation, localization, and modeling in traditional methods. Through the deep learning backpropagation mechanism, global optimization can be achieved for the entire process. The core process is the generation of 3D point cloud datasets and the implementation of 3D object detection algorithms.

REFERENCES


[1] Hsieh, M. C., Huang, G. H., Dmitriev, A., & Lin, C. H. (2022). Deep learning application for classification of ionospheric height profiles measured by radio occultation technique. Remote. Sens., 14(123), 4521.

[2] Alam, M., Zhao, E. J., Lam, C. K., & Rubin, D. L. (2023). Segmentation-assisted fully convolutional neural network enhances deep learning performance to identify proliferative diabetic retinopathy. Journal of clinical medicine, 12(1),79.

[3] Wang, H., Li, C., Zhang, Z., Kershaw, S., Holmer, L. E., & Zhang, Y., et al. (2022). Fossil brachiopod identification using a new deep convolutional neural network. Gondwana research: international geoscience journal88(105-), 105.

[4] B, H. W. A., A, W. G., A, H. N., B, J. S. A., & A, M. Z. (2022). Classification of giemsa staining chromosome using input-aware deep convolutional neural network with integrated uncertainty estimates. Biomedical Signal Processing and Control, 71(1123), 103120-.

[5] Sang, X., Zhou, R. G., Li, Y., & Xiong, S. (2022). One-dimensional deep convolutional neural network for mineral classification from raman spectroscopy. Neural processing letters457(1), 54.

[6] Gao, F., Li, F., Fu, Y., You, S., Zhang, S., & Cao, K., et al. (2022). Marine aquaculture mapping using gf-1 wfv satellite images and full resolution cascade convolutional neural network. International Journal of Digital Earth, 15(1), 2047-2060.

[7] Joshi, A., Lachure, J.,& Doriya, R. (2023). Drone security for precision agriculture by using one-dimensional convolutional neural network. Journal of Uncertain Systems, 16(04),24.

[8] Vinolin, V., & Sucharitha, M. (2022). Taylor-rider-based deep convolutional neural network for image forgery detection in 3d lighting environment. Data technologies and applications687(1), 56.

[9] Kyeong-Beom, P., & Yeol, L. J. (2022). Swine-net: hybrid deep learning approach to novel polyp segmentation using convolutional neural network and swin transformer. Journal of Computational Design and Engineering56655(2), 2.

[10] Aljohani, N. R., Fayoumi, A., & Hassan, S. U. (2023). A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations:. Journal of Information Science, 49(1), 79-92.

[11] Shin, H. (2022). Deep convolutional neural network-based signal quality assessment for photoplethysmogram. Computers in Biology and Medicine, 145(123), 105430-.

[12] Radman, A., Mahdianpari, M., Brisco, B., Salehi, B., & Mohammadimanesh, F. (2023). Dual-branch fusion of convolutional neural network and graph convolutional network for polsar image classification. Remote Sensing, 15(1), 75-.

[13] Fan, Y., Xu, B., Zhang, L., Song, J., Zomaya, A., & Li, K. C. (2023). Validating the integrity of convolutional neural network predictions based on zero-knowledge proof. Inf. Sci., 625(345), 125-140.

[14] Puneet, Kumar, R., & Gupta, M. (2022). Optical coherence tomography image based eye disease detection using deep convolutional neural network. Health information science and systems, 10(1), 13.

[15] Ho, C. C., & Yu, C. W. (2022). Deep convolutional neural network based fabric color difference detection. 2022 25th International Conference on Mechatronics Technology 547(ICMT), 1-5.

[16] Liu, X., Zhou, H., Wang, Z., Liu, X., Li, X., & Nie, C., et al. (2022). Fully convolutional neural network deep learning model fully in patients with type 2 diabetes complicated with peripheral neuropathy by high-frequency ultrasound image. Computational and mathematical methods in medicine, 2022(3325), 5466173.

[17] Yang, X. B., & Zhang, W. (2022). Heterogeneous face detection based on multi-task cascaded convolutional neural network. IET image processing8967(1), 16.

[18] Korkmaz, D., & Acikgoz, H. (2022). An efficient fault classification method in solar photovoltaic modules using transfer learning and multi-scale convolutional neural network. Eng. Appl. Artif. Intell., 113(13), 104959.

[19] Liu, L., Prodanovi, M., & Pyrcz, M. J. (2023). Impact of geostatistical nonstationarity on convolutional neural network predictions. Computational Geosciences, 27(1), 35-44.

[20] Yang, E., Kim, C. K., Guan, Y., Koo, B. B., & Kim, J. H. (2022). 3d multi-scale residual fully convolutional neural network for segmentation of extremely large-sized kidney tumor. Computer Methods and Programs in Biomedicine, 215(4), 106616-.


*Edited by:* Zhigao Zheng
*Special issue on:* Graph Powered Big Aerospace Data Processing
*Received:* Jan 5, 2024
*Accepted:* Feb 26, 2024