

Research Article

3D Model Classification Based on Bayesian Classifier with AdaBoost

Xue-Yao Gao , Kai-Peng Li , Chun-Xiang Zhang , and Bo Yu 

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Correspondence should be addressed to Chun-Xiang Zhang; z6c6x666@163.com

Received 10 May 2021; Accepted 2 November 2021; Published 30 November 2021

Academic Editor: Youssef N. Raffoul

Copyright © 2021 Xue-Yao Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the exponential increase of 3D models, 3D model classification is crucial to the effective management and retrieval of model database. Feature descriptor has important influence on 3D model classification. Voxel descriptor expresses surface and internal information of 3D model. However, it does not contain topological structure information. Shape distribution descriptor expresses geometry relationship of random points on model surface and has rotation invariance. They can all be used to classify 3D models, but accuracy is low due to insufficient description of 3D model. This paper proposes a 3D model classification algorithm that fuses voxel descriptor and shape distribution descriptor. 3D convolutional neural network (CNN) is used to extract voxel features, and 1D CNN is adopted to extract shape distribution features. AdaBoost algorithm is applied to combine several Bayesian classifiers to get a strong classifier for classifying 3D models. Experiments are conducted on ModelNet10, and results show that accuracy of the proposed method is improved.

1. Introduction

With the development of multimedia technology, 3D model has been applied in many fields such as mechanical design, computer vision, construction industry, entertainment, medical treatment, education, e-commerce, and molecular biology [1]. The number of 3D models is becoming larger and larger in our lives. Therefore, the classification of 3D models becomes more and more important. Now, scholars at home and abroad are studying 3D model classification. 3D model classification is divided into voxel-based classification method, view-based one, and point cloud-based one.

In voxel-based classification method, 3D model is expressed as the distribution of 3D voxel grid. 3D model is denoted by 3D matrix, from which 3D CNN is used to extract features for classifying 3D models [2]. In view-based one, 3D model is converted into a series of 2D images, from which 2D deep learning algorithm is applied to extract view features. Then, view features are merged to represent 3D model for model classification [3]. In point cloud-based one, point cloud is preprocessed to solve sparseness and disorder.

Then, CNN is used to extract features from point cloud for classifying 3D models [4].

AdaBoost is an ensemble learning technique which converts weak classifier into strong one [5, 6]. AdaBoost provides a simple and useful method to generate strong classifier. Strong classifier depends on the diversity between weak classifiers and the performance of weak classifiers [7].

Main contributions of this paper are shown as follows:

- (1) Voxel descriptor is used to express surface and internal information of 3D model. Shape distribution descriptor is adopted to express geometry relationship of random points on model surface. Two kinds of descriptors are applied to describe 3D models completely from different perspectives.
- (2) 3D CNN is used to extract voxel features, and 1D CNN is adopted to extract shape distribution features. These features are merged to improve the performance of 3D model classifier.
- (3) AdaBoost algorithm is used to obtain multiple Bayesian classifiers by iterations. The weighted

combination of Bayesian classifiers is used as strong 3D model classifier.

This paper is organized as follows. Related work is reported in Section 2. Voxel and shape distribution features of 3D model are given in Section 3. The proposed method is described in Section 4. Experimental results are given and analyzed in Section 5. Conclusion is presented in Section 6.

2. Related Work

Feature descriptors are used to express shape and structure of 3D model. Machine learning algorithms are applied to classify 3D models based on feature descriptors. According to feature descriptors, 3D model classification is divided into voxel-based classification method, view-based one, and point cloud-based one.

In voxel-based classification method, 3D voxel matrix is used to describe 3D model. Sirma et al. [8] voxelize and scale 3D models with certain parameters. Two CNNs are combined to classify 3D models. Kang et al. [9] propose a voxel-based method to classify 3D pole-like objects by analyzing spatial characteristics of objects. Plaza et al. [10] use multilayer perceptrons to classify voxels by analyzing spatial distribution of inner points. They identify tubular structures and flat surfaces for unstructured and natural environments. Mughees and Tao [11] present hyper-voxel stacked autoencoder to exploit spatial context within similar contiguous pixels to classify hyperspectral images. Babahajiani et al. [12] design a novel street scene semantic recognition framework, which takes advantage of 3D point cloud captured by a high definition laser scanner. Ahmad et al. [13] classify urban scenes based on a super-voxel segmentation of 3D data obtained from LiDAR sensors. Agnello et al. [14] select useful MRI data according to fuzziness values of Fuzzy C-Means and train the fully connected cascade neural network to classify brain tissues into White Matter, Gray Matter, and Cerebrospinal Fluid in an unsupervised way.

In view-based classification method, 3D model is projected into a series of 2D views from different directions. Chen et al. [15] propose multimodal support vector machine to combine sift descriptor, outline Fourier transform descriptor, and Zernike moments descriptor to discriminate multiple classes of object, which considers the independence of each modality and the interrelation between them. Liu et al. [16] design a semantic and context information fusion network, which extracts contextual information in continuous view sequence and semantic information of individual view. Kim et al. [17] give a viewpoint and image-resolution estimation method for view-based 3D shape retrieval from point cloud queries. Automatic selection method of viewpoint and image resolution is proposed based on data acquisition rate and density calculations. Huang et al. [18] design a view-based weight network for 3D object recognition where view-based weights are assigned to projections. Shajahan et al. [19] present a multiview CNN with self-attention, which takes multiple views of a roof point cloud as input and outputs category of the roof. Zhou et al. [20] design multiview saliency guided deep neural network to

retrieve and classify 3D objects. It selects representative views, exploits multiview context to compute similarities, and discovers discriminative structure of multiview sequence. Zhou et al. [21] propose CNN trained by a single polar view of 3D shape to obtain polar view representation, which is used to classify and retrieve 3D shapes. Yang et al. [22] design a network structure combining multiview CNN, extreme learning machine autoencoder, and ELM classifier, which converts 3D model into a single compact feature descriptor.

In point cloud-based classification method, neural network is used to extract features from point cloud. Lee et al. [23] propose DenX-Conv to improve accuracy of object classification and secure the connectivity of points from raw point cloud. Hu et al. [24] present a convolution operation and design a feature transformation module to capture local structural information and prior information from 3D point cloud. Li et al. [25] design a multiscale receptive field graph attention network with semantic features of local patch for point cloud, which captures abundant features of point cloud. Zhang et al. [26] propose a hybrid feature CNN to describe features of 3D point cloud, which can handle 3D point cloud data with unstructured and unordered properties. Li et al. [27] use X-Conv transformation to solve disorder of point clouds, and apply CNN to classify 3D point clouds. Huang et al. [28] project features of disordered points into an ordered sequence of feature vectors and use recurrent neural network to classify 3D point clouds.

