

Deep Belief Network Based 3D Models Classification in Building Information Modeling

<http://dx.doi.org/10.3991/ijoe.v11i5.4953>

WANG Li, ZHAO Zhi-kai*, XU Na
China University of Mining and Technology, Xuzhou, China

Abstract—3D model classification is a critical process of building information modeling (BIM). A deep learning approach was proposed to classify 3D models in BIM environments. A ray-based feature extraction algorithm was used to extract the features of 3D models and form a features matrix. The deep belief network (DBN) constructed by restricted Boltzmann machines applies a features matrix and classifies the models, adopting the effective training process. The process of training DBN is layer by layer. Experiments were taken from a public 3D model library of a PSB model database. The results show that compared with several commonly used classification methods, the method proposed in this paper achieved good results in the 3D model classification for efficient BIM.

Index Terms—Deep learning, BIM, 3D model classification, Deep Belief Network, Feature extraction

I. INTRODUCTION

There have always been problems with information loss and unclear expression in the traditional construction industry, and the degree of spreading information is relatively low. Considering the wide application of computer technology in other fields, the traditional construction industry urgently needed renovation, and building information modeling (BIM) has emerged [1]. The concept of BIM refers to product model definitions in the manufacturing industry. An information model contains the product's composition and function and the behavior of the data, which describes the product in its entire life cycle. BIM is such a technology, method, and system. Through the integration of information collection, management, exchange, storage, update processes, and project business processes a BIM provides timely, accurate, and sufficient information for the construction of a project's life cycle at different stages and for different parties. It also supports information exchange and sharing between the various stages of the project, between the parties involved in projects, and between different application software. Thus, it achieves a project's design, construction, operation, maintenance, and improvement as well as supports the construction industry's productivity and ability to upgrade.

From the application field, foreign countries have already applied BIM technology to the design, construction, and maintenance and management stages. The corresponding application software has been gradually maturing. In the past ten years, the application of BIM has boomed in China [2]. But due to the impact of the hardware and software environment, BIM technology has

mainly stayed in the architectural design phase of a major design institute.

The 3D geometric model is the main data expression of modes in the BIM environment, but because of the complexity of 3D modeling, the maintenance of the 3D model library in the BIM environment will require a lot of time and costs. With the rapid development and extensive application of CAD technology, the 3D CAD models have grown in number and variety [3]. These massive 3D CAD models provide rich resources for the modeling of 3D geometry in BIM environments. The reuse of existing design models can improve the products' design quality and design efficiency, while reducing the diversity and repeatability of production and organization to reduce production costs. However, due to the lack of an effective auxiliary tool, design engineering staff can only rely on manual text retrieval to find appropriate reuses. This often requires spending 60% or more of the time on product design and eventually has led to the abandonment of reusing existing design ideas and choosing to start a new design [4]. This setback in search technology has led to a large number of existing design models and knowledge that cannot be used effectively, causing a great loss in design time and energy and a waste of enterprise economy. Therefore, how to study effective classification methods of 3D geometry models in a BIM environment to help project designers quickly find suitable options for reuse has become urgent.

Content-based 3D model retrieval techniques attempt to direct extraction of the feature information of a 3D model, establish an index, and then compare similar query results to help the users quickly find the needed model. Compared with text retrieval, retrieval technology with high efficiency and reliable characterization methods, the stability of the retrieval results has received more and more attention. As an important step of 3D model retrieval accuracy, 3D model classification algorithms play a key role. Some 3D model classification algorithms have achieved good results. In recent years deep learning has become a new method of machine learning, which has very good application in the areas of images and text because it conforms to cognitive rules. This paper proposes a classification algorithm of a 3D model based on a deep belief network and provides a new idea to improve the efficiency of 3D modeling under BIM environments.

II. RELATED WORK

A. Building Information Modeling (BIM)

In the past ten years, BIM technology has experienced rapid development. Internationally, IAI has released BIM

data standards IFC, has developed a standard format for BIM data exchange; and BuildingSMART has released the IDM standards used to describe and standardize the process of BIM exchanges [5]. The USA has issued national BIM standards (NBIMS) to guide the practical application of BIM [6]. Domestically, during the "ten five" and "eleven five" period, multiple national science and technology support programs have researched BIM theory, methods, tools and standards. Research and application of construction in each stage of BIM has increased [7]. Xue et al. [8] analyzed BIM in visual design, parametric design, application of collaborative design, and working mechanisms and illustrated the application of parametric design for building structures. Li et al. [9] selected Revit and Project software to create 3D building information modeling, and schedule information construction was analyzed. as was the integration of existing Project, P3 and other software for schedule management advantages to build a 4D schedule management system based on the Revit platform. Wang et al. [10] constructed a schedule risk analysis model based on the BIM platform project using the IFC parser in BIM tools derived from a construction project IFC data file. Case-based reasoning is given under the guidance of the combination of manual adjustments and supplements to complete progress plan generation of construction projects and construction of the overall architecture of a BIM construction schedule risk based on project analysis to put forward the environment and countermeasures of an application model and its application. Lou et al. [11] constructed the cost budget software requirement for an information model based on a BIM management platform and an architectural engineering budget based on an IFC standard information model. The platform model is divided into two parts of a BIM data management kernel and platform user interface and corresponds to four function layers: the data exchange layer, data storage layer, data application layer, and user interface layer, respectively. Hu et al. [12] built a 4D spatio-temporal model of a construction site and facilities entity and provided the construction site physical collision detection algorithm. The model can detect and analyze physical collisions between the real-time dynamic of the facilities and the main time-varying structure to make a reasonable planning and real-time adjustment on the construction site. Fan et al.'s [13] comprehensive application of 4DCAD technology and BIM technology constructed the 4D resource information model of the construction phase, developed the 4D resource dynamic management system based on BIM technology, and realized the dynamic query of the engineering in the construction process as well as the dynamic management of people, materials, and machine construction resources, and monitored real time construction costs and payment management of the project's progress. Jiang et al. [14] proposed a BIM based construction project document management system, analyzed the necessity and feasibility of the implementation of the system, and constructed the system framework. Zhang et al. [15] presented the integration of the BIM framework for building a life cycle through the research on the integration of the BIM basic structure, process modeling, application framework, and key technologies of modeling. Zhang et al. developed a prototype system of BIM data integration and a service platform (BIMDISP) and verified the feasibility of the BIMDISP by a practical application in engineering.

B. 3D model classification

3D model classification is a key step in 3D model retrieval to improve the retrieval efficiency. Some scholars have used machine learning methods to research 3D model classification. So far, in the engineering field, CAD model classification research has achieved certain results. Ip et al. [16] used surface curvature and SVM to classify CAD models. Hou et al. [17] proposed a semi-supervised clustering method based on SVM to semantically organize 3D models. Each of the input data is a mixture of three feature vector representations: moment invariants, geometric ratio, and principal component, from different angles to represent the 3D model. In addition, for the general 3D shape classification, the researchers also put forward some corresponding methods. Barutcuoglu et al. [18] proposed a Bayesian framework for hierarchical classification of 3D shapes. Given a set of classifiers corresponding to the arbitrary shape descriptor, a Bayesian framework organically organizes the classification results together. The framework coordinates those inconsistent prediction results and obtains optimal predicted results. In the experiment, a spherical harmonic descriptor verified that the Bayesian framework can improve classification accuracy in most categories. Wei et al. [19] adopted the Hopfield neural network to identify the 3D model with color. The color of the surface of the 3D VRML model was used as input vectors to the Hopfield neural network. Philipp-Foliguet et al. [20] proposed a framework for indexing and retrieving 3D archaeological art. The framework includes two parts that support the search and classification of global and local models. The first part uses global shape descriptors; the search engine "RETIN-3D" uses the strategies to combine the SVM classifiers and active learning to retrieve similar categories. The second part uses the model's local descriptors to improve and enhance the classification results. Wang et al. [21] proposed a new method to automatically select the appropriate descriptors to classify and retrieve 3D models. First, some shape descriptions were calculated based on a histogram to form a feature space and then to automatically select the important descriptors based on sparse theory. Finally, spectral clustering was used to generate new shape descriptors. The method is not only applicable to complete the 3D model but also applies to the incomplete 3D model. The deficiency lies in the feature space descriptor that must be based on a histogram.

C. Deep Learning

In recent years, the deep learning method appeared in the machine learning field through learning a deep nonlinear network structure, realizing complex function approximation, characterization of the input data distributed representation, and showing the strong ability of learning essential characteristics from a small sample of the data set. By constructing multiple hidden layers of machine learning models, more useful characteristics were learned and ultimately promoted the accuracy of the classification or prediction. In 2006, Hinton et al. [22] proposed an unsupervised learning algorithm of the deep belief network to solve the optimization problem model of deep learning. The core of solving the DBN method is a greedy layer wise pre-training algorithm. Through optimizing DBN weights under linear time complexity with network size and depth, the problem is decomposed

into several simple sub problems to be solved. Bengio et al. [23] and Ranzato et al. [24] proposed the idea of initializing each layer of neural network using unsupervised learning. Erhan et al. [25] tried to understand that how unsupervised learning helps deep learning. Glorot and Bengio [26] studied the causes of the original training process's failure in a deep neural network. Bengio [27] gave a comprehensive review on the deep learning. He proposed a greedy layer wise pre-training and learning process for initializing the parameters of the deep learning model based on unsupervised learning techniques. The bottom the input representation of each layer was formed by training each layer's neural network. After the unsupervised initialization, each layer of the neural network was stacked into a deep supervised feed-forward neural network, which was fine tuned by gradient descent. The learning method of deep learning mainly focused on learning useful data representation and made sure that the learned features did not change with the factor change and were robust to sudden changes in the actual data [28]. This paper [29] identified related skills of training in the deep learning model, especially the Restricted Boltzmann Machine (RBM). Many training idea from the neural network can also be used for deep neural network learning. Bengio [30] gives guidance on the training methods for different types of deep neural networks.