Boosting is a very successful technology for solving two classification problems. Hastie et al. [29] develop an algorithm which extends AdaBoost to solve multiple classification problem. Wang and Yan [30] propose a gait classification framework based on CNN ensemble. Wu and Nagahashi [31] present parameterized AdaBoost whose parameter is devised to penalize the misclassification of instances. Compared with Real AdaBoost, the proposed method achieves faster convergence and improves the generalization ability. Li et al. [32] give a novel multifeature joint learning ensemble framework which combines global features with local key features to consider multiple labels of expressions in facial action units. Taherkhani et al. [33] propose AdaBoost-CNN, which combines AdaBoost and CNN to deal with data imbalance. Kavitha et al. [34] merge Hough transform with the improved Dragonfly algorithm to address face recognition problem in unconstrained conditions. Feng [35] uses AdaBoost algorithm to strengthen Bayesian classifier for classifying text emotion accurately. Xiao et al. [36] propose a multitemporal ensemble learning framework to extract snow cover from images of mountain areas. Kaur and Kumar [37] extract zoning features, diagonal features, intersection features, and open-end point features from word images. Majority voting scheme of K-nearest neighbor, support vector machine, random forest, and AdaBoost algorithm are used to recognize handwritten Gurmukhi words. Chen et al. [38] integrate deep learning and subspace-based ensemble learning to boost the performance of HSI classification. Lee et al. [39] propose a new sparse coding method based on an ensemble of image patches. Zhang et al. [40] give a bioinspired fault diagnosis

method based on neural P system with belief AdaBoost for oil-immersed power transformer. Oyewole and Olugbara [41] design an image classification architecture including data preprocessing, feature extraction, dimensionality reduction, and ensemble of machine learning methods. Song et al. [42] propose an innovative classification method which combines texture features of images with extreme learning machine. Zeng et al. [43] present a learning-based multiple pooling fusion method, in which multiview feature maps projected from 3D model are compiled as a simple and effective feature descriptor. Li et al. [44] give a novel unsupervised hashing method to solve large-scale image retrieval on scenarios of single information source.

These 3 methods have their own shortcomings. More research is needed no matter which method is used for 3D model classification. For voxel-based classification method, network structure needs to be optimized, and voxel resolution should be set. At the same time, there is less research on fusing voxel features with other ones. Most networks use softmax as classifier, and there is less research on using strong classifier obtained by AdaBoost algorithm to classify 3D models.

3. Voxel and Shape Distribution Features of 3D Model

Voxel is basic data unit in volume graphics which can describe volume model. 3D model is denoted as polygon mesh structure which is a collection of vertices, edges, and faces. This paper voxelizes polygon mesh structure and represents 3D model with voxel feature. Here, a simple and effective method is used to voxelize 3D model. The resolution of model voxelization is set to $N * N * N$. AABB of 3D model is determined. Vertex coordinates of polygon mesh model are traversed to obtain the maximum $(x_{\max}, y_{\max}, z_{\max})$ and the minimum $(x_{\min}, y_{\min}, z_{\min})$ as shown in Figure 1. Here, x_{\max} , x_{\min} represent, respectively, the maximum and minimum in x -axis direction of all vertex coordinates. AABB is a cube which is composed of diagonal vertices $(x_{\max}, y_{\max}, z_{\max})$ and $(x_{\min}, y_{\min}, z_{\min})$. The size of voxel unit is set to $V * V * V$, where

$$V = \frac{\max(x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min})}{N} \quad (1)$$

AABB is divided equidistantly according to V along x -axis, y -axis, and z -axis. If there is a side whose length is less than V , it is rounded up. Because the size of voxel unit is $V * V * V$, AABB is divided into $X_N * Y_N * Z_N$ voxel units. X_N , Y_N , Z_N represent, respectively, layer number of AABB divided along x -axis, y -axis and z -axis, as shown in the following formulas:

$$X_N = \text{ceil}\left(\frac{x_{\max} - x_{\min}}{V}\right), \quad (2)$$

$$Y_N = \text{ceil}\left(\frac{y_{\max} - y_{\min}}{V}\right), \quad (3)$$

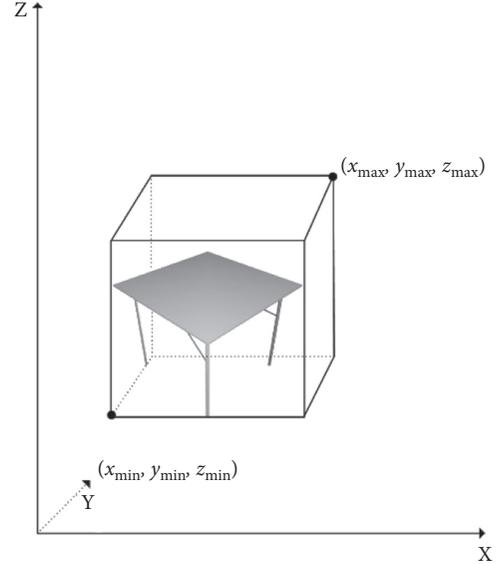


FIGURE 1: AABB of "table."

$$Z_N = \text{ceil}\left(\frac{z_{\max} - z_{\min}}{V}\right), \quad (4)$$

where $\text{ceil}(\cdot)$ is ceiling function. Vertex (x, y, z) of 3D model corresponds to a voxel unit in the X_V -th row, Y_V -th column, and Z_V -th layer. X_V , Y_V , and Z_V are computed, respectively, as shown in the following formulas:

$$X_V = \text{ceil}\left(\frac{x - x_{\min}}{V}\right), \quad (5)$$

$$Y_V = \text{ceil}\left(\frac{y - y_{\min}}{V}\right), \quad (6)$$

$$Z_V = \text{ceil}\left(\frac{z - z_{\min}}{V}\right). \quad (7)$$

If a voxel unit is occupied by 3D model, its value is 1 and it is called nonempty voxel. Otherwise, its value is 0 and it is called empty voxel. 3D model is denoted as a binary string. In this paper, the voxelization resolution of 3D model is set to $24 * 24 * 24$, and the size of voxel matrix is $24 * 24 * 24$. In order to extract features from model surface accurately, the size of voxel matrix is expanded into $30 * 30 * 30$ with 0.

In order to describe shape characteristics of 3D model on the whole, this paper uses $D1$, $D2$, $D3$, and $A3$ shape functions which have good rotation invariance. $D1$, $D2$, $D3$, and $A3$ shape functions are described as shown in Figure 2. O is model centroid. P_1 , P_2 , and P_3 are 3 random sampling points on 3D model surface. $D1$ shape function is the distance between O and P_1 . $D2$ shape function is the distance between P_1 and P_2 . $D3$ shape function is the square root of area formed by P_1 , P_2 , and P_3 . $A3$ shape function is angle A formed by P_1 , P_2 , and P_3 .

$Points = \{(x_1, y_1, z_1), \dots, (x_i, y_i, z_i), \dots, (x_n, y_n, z_n)\}$ is point set on 3D model surface. Here, (x_i, y_i, z_i) is point coordinates. $Bins$ is the number of statistical intervals, and $BinSize$ is the length of interval.