III. 3D MODEL FEATURE DESCRIPTORS

A. 3D model representation

The 3D model of building is the core of a building information model, realistically representing the real building. There are three kinds of methods to represent the 3D model: wire-frame model, surface model, and solid model [31].

1) Wire-frame model

A wire-frame model is a visual presentation of a 3D or physical object used in 3D computer graphics. It is created by specifying each edge of the physical object where two mathematically continuous smooth surfaces meet or by connecting an object's constituent vertices using straight lines or curves. The object is projected onto a display screen by drawing lines at the location of each edge.

This method has the advantages of a simple structure and easy processing, but there are also shortcomings for 3D objects: 1) there is no surface information, and it cannot express surface of curved objects; 2) it does not explicitly define the relationship between point and object; and 3) probably there is ambiguity for point and edge information.

2) Surface model

Based on the wire-frame model, the surface model also contains the surface information of the objects. It represents objects using a surface set. Each surface consists of a plurality of directed edges, which are used to define the boundary of a ring. That is, it describes objects using a vertex table, an edge table, and a surface table.

The disadvantages of the surface model include that 1) it can only express the surface boundary objects but not the attributes of a real entity; and 2) it is difficult to confirm whether the 3D graphical representation of a surface model is an entity or an empty shell. This problem will be resolved in the solid model.

3) Solid model

The solid model is the most advanced model. It can completely represent all of the shape information of the object, of which a point can be unambiguously identified as an external or internal or surface.

The solid model is the priority for expressing the 3D object in a computer for Building Information Model technology.

B. Model coordinates standardization and pretreatment

To realize the retrieval and classification of a content-based 3D model, the key point is to require the extracted 3D model feature description is invariant and robust to translation, rotation, scale size and orientation transformation. Currently, there are mainly three methods to realize the feature description invariance: 1) Using principal component analysis and other methods to standardize the 3D model coordinates, calculate the standard coordinate frame, and then extract the corresponding feature under the standard coordinate system. 2) Align the coordinates of the two compared 3D models. This method is very time consuming and is rarely used. 3) Define and extract the invariant features descriptions. But in fact, the invariance of these features descriptions are often not comprehensive; moreover, most of the calculations of these features are also sometimes needed to finish in a normalized coordinate system. Therefore, to further improve the accuracy of classification and retrieval, it often also needs to standardize model coordinates.

In this project, we used existing technology to standardize the 3D model. To ensure the model was translation invariant, translational transform methods were used to align the centroid and origin coordinate of the model. To ensure the rotation invariance, PCA was used to rotate the model to the uniform rotation angle. To ensure the scale invariance, the 3D models were scaled up to a uniform scale, reflecting the transformation executed to guarantee mutual consistency of two 3D models in the mirror.

C. Feature extraction

We used ray-based feature extraction algorithm to extract features of 3D model, the basis algorithm process as follows:

1) Firstly, the model is a standardized pretreatment, moving the center to the origin of the coordinate, scaling the model to the unit size, and making all of the model's directions and rotation angles consistent.

2) Spherical parameterization for the model used the surrounding spherical ball with the center at the origin, projecting a series of uniform sampling rays from the origin to the spherical warp and weft direction, respectively, to calculate the intersection point of each ray and the model surface.

3) For each ray, using the maximum value from the model origin to the surface intersection as the sampling value of the direction, the sampled values of all directions constitute the feature vector.

The calculation method based on a ray is shown in Table 1. The inputs of function include the 3D mesh model, the ray number N along the warp, and the ray number M along the weft. The outputs are the distance of

each direction from the intersection of the ray and model surface to the model centroid.

Table 1. Ray based feature extraction algorithm

```

Ray-based
{
  for(i=0;i<N; i++)
  {
     $\phi = 2i \pi / N$  ;
    for(j=0;j<M; j++)
    {
       $\theta = 2j \pi / M$  ;
      for(k=0;k<The total number of triangles; k++)
      {
        Calculate intersection of ray with direction(  $\theta$  ,
           $\phi$  ) and triangle k;
        If intersection exist, then calculate and save
          the distance from the origin to the
          intersection point;
      }
    }
  }
}
    
```

IV. DBN BASED CLASSIFICATION

A deep belief network was proposed by Geoffrey Hinton in 2006. It is a generative model. The component of DBN is a Restricted Boltzmann Machine (RBM). The process of training DBN is layer by layer. In each layer, a data vector is used to infer the hidden layer, which is the data vector of the next layer.

A. Restricted Boltzmann Machine (RBM)

RBM consists of two layers: the visible and hidden layers. The connections between neurons have the following characteristics: no connection between layers and fully connected within the layer. Therefore, the corresponding map of an RBM is a bipartite graph. The diagram of RBM network structure is shown in Fig. 1.

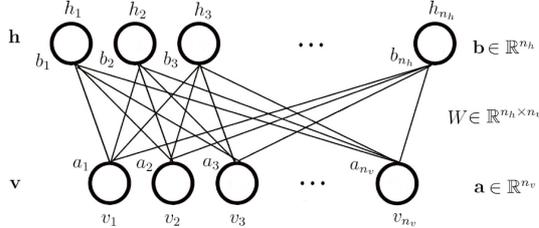


Figure 1. The diagram of RBM network structure

n_v, n_h : Denote the numbers of neurons contained in visible layer and hidden layer.

$\mathbf{v} = (v_1, v_2, \dots, v_{n_v})^T$: The state vector of visible layers, v_i denotes the state of the i th neuron in visible layer.

$\mathbf{h} = (h_1, h_2, \dots, h_{n_h})^T$: The state vector of hidden layers, h_j denotes the state of the j th neuron in hidden layer.

$\mathbf{a} = (a_1, a_2, \dots, a_{n_v})^T \in \mathbb{R}^{n_v}$: The bias vector of visible layer, a_i denotes the bias of the i th neuron in visible layer.

$\mathbf{b} = (b_1, b_2, \dots, b_{n_h})^T \in \mathbb{R}^{n_h}$: The state vector of hidden layers, b_j denotes the bias of the j th neuron in hidden layer.

$\mathbf{W} = (w_{i,j}) \in \mathbb{R}^{n_h \times n_v}$: The weight matrix between hidden layer and visible layer, $w_{i,j}$ denotes the connection weight

between the i th neuron in hidden layer and j th neuron in visible layer.

B. DBN model

The effective training process of an RBM makes it suitable as a module of DBN. The hidden units of each layer of an RBM learn to express higher degree characteristics than the original input data. The DBN can be considered as a neural network with well trained initial weights. When the DBN structure does not use the class label and backward propagation, it can be used for dimensionality reduction. When there are class labels of feature vectors, DBN can be used for classification. The following are two models of DBN classifiers: backward propagation DBN and associative memory DBN. They all use a greedy layer-wise contrastive divergence criterion for weight initialization.

1) Wire-frame model

A last layer, which represents the desired output variables, was added to a multilayer RBM. It uses the output of the last layer of the RBM network as the input of a BP neural network. Then the supervised BP neural network was used to do error-back propagation to fine tune the whole model top-down. After the multi-layer RBM network optimization, the information was used as the input of a BP neural network, which can solve the problem that falls into a local minimum and slows the convergence speed, etc.; the network structure is shown in Fig. 2. This model uses a greedy layer-wise contrastive divergence criterion for weight initialization.

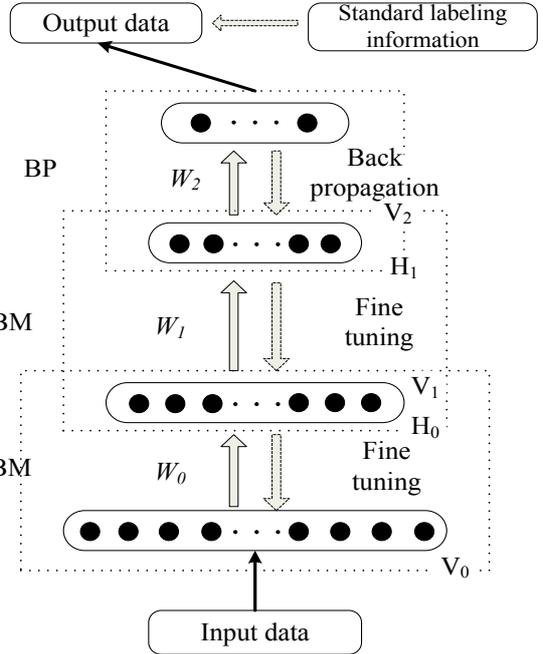


Figure 2. Structure of a DBN

2) associative memory DBN

The top RBM of the DBN can learn the abstract features of the experimental data. However, these abstract features still need to be associated with particular labels for the classification task: the top RBM model, the joint probability distribution for input vectors, and the associated labels. When there are multiple categories, the frequency distributions of different classes are far from consistent.