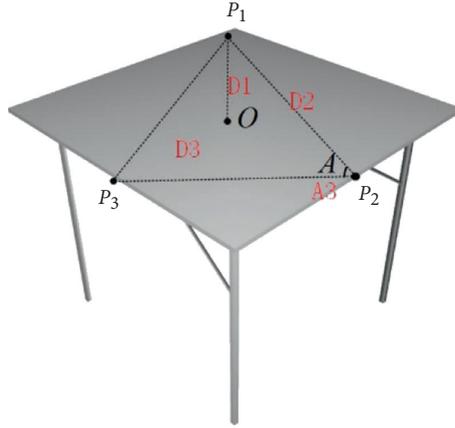


FIGURE 2: Shape functions.

Extract a set of points $NPoints = \{ai_1, ai_2, \dots, ai_N\}$ from Points by N sampling. The set of D1 shape distribution

features is $\{D1_{v_1}, \dots, D1_{v_i}, \dots, D1_{v_{Bins}}\}$. $D1_{v_i}$ is computed as

$$D1_{v_i} = \{(P_1 | \text{dist}(P_1, O) \in (\text{BinSize} * (i - 1), \text{BinSize} * i), P_1 \in NPoints)\}, \quad (8)$$

where $\text{BinSize} = \max(\{\text{dist}(P_1, O) | P_1 \in NPoints\})/N$. Function $\text{dist}(\cdot)$ is Euclidean distance between points P_1 and O . O is the centroid of 3D model.

Construct a set of point pairs $NPoints = \{(ai_1, bi_1), (ai_2, bi_2), \dots, (ai_N, bi_N)\}$ by N sampling from Points. The set of D2 shape distribution features is $\{D2_{v_1}, \dots, D2_{v_i}, \dots, D2_{v_{Bins}}\}$. $D2_{v_i}$ is calculated as

$$D2_{v_i} = \{(P_1, P_2) | \text{dist}(P_1, P_2) \in (\text{BinSize} * (i - 1), \text{BinSize} * i), (P_1, P_2) \in NPoints\}, \quad (9)$$

where $\text{BinSize} = \max(\{\text{dist}(P_1, P_2) | (P_1, P_2) \in NPoints\})/N$.

Construct a set of point triples $NPoints = \{(ai_1, bi_1, ci_1), \dots, (ai_n, bi_n, ci_n)\}$ by N sampling from Points. The set of D3

shape distribution features is $\{D3_{v_1}, \dots, D3_{v_i}, \dots, D3_{v_{Bins}}\}$. $D3_{v_i}$ is computed as

$$D3_{v_i} = \{(P_1, P_2, P_3) | \text{herson}(P_1, P_2, P_3) \in (\text{BinSize} * (i - 1), \text{BinSize} * i), (P_1, P_2, P_3) \in NPoints\}, \quad (10)$$

where $\text{BinSize} = \max(\{\text{herson}(P_1, P_2, P_3) | (P_1, P_2, P_3) \in NPoints\})/N$. Function $\text{herson}(\cdot)$ is used to compute the area of triangle (P_1, P_2, P_3) as shown as

$$\text{herson}(P_1, P_2, P_3) = \sqrt{p * (p - a) * (p - b) * (p - c)}, \quad (11)$$

$$p = \frac{1}{2} * (a + b + c), \quad (12)$$

where $|a| = \text{dist}(P_1, P_2)$, $|b| = \text{dist}(P_1, P_3)$, $|c| = \text{dist}(P_2, P_3)$.

The set of A3 shape distribution features is $\{A3_{v_1}, \dots, A3_{v_i}, \dots, A3_{v_{Bins}}\}$. $A3_{v_i}$ is calculated as

$$A3_{v_i} = \{(P_1, P_2, P_3) | \text{cosine}(P_1, P_2, P_3) \in (\text{BinSize} * (i - 1), \text{BinSize} * i), (P_1, P_2, P_3) \in NPoints\}, \quad (13)$$

where $BinSize = \max(\{\cosine(P_1, P_2, P_3) | (P_1, P_2, P_3) \in NPoints\})/N$. $\cosine(\cdot)$ is cosine function. Angle A formed by sides a , b , and c is shown in (14). A is angle corresponding to side a .

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc}. \quad (14)$$

4. 3D Model Classification Based on AdaBoost Algorithm

3D model classification framework is shown in Figure 3.

3D CNN is applied to extract deep features from voxel descriptor. 1D CNN is used to extract deep features from shape distribution descriptors. Then, Merge layer fuses voxel feature and shape distribution features. Finally, the fused feature is input into Bayesian classifier.

4.1. 3D CNN for Extracting Voxel Feature. 3D CNN is used to extract voxel feature from voxel matrix including 4 convolutional layers as shown in Figure 4.

Dropout layer is added between two convolutional layers. The input of 3D CNN is voxel matrix whose size is $30 * 30 * 30$ and its output is a $1024 * 1$ feature vector. In Figure 4, Conv1 is the first convolutional layer, and the size of convolutional kernel is $6 * 6 * 6$. The number of convolutional kernels is 16, and the sliding step is 2. The size of convolutional kernel in Conv2, Conv3, and Conv4 is $5 * 5 * 5$, and the sliding step is 1. The number of convolutional kernels is, respectively, 64, 128, and 1024.

In Figure 4, $x_{V_{i,j,k}}^l$ is element in the l -th convolutional layer. Here, i, j , and k represent, respectively, the i -th row, the j -th column, and the k -th layer in voxel matrix, where $0 \leq i < W^l$, $0 \leq j < L^l$, and $0 \leq k < H^l$. Here, W^l , L^l , and H^l represent, respectively, the length, width, and height of voxel matrix in the l -th convolutional layer. $x_{V_{i,j,k}}^l$ is computed as

$$x_{V_{i,j,k}}^l = f(e_V) = f\left(\sum_{r \in M_i^l} \sum_{s \in M_j^l} \sum_{t \in M_k^l} x_{V_{r,s,t}}^{l-1} * w_{V_{r,s,t}}^l + b_{V_{i,j,k}}^l\right), \quad (15)$$

where $f(\cdot)$ represents activation function as shown in (16) and e_V is net activation. $x_{V_{i,j,k}}^l$ is convolved from feature maps in the $(l-1)$ -th layer. Here, $w_{V_{r,s,t}}^l$ represents the element in the r -th row, the s -th column, and the t -th layer of convolutional kernel. Indices r, s, t take, respectively, values from M_i^l, M_j^l, M_k^l . $b_{V_{i,j,k}}^l$ represents its bias.

$$f(x) = \max(x, 0) = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (16)$$

4.2. 1D CNN for Extracting Shape Distribution Feature. 1D CNN is used to extract shape distribution feature including 4 convolutional layers as shown in Figure 5. Dropout layer is added between two convolutional layers. The input of 1D CNN is shape distribution descriptors with $1 * 512$, and its output is a 1792-feature vector.

$x_{G_i}^l$ is element in the l -th convolutional layer. 1D convolutional kernel is used, whose size is $1 * 5$. The sliding step is 2. $x_{G_i}^l$ is computed as

$$x_{G_i}^l = f(e_G) = f\left(\sum_{s \in M_i^l} x_{G_s}^{l-1} * w_{G_s}^l + b_{G_i}^l\right), \quad (17)$$

where $f(\cdot)$ represents activation function and e_G denotes net activation. $x_{G_i}^l$ in the l -th layer is convolved from feature maps in the $(l-1)$ -th layer. Index s takes values from M_i^l . $w_{G_s}^l$ represents the s -th element of convolutional kernel in l -th layer, and $b_{G_i}^l$ denotes its bias.

Maximum pooling operation is used in pooling layer as

$$x^l = \max(e) = \max(x^{l-1}), \quad (18)$$

where $\max(\cdot)$ is the operation of taking maximum. The size of pooling kernel is $1 * 2$, and step size is 2.

4.3. Fusing 3D CNN and 1D CNN to Classify 3D Model.

3D CNN is used to extract voxel feature. After voxel feature is processed by Flatten layer, X_V is gotten. 1D CNN is adopted to extract D1, D2, D3, and A3 shape distribution features. After D1, D2, D3, and A3 shape distribution features are processed by Flatten layer, X_G is obtained. Then, X_V and X_G are fused in Merge layer as

$$X = \text{Merge}(X_V, X_G). \quad (19)$$

Bayesian classifier is a simple probabilistic one based on Bayes' theorem with strong independence assumptions between features. The classification principle is that the posterior probability is calculated by Bayesian formula and the category with the largest probability is selected. Bayesian classifier is used as weak one in AdaBoost algorithm to classify 3D models here. Model M has m categories including s_1, s_2, \dots, s_m . Feature of model M is denoted as X_M . The process of Bayesian classifier predicting category of M is described as follows: when $P_{Bayes}(s_i | X_M) \geq P_{Bayes}(s_j | X_M)$, where P_{Bayes} is the output probability of Bayesian model, $j=1, 2, \dots, m$, category of M is determined as s_i . $\text{Classifier}_{Bayes}(X_M)$ is calculated as

$$\text{Classifier}_{Bayes}(X_M) = \arg \max_{i=1,2,\dots,m} (P_{Bayes}(s_i | X_M)). \quad (20)$$

4.4. 3D Model Classification Algorithm Based on 3D and 1D CNNs. 3D model classification framework mainly includes 3 parts. The first part is 3D CNN for extracting voxel features. The second one is 1D CNN for extracting shape distribution features. The third one is strong classifier, $\text{Classifier}(X)$, constructed by AdaBoost algorithm. The process of constructing $\text{Classifier}(X)$ based on Adaboost is shown in Algorithm 1.

The process of classifying model M is described as follows: 3D CNN is used to extract voxel feature. After voxel feature is processed by Flatten layer, X_V is gotten. 1D CNN is adopted to extract D1, D2, D3, and A3 shape distribution features. After D1, D2, D3, and A3 features are processed by

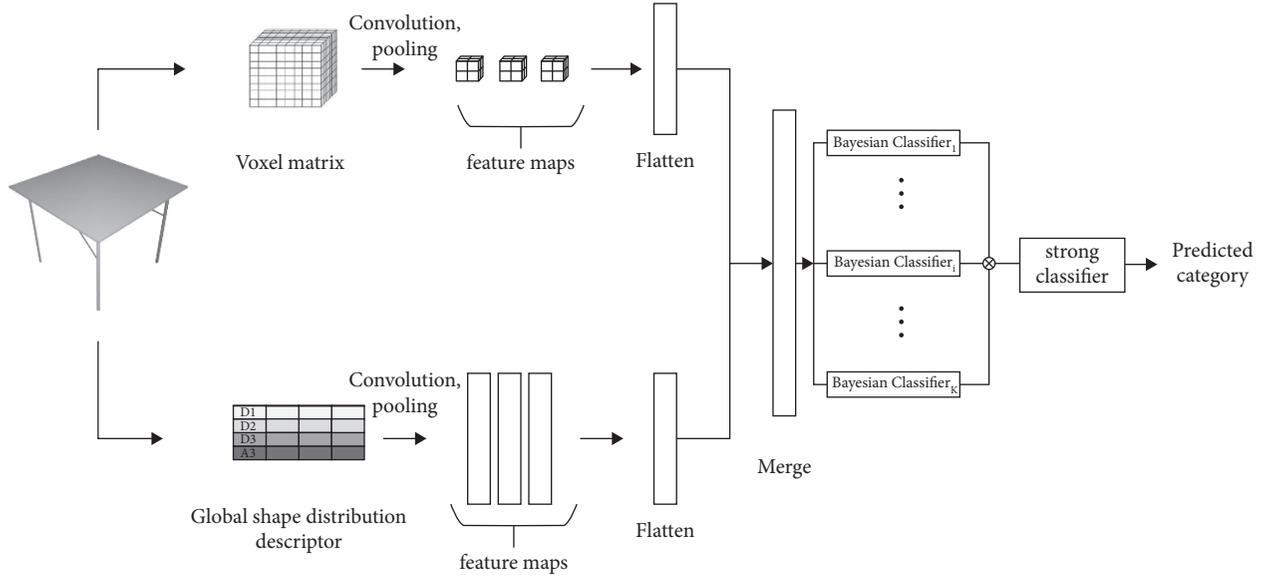


FIGURE 3: 3D model classification framework.

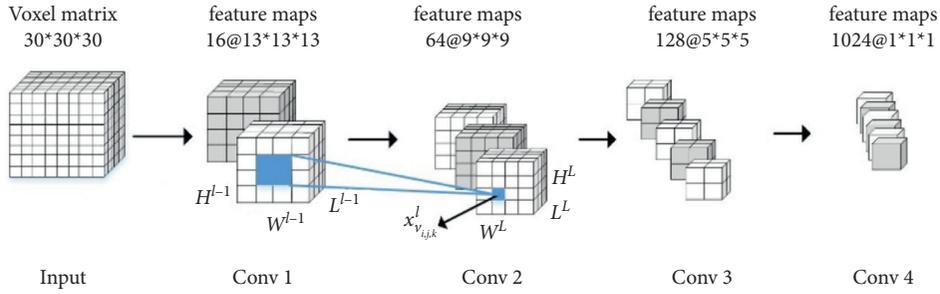


FIGURE 4: 3D CNN structure.

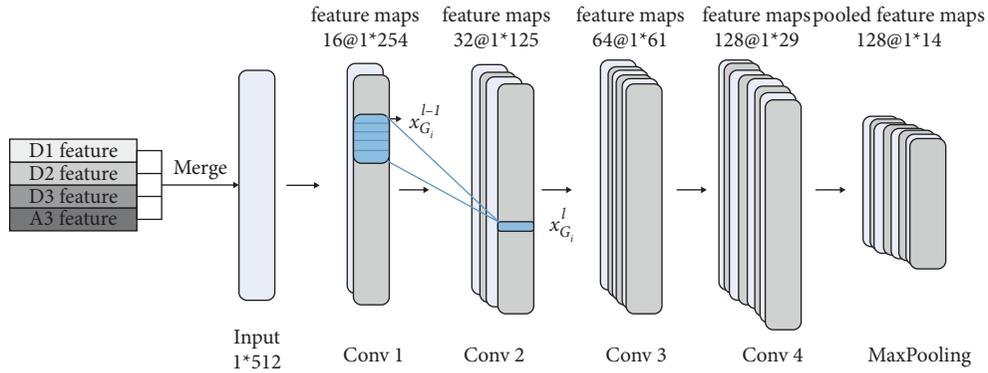


FIGURE 5: 1D CNN structure.