Using different code for the target categories has more advantages over the standard 1-k softmax code. It is quite direct to use an arbitrary fixed two value coding for each class. Assuming that q dimensional code vector was used to express each class, the model will get a q output unit. Let y be the q dimensional column vector that is generated by a network with corresponding input. Then y needs to be transformed into a categories forward probability vector.

C. DBN based 3D model classification

The characteristics of input data can be obtained by a ray-based feature extraction algorithm, and DBN has good classification efficiency. Therefore, we proposed a 3D model recognition and classification method based on DBN algorithm. The main process of the algorithm is described as follows.

Table 2. DBN based 3D model classification algorithm

Step 1 Choose appropriate training and testing set.
Step 2 Process 3D model feature description, and represent the data set using matrix.
Step 3 Initialization of parameter DBN (learning rate, number of cycles, the hidden layer unit number etc.).
Step 4 Input the training set to DBN; modify the parameters; test the classification effect using testing set.

Table 3. The training algorithm of RBM

Step 1 Initialization
(1) Given training sample set S ($ S =n_s$).
(2) Given the training cycle J , the learning rate η and the parameter k of contrastive divergence algorithm CDK .
(3) Specify the number of visible layer and the hidden layer unit n_v, n_h .
(4) Initialize bias vector \mathbf{a}, \mathbf{b} and the weight matrix \mathbf{W} .
Step 2 Training
FOR $iter = 1, 2, \dots, J$ DO
{
1. Call contrastive divergence algorithm $CDK(k, S, RBM(\mathbf{W}, \mathbf{a}, \mathbf{b}); \Delta\mathbf{W}, \Delta\mathbf{a}, \Delta\mathbf{b})$, and generate $\Delta\mathbf{W}, \Delta\mathbf{a}, \Delta\mathbf{b}$.
2. Refresh parameters: $\mathbf{W} = \mathbf{W} + \eta(\frac{1}{n_s} \Delta\mathbf{W})$, $\mathbf{a} = \mathbf{a} + \eta(\frac{1}{n_s} \Delta\mathbf{a})$, $\mathbf{b} = \mathbf{b} + \eta(\frac{1}{n_s} \Delta\mathbf{b})$.
}

V. EXPERIMENTS

The main content of this section is to verify the effectiveness of the proposed algorithm in some public data sets. First of all, a public 3D model library was chosen as the data set. The experimental results are shown in Table 5 and Fig. 5. Finally, we provide analysis of the algorithm according to the experiment results.

A. Dataset

The 3D shape retrieval and analysis team of Princeton University combined the experimental database of the computer graphics and image processing laboratory of Germany University of Leipzig and found a 3D model through network crawling, establishing a standard experimental database after converting those models into a uniform format. This database included 1,814 models, covering most types of models common in daily life.

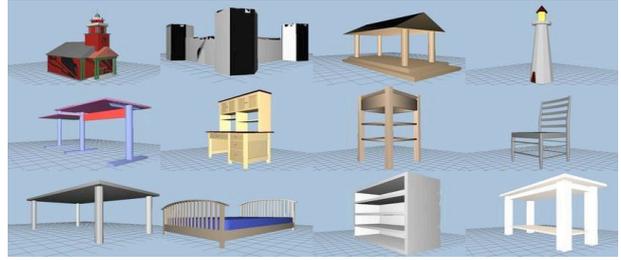


Figure 3. Some 3D models of PSB database

The PSB model database uses artificial classification according to the natural attributes of the model. The previous experiments demonstrated that the different classification methods had a different impact on the retrieval results. So the models provided coarse to fine hierarchical classification to ensure that the number of each species was balanced. To prevent certain types of models from impacting the fairness of the assessment, the 1,814 models were divided into a training set and a testing set, each with 907 models. In the division, not only was the total averaged, each type also tried to ensure uniformity. The purpose of the division was to adjust the parameters of the method using a training set and then doing a search test using the test set to determine the stability of the parameter adjustments.

B. Feature Description

Before classifying the 3D models, we first used the ray-based feature extraction algorithm to extract the features of the 3D model. In our experiments, we set the parameter N equal to 12 and the parameter M equal to 3. Then each 3D model was described with 36 dimension features. We choose four 3D models (specifically, m819, m820, m1713, and m1716) to show the features description.