Flatten layer, X_G is gotten. X_V and X_G are fused to obtain X_M as shown in (19). Then, X_M is input into $Classifier(X)$ to determine category of M .

5. Experiments and Result Analysis

5.1. Experimental Data Set. ModelNet10 is used for experiments, which contains 4899 models in 10 categories. Training data contains 3991 models, and test data contains

908 ones. They are applied to train and test the proposed model. Firstly, 3D model is voxelized into voxel units with resolution of $24 * 24 * 24$. Secondly, D1, D2, D3, and A3 shape functions are adopted to express 3D model.

5.2. Analysis of Experimental Results. This paper uses five classifiers. The first one is softmax classifier. The second one is Bayesian classifier. The third one is as follows: AdaBoost

Input: training instances $(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)$.

Output: strong classifier $\text{Classifier}(X)$.

Step 1. Initialize weight w_{1i} of training instances, and iteration time K :

$$w_{1i} = 1/n, i = 1, 2, \dots, n$$

Step 2. for $(k=1; k \leq K; k++)\{$

Step 2.1 Use training instances with weight $w_{k1}, w_{k2}, \dots, w_{kn}$ to train classifier $\text{Classifier}_k(X)$.

Step 2.2 Calculate error rate e_k of $\text{Classifier}_k(X)$:

$$e_k = \sum_{i=1}^n w_{ki} I(\text{Classifier}_k(X_i) \neq y_i)$$

where $I(\text{Classifier}_k(X_i) \neq y_i)$ is 0 or 1. If $\text{Classifier}_k(X_i) \neq y_i$, then $I(\text{Classifier}_k(X_i) \neq y_i) = 1$. Otherwise, $I(\text{Classifier}_k(X_i) \neq y_i) = 0$.

Step 2.3 Calculate weight coefficient α_k of $\text{Classifier}_k(X)$:

$$\alpha_k = 1/2 \log 1 - e_k / e_k$$

Step 2.4 Update weight $w_{k+1,i}$:

$$w_{k+1,i} = w_{ki} / Z_k \exp(-\alpha_k y_i \text{Classifier}_k(X_i))$$

$$Z_k = \sum_{i=1}^n w_{ki} \exp(-\alpha_k y_i \text{Classifier}_k(X_i))$$

}

Step 3. Combine $\text{Classifier}_k(X)$ according to weight coefficient α_k , $(k=1, 2, \dots, K)$:

$$\text{Classifier}(X) = \text{sign} \sum_{k=1}^K (\alpha_k \text{Classifier}_k(X))$$

ALGORITHM 1: The construction of $\text{Classifier}(X)$ based on Adaboost.

algorithm is adopted to obtain multiple Bayesian classifiers by iterations. The weighted combination of Bayesian classifiers is used as strong Bayesian classifier. The fourth one is Decision tree. The fifth one is as follows: AdaBoost algorithm is adopted to obtain multiple Decision trees by iterations. The weighted combination of Decision trees is used as strong Decision classifier.

3D CNN is used to extract voxel feature, and 1D CNN is adopted to extract $D1 + D2 + D3 + A3$ feature. They are combined to obtain voxel + $D1 + D2 + D3 + A3$ feature.

Experiment 1 and Experiment 2 are conducted to testify accuracy of 3D model classification under different inputs. In Experiment 1, softmax classifier is used to classify 3D models based on voxel feature and voxel + $D1 + D2 + D3 + A3$ feature. In Experiment 2, strong Bayesian classifier is adopted based on voxel feature and voxel + $D1 + D2 + D3 + A3$ feature. Training data is used to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 1 and Experiment 2 are shown in Table 1. From Table 1, it can be seen that accuracy of Experiment 2 is better than that of Experiment 1. This is because strong Bayesian classifier makes full use of the diversity between multiple Bayesian classifiers. Under the same classifier, voxel feature is better than $D1 + D2 + D3 + A3$ feature at classification ability. It shows that voxel feature has better discriminative ability than $D1 + D2 + D3 + A3$ feature. The classification ability of voxel + $D1 + D2 + D3 + A3$ feature is the best. This is because the fused feature can describe 3D model completely from different perspectives.

Experiment 3 and Experiment 4 are conducted to testify the influence of shape distribution feature on classification accuracy. In Experiment 3, softmax classifier is used to classify 3D models based on D1 feature, D2 feature, D3 feature, A3 feature, and $D1 + D2 + D3 + A3$ feature. In Experiment 4, strong Bayesian classifier is adopted to classify 3D models based on D1 feature, D2 feature, D3 feature, A3 feature, and $D1 + D2 + D3 + A3$ feature. Training data is used

to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 3 and Experiment 4 are shown in Table 2. From Table 2, it can be seen that under the same classifier, $D1 + D2 + D3 + A3$ feature is better than D1 feature, D2 feature, D3 feature, and A3 feature in classification ability. This is because $D1 + D2 + D3 + A3$ feature has better discriminative ability than D1 feature, D2 feature, D3 feature, and A3 feature. The discriminative ability of D1 feature is the worst.

Experiment 5 and Experiment 6 are conducted to testify classification accuracy under voxel feature with different shape distribution features. In Experiment 5, softmax classifier is used to classify 3D models based on voxel + D1 feature, voxel + D2 feature, voxel + D3 feature, voxel + A3 feature, and voxel + $D1 + D2 + D3 + A3$ feature. In Experiment 6, strong Bayesian classifier is adopted to classify 3D models based on voxel + D1 feature, voxel + D2 feature, voxel + D3 feature, voxel + A3 feature, and voxel + $D1 + D2 + D3 + A3$ feature. Training data is used to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 5 and Experiment 6 are shown in Table 3. From Table 3, it can be seen that under the same classifier, the classification ability of voxel + $D1 + D2 + D3 + A3$ feature is the best. This is because voxel + $D1 + D2 + D3 + A3$ feature can describe shape and structural information of 3D model from different perspectives. Voxel + D2 and voxel + A3 features are better than voxel + D1 and voxel + D3 features in classification ability.

Experiment 7 and Experiment 8 are conducted to testify accuracies of 3D model classification at different ratios of voxel feature to shape distribution feature. The purpose is to find an optimal combination of voxel feature and $D1 + D2 + D3 + A3$ feature. The ratio of voxel feature to $D1 + D2 + D3 + A3$ feature is, respectively, set to 1:16, 1:8, and 1:4. In Experiment 7, softmax classifier based on different ratios of voxel feature to $D1 + D2 + D3 + A3$ feature is used. In Experiment 8, strong Bayesian classifier based on

TABLE 1: Accuracy of 3D model classification under different inputs.

Input	Experiment 1 (softmax classifier)	Experiment 2 (strong Bayesian classifier)
Voxel feature	86.7	89.0
$D1 + D2 + D3 + A3$ feature	53.9	56.3
voxel + $D1 + D2 + D3 + A3$ feature	88.9	90.2

TABLE 2: Influence of shape feature on classification accuracy.

Input	Experiment 3 (softmax classifier)	Experiment 4 (strong Bayesian classifier)
D1 feature	35.7	37.3
D2 feature	43.3	48.2
D3 feature	36.7	37.8
A3 feature	45.3	50.4
$D1 + D2 + D3 + A3$ feature	53.9	56.3

TABLE 3: Classification accuracy under voxel feature with different shape distribution features.

Input	Experiment 5 (softmax classifier)	Experiment 6 (strong Bayesian classifier)
voxel + D1 feature	83.2	88.2
voxel + D2 feature	86.4	89.3
voxel + D3 feature	83.3	88.3
voxel + A3 feature	87.4	89.5
voxel + $D1 + D2 + D3 + A3$ feature	88.9	90.2

TABLE 4: Accuracies of 3D model classification at different ratios.

Ratios	Experiment 7 (softmax classifier)	Experiment 8 (strong Bayesian classifier)
16:1	86.7	89.7
8:1	88.9	90.2
4:1	86.9	89.4

different ratios of voxel feature to $D1 + D2 + D3 + A3$ feature is adopted. Training data is used to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 7 and Experiment 8 are shown in Table 4. From Table 4, it can be seen that under the same classifier, accuracy of 3D model classification increases firstly and then decreases with the increase of the ratio. The reason is that voxel feature contains surface and internal information of 3D model. However, it does not describe topological structure of 3D model. Shape distribution feature expresses geometry relationship of random points on model surface. With the increase of the ratio, shape distribution feature begins to make up for voxel feature in the ability of describing 3D model. However, voxel feature has better classification ability than shape distribution features. Therefore, accuracy of 3D model classification decreases as the ratio increases continuously.

Experiment 9 is conducted to testify the influence of iteration time on classification accuracy. The purpose is to find an optimal iteration time for AdaBoost algorithm. In Experiment 9, strong Bayesian classifier based on voxel + $D1 + D2 + D3 + A3$ feature is used. Training data is used to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 9 are shown in Table 5. From Table 5, it can be seen that as iteration time increases, accuracy of 3D model classification becomes bigger and bigger. When iteration

TABLE 5: Influence of iteration time on classification accuracy.

Iteration time	Accuracy
1	49.6
100	76.6
200	87.3
300	89.2
400	89.8
500	89.6
600	90.0
700	90.0
800	90.2
900	90.1
1000	90.1

TABLE 6: Accuracy of 3D model classification under different weak classifiers.

	Decision tree	Bayesian classifier
One iteration	31.7	49.6
AdaBoost	87.7	90.2

time is 800, the best classification accuracy is achieved. However, as iteration time continues to increase, classification accuracy begins to decline. This is because the overfitting phenomenon occurs in AdaBoost algorithm.

Experiment 10 is conducted to testify accuracy of 3D model classification under different weak classifiers. The

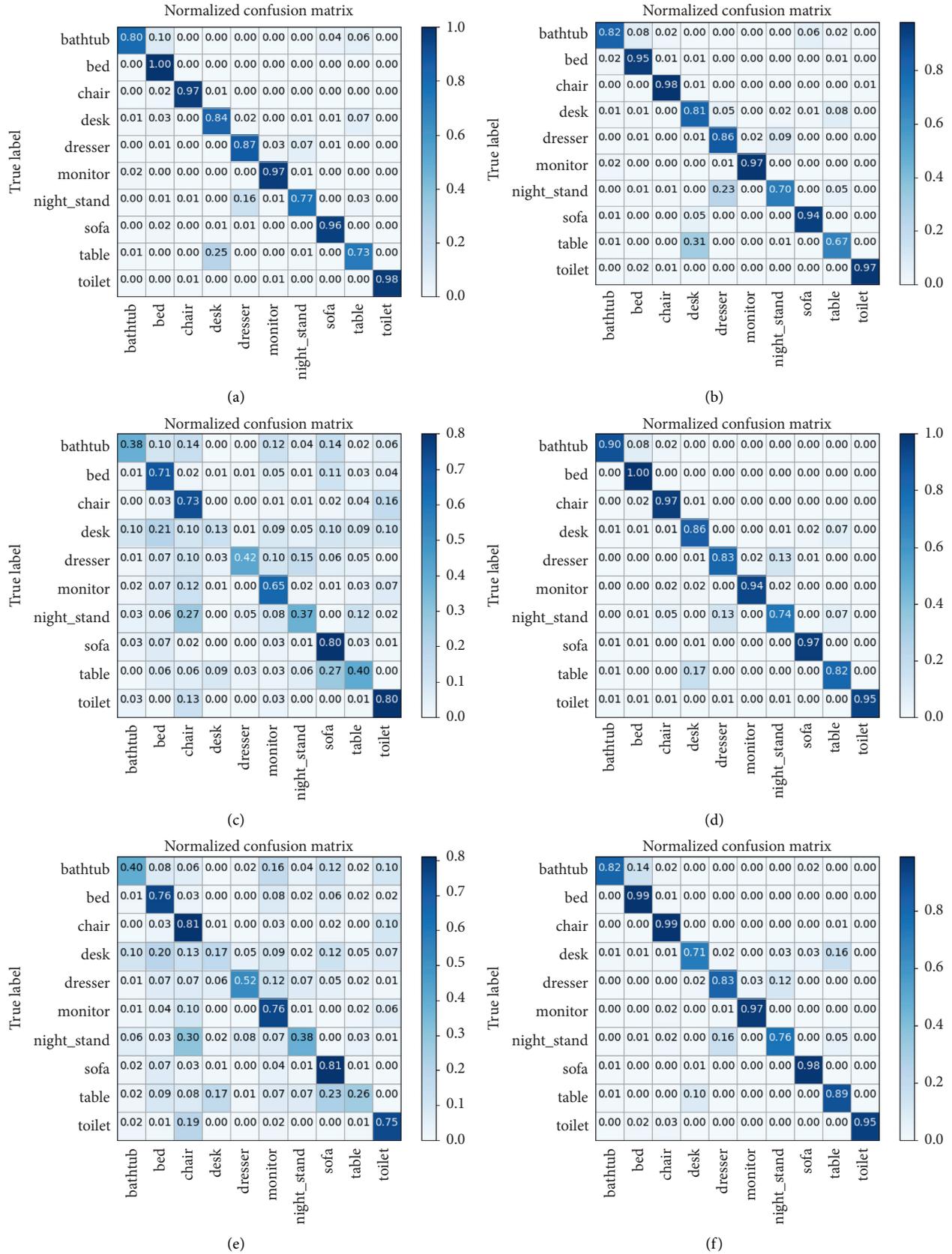


FIGURE 6: Confusion matrix: (a) softmax classifier based on voxel + $D1 + D2 + D3 + A3$ feature; (b) softmax classifier based on voxel feature; (c) softmax classifier based on $D1 + D2 + D3 + A3$ feature; (d) strong Bayesian classifier based on voxel + $D1 + D2 + D3 + A3$ feature; (e) strong Bayesian classifier based on $D1 + D2 + D3 + A3$ feature; (f) strong Bayesian classifier based on voxel feature.

purpose is to find a better weak classifier for AdaBoost algorithm. Bayesian classifier, strong Bayesian classifier, Decision tree, and strong Decision tree based on voxel + $D1 + D2 + D3 + A3$ feature are adopted. Training data is used to optimize 3D model classifier, and the optimized classifier is applied to classify test data. Accuracies of Experiment 10 are shown in Table 6. From Table 6, it can be seen that Bayesian classifier is better than Decision tree, and strong Bayesian classifier is better than strong Decision tree. Strong Bayesian classifier is better than Bayesian classifier, and strong Decision tree is better than Decision tree. This is because AdaBoost algorithm can combine several weak classifiers to construct strong classifier for classifying 3D models better.