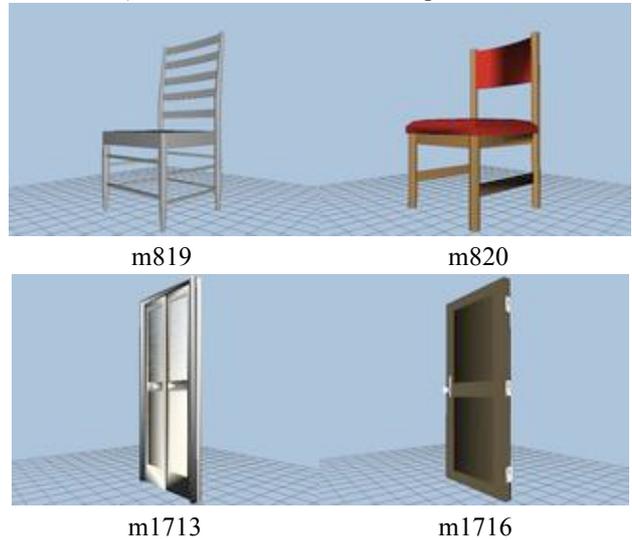


Figure 4. Four 3D models from two classes of PSB database

The features description of the four models is shown in Table 4. Through the features we can see that the models of same class have similar features. For example, the chair models of m819 and m820 are close in feature space. On the other hand, the features of the models from different classes are also much different from each other. For example, chair models m819 and m820 are different from the door models m1713 and m1716.

Table 4. The feature description of 3D models

Model	Feature description											
m819	0.636466	0.218169	0.807545	0.388827	0.845511	0.464452	0.971912	0.476198	1.194971	0.553238	1.472580	0.609615
	0.267773	0.273963	0.302317	0.338132	0.433495	0.532739	0.573372	0.469454	0.459359	0.563853	0.672686	0.646660
	0.076127	0.112836	0.158655	0.173207	0.208316	0.217182	0.237103	0.247347	0.264285	0.283031	0.304950	0.325778
m820	0.679774	0.244436	0.789147	0.384267	0.758758	0.457001	0.756461	0.588699	0.886894	0.739447	1.111364	0.803619
	0.454488	0.310021	0.518568	0.550957	0.601673	0.677295	0.799724	0.909854	0.824614	1.033332	0.914540	1.147084
	0.123021	0.195678	0.242293	0.256121	0.260974	0.231956	0.238723	0.286496	0.339639	0.387788	0.418323	0.440148
M1713	0.445214	0.110099	0.631029	0.137339	0.690518	0.153374	0.722047	0.164315	0.742181	0.175988	0.774874	0.182429
	0.258969	0.031381	0.413673	0.057124	0.419756	0.059253	0.446875	0.084505	0.500500	0.074432	0.504499	0.093340
	0.224942	0.005889	0.409693	0.011503	0.441677	0.014024	0.408345	0.013486	0.443955	0.017671	0.448484	0.020841
M1716	0.496520	0.076820	0.633375	0.115738	0.672814	0.151405	0.707474	0.186319	0.745610	0.218046	0.772542	0.244154
	0.253730	0.032720	0.373184	0.087912	0.446248	0.100971	0.439525	0.118001	0.506903	0.145534	0.501521	0.157119
	0.103795	0.148413	0.171202	0.168255	0.201311	0.229009	0.284761	0.309115	0.316267	0.326010	0.281908	0.304867

C. Classification Performance evaluation

We used the precision (P) as the standard to evaluate the performance of 3D model classification accuracy:

$$P = \frac{a}{a + b}$$

Among them, a is the number of models that were correctly judged to the class, b is the number of models wrongly judged to the class, c is the number of models judged to other classes.

D. Experimental results and analysis

The different number of training sets was selected in this paper to compare with other algorithms. We used 100 samples of training sets when analyzing the impact on the performance of the algorithm itself and the initial parameters in the RBM algorithm set as: the learning rate is equal to 0.1; the rate of decline is equal to 0.3; the cycle number is equal to 50. In the algorithm, the hidden layer unit is a feature representation of the input data. Therefore, the hidden layer unit numbers can be preset to 10 and then increased according to the accurate rate. The parameters in the DBN algorithm are set as: the contrastive divergence rate of decline is equal to 0.1, the number of cycles is equal to 100, the rate of fine-tuning is equal to 0.1, the number of times of fine-tuning is equal to 100, and the number of output unit is equal to 36.

The experiment was performed to compare the algorithm of this paper with several commonly used algorithms; DBN achieved better performance than KNN, NN and SVM on the classification accuracy. Table 5 lists the average accuracy of several different algorithms.