In order to testify the influence of features on accuracy of each category, confusion matrices from Experiment 1 and Experiment 2 are constructed. Figures 6(a)–6(c) are obtained by softmax classifier. Figures 6(d)–6(f) are obtained by strong Bayesian classifier.

From Figures 6(a)–6(c), it can be seen that discriminative abilities of voxel feature and voxel + $D1 + D2 + D3 + A3$ feature are better than that of $D1 + D2 + D3 + A3$ feature. Compared with Figures 6(a) and 6(b), voxel + $D1 + D2 + D3 + A3$ feature is better than voxel feature in classification ability for bed class, night_stand class, and table class. This is because $D1 + D2 + D3 + A3$ feature can describe 3D model on the whole and make up for shortcomings of voxel feature. From Figure 6(a), it can be seen that when voxel + $D1 + D2 + D3 + A3$ feature is used, the probability that night_stand is mistakenly classified as dresser is 16%. From Figure 6(b), it can be seen that when voxel feature is used, the probability that night_stand is mistakenly classified as dresser is 23%. Error rate of classification decreases by 7% after $D1 + D2 + D3 + A3$ feature is fused. From Figure 6(a), it can be seen that when voxel + $D1 + D2 + D3 + A3$ feature is used, the probability that table is mistakenly classified as desk is 25%. From Figure 6(b), it can be seen that when voxel feature is used, the probability that table is mistakenly classified as desk is 31%. It shows that the misclassification of table cannot be solved after $D1 + D2 + D3 + A3$ feature is fused. This is because table and desk are similar to each other. The proposed method cannot distinguish this gap well, which results in low accuracy.

From Figures 6(d)–6(f), it can be seen that discriminative abilities of voxel feature and voxel + $D1 + D2 + D3 + A3$ feature are better than that of $D1 + D2 + D3 + A3$ feature. Compared with Figures 6(d) and 6(f), voxel + $D1 + D2 + D3 + A3$ feature is better than voxel feature in classification ability for bathtub class and desk class. From Figure 6(f), it can be seen that when voxel feature is used, the probability that desk is mistakenly classified as table is 16%. From Figure 6(d), it can be seen that when voxel + $D1 + D2 + D3 + A3$ feature is used, the probability that desk is mistakenly classified as table is 7%. Error rate of classification decreases by 9% after $D1 + D2 + D3 + A3$ feature is fused.

From Figures 6(a) and 6(d), it can be seen that strong Bayesian classifier performs better than softmax classifier in

bathtub class and table class. Accuracy increases by 10% and 9%, respectively. At the same time, the problem of table being misclassified as desk has been solved to a certain extent. Error rate of classification decreases by 8%. However, strong Bayesian classifier performs poorly in dresser class. The probability that dresser is mistakenly classified as night_stand increases by 6%.

6. Conclusions

In this paper, voxel descriptor is used to express surface and internal information of 3D model. Shape distribution descriptor is adopted to denote geometry relationship of random points on model surface. 3D CNN is used to extract voxel feature, and 1D CNN is adopted to extract shape distribution features. Based on voxel feature and shape distribution feature, Bayesian classifier is used to classify 3D models. AdaBoost algorithm is adopted to combine several Bayesian classifiers to get a strong classifier for classifying 3D models. Experimental results show that accuracy of the proposed method is effective.

Data Availability

Princeton Shape Benchmark model library can be downloaded from <http://modelnet.cs.princeton.edu/>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61502124 and 60903082), Fundamental Research Foundation for Universities of Heilongjiang Province (Grant no. LGYC2018JC014), and Natural Science Foundation of Heilongjiang Province of China (Grant no. F2017014).

References

- [1] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, 2008.
- [2] X. Xu, S. Caulfield, J. Amaro, G. Falcao, and D. Moloney, "1.2 watt classification of 3D voxel based point-clouds using a CNN on a neural compute stick," *Neurocomputing*, vol. 393, pp. 165–174, 2020.
- [3] Z. Zhang, H. Lin, X. Zhao, R. Ji, and Y. Gao, "Inductive multi-hypergraph learning and its application on view-based 3D object classification," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5957–5968, 2018.
- [4] R. Guo, Y. Zhou, J. Zhao et al., "Point cloud classification by dynamic graph CNN with adaptive feature fusion," *IET Computer Vision*, vol. 15, no. 3, pp. 235–244, 2021.
- [5] W. Fu, X. Sun, Y. Ji, and Y. Bai, "Spectral-spatial hyperspectral image classification with dual spatial ensemble learning," *Remote Sensing Letters*, vol. 12, no. 12, pp. 1194–1206, 2021.
- [6] Y. B. Peng and Z. J. Zheng, "A two-stage sampling based ensemble learning method for hyperspectral image