Table 5. The accuracy of four algorithms

Algorithm	Accuracy (%)
KNN	78.5
NN	75.0
SVM	82.3
DBN	85.5

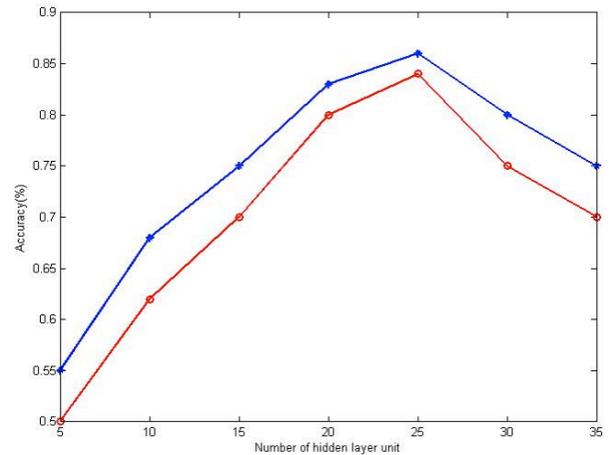


Figure 5. The impact on accuracy with different hidden unit number

The number of hidden layer units has a greater impact on the accuracy of the results of the classification in the RBM algorithm. Since the number of the hidden layer units determines whether the data characteristics can be correctly represented, if the number of hidden unit layers is too little, then the feature vectors are mostly 1; only a small part is 0, which is unable to correctly describe the characteristics. On the contrary, if the number of elements is too great, the feature vector is too sparse to describe the characteristics of the data effectively. It can be seen from the figure that when the input unit is 36 dimensions, the optimal classification results can be found when the hidden units are 25. More hidden units will increase the training time, but will not improve the accuracy of classification. At present, there is no method of quickly setting the number of hidden units; only experience and continuing to experiment to find the optimal number of hidden units is the next research direction.

VI. CONCLUSION

In this paper, a method of deep learning-based 3D model classification under a BIM environment was proposed. Deep learning is a new research direction in the field of machine learning for Computer Science, which is committed to machine learning to master the ability to simulate human learning and analysis as well as image, sound and text data. A ray-based 3D model feature extraction algorithm was used to extract features and then to combine the DBN to classify the models. Experiments show the impact of the number of hidden layer units and

the number of times to fine tune the classification accuracy. Finally, a DBN algorithm was compared with several commonly used in the classification method; the results showed that on the premise of a large number of samples, the DBN algorithm performed better than other classification algorithms.

REFERENCES

- [1] Bentley Systems Co., Ltd. "Advanced concepts of building industry--Building Information Modeling," *Intelligent Building and City Information*, vol.103, no.6, pp. 112-124, 2005.
- [2] Minghua Zheng. "Development Trends of BIM application in China," *China Exploration & Design*, vol. 9, pp. 74-75, 2014.
- [3] Zhang, Shan-Hui; Yang, Chao-Ying; Thomas, Steffen. "Design knowledge and process management method based on 3D CAD system", *Journal of Digital Information Management*, v.12, n.3, pp.192-200, June 2014.
- [4] E. Sankar and K. Ashokkumar. "Efficient Method for Parallel Process and Matching of Large Data set in Grid Computing Environment", *Journal of Engineering Science and Technology Review*, v.7, n.4, pp.109-113, 2014.
- [5] IDMC, I. "Information Delivery Manual Guide to Components and Development Methods," http://www.iai.no/idm/idm_resources/idm_methods_guides/IDMC_004_1_2.pdf.
- [6] Sciences, N.I.O.B. "United States National Building Information Modeling Standard Version 1-Part 1: Overview, Principles, and Methodologies," http://www.wbdg.org/pdfs/NBIMSv1_p1.pdf, 2011.
- [7] Jianping Zhang. "Research and application of BIM Technology," *Construction Technology*, vol.02, pp. 15-18, 2011.
- [8] Xiaojuan Xue, Xi Zhao, Jiemin Ding. "The State of the Art of the Application of BIM in Building Structural Design," *Structural Engineers*, vol.27, no.1, pp. 14-18, 2011.
- [9] Xiangrong Li. "BIM(Building Information Modeling) Applications In Real Estate Project Management Information," *Beijing Jiaotong University*, 2011.
- [10] Aijuan Wang. "Modeling of Schedule Risk Analysis for Construction Projects Based-on BIM," *Harbin Institute of Technology*, 2011.
- [11] Zhe Lou. "Modeling of Cost Estimation Software for Buildings Based on BIM Techniques," *Tsinghua University*, 2009.
- [12] Zhenzhong Hu, Jianping Zhang, Xin Zhang. "Construction collision detection for site entities based on 4-D space-time model," *Journal of Tsinghua University(Science and Technology)*, vol.50, no.6, pp. 820-825, 2010.
- [13] Zhe Fan. "BIM-Based 4D Resources Dynamic Management in Construction," *Tsinghua University*, 2010.
- [14] Shaohua Jiang, Qian Li. "BIM-based Construction Document Management System Design," *Journal of Engineering Management*, vol.26, no.1, pp. 59-63, 2012.
- [15] Jianping Zhang, Fangqiang Yu, Ding Li. "A Modeling Technology of integrated BIM for Building Lifecycle," *Journal of Information Technology in Civil Engineering and Architecture*, vol.4, no.1, pp. 6-14, 2012.
- [16] Ip, C.Y., Regli, W.C. "Manufacturing classification of CAD models using curvature and SVMs," *Shape Modeling and Applications, 2005 International Conference. IEEE*, pp. 361-365, 2005. <http://dx.doi.org/10.1109/smi.2005.27>
- [17] Hou, S., K. Lou, and K. Ramani. "SVM-based semantic clustering and retrieval of a 3D model database," *Journal of Computer Aided Design and Application*, vol.2, no.1, pp. 155-164, 2005. <http://dx.doi.org/10.1080/16864360.2005.10738363>
- [18] Barutcuoglu, Z. and C. DeCoro. "Hierarchical shape classification using Bayesian aggregation," *Shape Modeling and Applications*, pp. 44-48, 2006. <http://dx.doi.org/10.1109/smi.2006.15>
- [19] Wei, W., Y. Yang, and J. Lin. "Color-based 3d model classification using hopfield neural network," *Computer Science and Software Engineering, International Conference on. IEEE*, pp. 883-886, 2008. <http://dx.doi.org/10.1109/csse.2008.1177>
- [20] Philipp-Foliguet, S., M. Jordan, and L. Najman. "Artwork 3D model database indexing and classification," *Pattern Recognition*, vol.44, no.3, pp. 588-597, 2011. <http://dx.doi.org/10.1016/j.patcog.2010.09.016>
- [21] Weiming WANG, Xiuping LIU, Ligang LIU. "Shape matching and retrieval based on multiple feature descriptors," *Computer Aided Drafting, Design and Manufacturing*, vol.23, no.1, pp. 60-67, 2013.
- [22] HINTON, G.E., S. OSINDERO, and Y.W. TEH. "A fast learning algorithm for deep belief nets," *Neural Computation*, vol.18, no.7, pp. 1527-1554, 2006. <http://dx.doi.org/10.1162/neco.2006.18.7.1527>
- [23] BENGIO, Y., et al. "Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems. MA: MIT Press*, vol.19, pp. 153-160, 2007.
- [24] RANZATO, M., et al. "Efficient learning of sparse representations with an energy-based model," *Cambridge, in Advances in Neural Information Processing Systems. MA: MIT Press*, pp. 1137-1144, 2007.
- [25] ERHAN, D., Y. BENGIO, and A. COURVILLE. "Why does unsupervised pre-training help deep learning," *Journal of Machine Learning Research*, vol.11, pp. 625-660, 2010.
- [26] GLOROT, X. and Y. BENGIO. "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research-Proceedings Track*, pp. 249-256, 2010.
- [27] Y, Y.B. "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol.2, no.1, pp. 1-127, 2009. <http://dx.doi.org/10.1561/22000000006>
- [28] GOODFELLOW, I., et al. "Measuring invariances in deep networks," *Advances in Neural Information Processing Systems*, vol. 22, pp. 646-654, 2009.
- [29] HINTON, G.E. "A practical guide to training restricted Boltzmann machines," *Neural Networks: Tricks of the Trade, Reloaded, Springer*, 2013.
- [30] BENGIO, Y. "Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*," *Lecture Notes in Computer Science*, vol.7700, pp. 437-478, 2012. http://dx.doi.org/10.1007/978-3-642-35289-8_26
- [31] Fu Yuchen, Zhou Dongru. "Computer Graphics: Theory, Method and Applications," *Wuhan: Huazhong University of Science and Technology Press*, 2003.

AUTHORS

WANG Li is with the School of Mechanics & Civil Engineering and State Key Laboratory for Geomechanics and Deep Underground Engineering,, China University of Mining and Technology, Xuzhou, 221116 China.(e-mail: wangliolly@126.com).

Zhikai Zhao is currently a full-time researcher in the Internet of Things (IoT) Research Center, China University of Mining and Technology (CUMT). (email: zhaozhikai@cumt.edu.cn).

This work was supported by the project of the Natural Science Foundation of Jiangsu Province, China (No. BK20140216),the China Fundamental Research Funds for the Central Universities (No. 2014QNB45), the Sixth Batch "sail plan" of China University of Mining and Technology Project (No. [2013]9) and the Jiangsu Province Postdoctoral Scientific Research Fund Project (No. 1402059C). Submitted, April, 20, 2015. Published as resubmitted by the authors on September, 16, 2015.