- classification,” *Journal of Computational Information Systems*, vol. 11, no. 14, pp. 5135–5142, 2015.
- [7] T. Xia, P. Zhuo, L. Xiao, S. Du, D. Wang, and L. Xi, “Multi-stage fault diagnosis framework for rolling bearing based on OHF Elman AdaBoost-Bagging algorithm,” *Neurocomputing*, vol. 433, pp. 237–251, 2021.
- [8] R. Sirma, B. Dinar, and Y. H. Şahin, “A convolutional neural networks oriented approach for voxel-based 3D object classification,” in *Proceedings of the 26th IEEE Signal Processing and Communications Applications Conference*, pp. 1–4, Izmir, Turkey, May 2018.
- [9] Z. Kang, J. Yang, R. Zhong, Y. Wu, Z. Shi, and R. Lindenbergh, “Voxel-based extraction and classification of 3-D pole-like objects from mobile LiDAR point cloud data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4287–4298, 2018.
- [10] V. Plaza, J. A. Gomez-Ruiz, A. Mandow, and A. Garcia-Cerezo, “Multi-layer perceptrons for voxel-based classification of point clouds from natural environments,” in *Proceedings of the 13th International Work-Conference on Artificial Neural Networks*, pp. 250–261, Palma, Spain, June 2015.
- [11] A. Mughees and L. Tao, “Hyper-voxel based deep learning for hyperspectral image classification,” in *Proceedings of the 2017 IEEE International Conference on Image Processing*, pp. 840–844, Beijing, China, September 2017.
- [12] P. Babahajiani, L. Fan, J. Kamarainen, and M. Gabbouj, “Automated super-voxel based features classification of urban environments by integrating 3D point cloud and image content,” in *Proceedings of the 2015 IEEE International Conference on Signal and Image Processing Applications*, pp. 372–377, Kuala Lumpur, Malaysia, February 2016.
- [13] A. Ahmad, C. Paul, and T. Laurent, “Segmentation based classification of 3D urban point clouds: a super-voxel based approach with evaluation,” *Remote Sensing*, vol. 5, no. 4, pp. 1624–1650, 2013.
- [14] L. Agnello, A. Comelli, E. Ardizzone, and S. Vitabile, “Un-supervised tissue classification of brain MR images for voxel-based morphometry analysis,” *International Journal of Imaging Systems and Technology*, vol. 26, no. 2, pp. 136–150, 2016.
- [15] F. Chen, R. Ji, and L. Cao, “Multimodal learning for view-based 3D object classification,” *Neurocomputing*, vol. 195, pp. 23–29, 2016.
- [16] A.-A. Liu, F.-B. Guo, H.-Y. Zhou, W.-H. Li, and D. Song, “Semantic and context information fusion network for view-based 3D model classification and retrieval,” *IEEE Access*, vol. 8, pp. 155939–155950, 2020.
- [17] H. Kim, C. Yeo, M. Cha, and D. Mun, “A method of generating depth images for view-based shape retrieval of 3D CAD models from partial point clouds,” *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10859–10880, 2021.
- [18] Q. Huang, Y. X. Wang, and Z. Yin, “View-based weight network for 3D object recognition,” *Image and Vision Computing*, vol. 93, 2020.
- [19] D. A. Shajahan, V. Nayel, and R. Muthuganapathy, “Roof classification from 3-D LiDAR point clouds using multiview CNN with self-attention,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 8, pp. 1465–1469, 2020.
- [20] H.-Y. Zhou, A.-A. Liu, W.-Z. Nie, and J. Nie, “Multi-view saliency guided deep neural network for 3-D object retrieval and classification,” *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1496–1506, 2020.
- [21] Y. Zhou, F. Zeng, J. Qian, and X. Han, “3D shape classification and retrieval based on polar view,” *Information Sciences*, vol. 474, pp. 205–220, 2019.
- [22] Z.-X. Yang, L. Tang, K. Zhang, and P. K. Wong, “Multi-view CNN feature aggregation with ELM auto-encoder for 3D shape recognition,” *Cognitive Computation*, vol. 10, no. 6, pp. 908–921, 2018.
- [23] J. Lee, S. Cheon, and J. Yang, “Connectivity-based convolutional neural network for classifying point clouds,” *Pattern Recognition*, vol. 112, 2021.
- [24] M. Hu, H. L. Ye, and F. L. Cao, “Convolutional neural networks with hybrid weights for 3D point cloud classification,” *Applied Intelligence*, vol. 51, no. 5, pp. 1–14, 2021.
- [25] X. A. Li, L. Y. Wang, and J. Lu, “Multiscale receptive fields graph attention network for point cloud classification,” *Complexity*, vol. 2021, Article ID 8832081, 9 pages, 2021.
- [26] X. Zhang, C. Fu, Y. Zhao, and X. Xu, “Hybrid feature CNN model for point cloud classification and segmentation,” *IET Image Processing*, vol. 14, no. 16, pp. 4086–4091, 2020.
- [27] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “Pointcnn: convolution on x -transformed points,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 820–830, 2018.
- [28] Q. Huang, W. Wang, and U. Neumann, “Recurrent slice networks for 3D segmentation of point clouds,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2626–2635, Salt Lake City, UT, USA, June 2018.
- [29] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class adaboost,” *Statistics and Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.
- [30] X. Wang and K. Yan, “Gait classification through CNN-based ensemble learning,” *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1565–1581, 2021.
- [31] S. Wu and H. Nagahashi, “Parameterized adaboost: introducing a parameter to speed up the training of real adaboost,” *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 687–691, 2014.
- [32] W. Li, M. Luo, P. Zhang, and W. Huang, “A novel multi-feature joint learning ensemble framework for multi-label facial expression recognition,” *IEEE Access*, vol. 9, no. 1, pp. 119766–119777, 2021.
- [33] A. Taherkhani, G. Cosma, and T. M. McGinnity, “AdaBoost-CNN: an adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning,” *Neurocomputing*, vol. 404, pp. 351–366, 2020.
- [34] N. Kavitha, K. Ruba Soundar, and T. Sathis Kumar, “An improved DFA based kernel ensemble learning machine using local feature representations for face recognition,” *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 1, pp. 1203–1216, 2021.
- [35] X. Feng, “Research of sentiment analysis based on Adaboost algorithm,” in *Proceedings of the 2019 International Conference on Machine Learning, Big Data and Business Intelligence*, pp. 279–282, Taiyuan, China, November 2019.
- [36] P. Xiao, C. Li, L. Zhu, X. Zhang, T. Ma, and X. Feng, “Multitemporal ensemble learning for snow cover extraction from high-spatial-resolution images in mountain areas,” *International Journal of Remote Sensing*, vol. 41, no. 5, pp. 1668–1691, 2020.
- [37] H. Kaur and M. Kumar, “On the recognition of offline handwritten word using holistic approach and AdaBoost methodology,” *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 11155–11175, 2021.

- [38] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 6, pp. 1882–1897, 2019.
- [39] J. Lee, T. Kong, and K. Lee, "Ensemble patch sparse coding: a feature learning method for classification of images with ambiguous edges," *Expert Systems with Applications*, vol. 124, no. 1, pp. 1–12, 2019.
- [40] X. H. Zhang, F. X. Zhang, and P. Paul, "Dissolved gas analysis for transformer fault based on learning spiking neural P system with belief AdaBoost," *International Journal of Unconventional Computing*, vol. 16, no. 2-3, pp. 239–258, 2021.
- [41] S. A. Oyewole and O. O. Olugbara, "Product image classification using Eigen colour feature with ensemble machine learning," *Egyptian Informatics Journal*, vol. 19, no. 2, pp. 83–100, 2018.
- [42] Y. Song, S. Zhang, B. He et al., "Gaussian derivative models and ensemble extreme learning machine for texture image classification," *Neurocomputing*, vol. 277, no. 1, pp. 53–64, 2018.
- [43] H. Zeng, Q. Wang, C. Li, and W. Song, "Learning-based multiple pooling fusion in multi-view convolutional neural network for 3D model classification and retrieval," *Journal of Information Processing Systems*, vol. 15, no. 5, pp. 1179–1191, 2019.
- [44] H. Y. Li, Y. Q. Li, and Y. F. Zha, "Image retrieval method based on multi-view generating and ensemble learning," *International Journal of Performability Engineering*, vol. 13, no. 5, pp. 657–669, 2017.