International Journal of Science and Engineering Applications
Volume 15-Issue 07, 01 – 04, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1507.1001

Research on Standard Polyhedron Point Cloud Classification Combining K-means Clustering and Neural Network

Xiang Zhang
School of Electronic Information and
Electrical Engineering
Yangtze University
Jingzhou, China

Abstract:Addressing the issues of uniform point distribution, severe surface feature homogenization, and low fine-grained classification efficiency in standard polyhedron point cloud data, this paper proposes a fast polyhedron point cloud classification method that combines K-means clustering and neural networks.
First, taking the standard polyhedron point cloud dataset as the research object, we adopt the K-means clustering algorithm to preprocess unordered 3D point cloud data. The surfaces of the target polyhedrons are rapidly segmented based on the spatial coordinates and geometric distance features of the point cloud, which effectively eliminates redundant point cloud data and concentrates feature extraction on the core surface features of the polyhedrons. This method resolves the problems of redundant feature extraction and long computation time that plague traditional point cloud classification methods for regular geometric objects.
On this basis, the structured feature data obtained after K-means clustering is input into a neural network model for iterative training and feature learning. This process fully explores the surface topological features and spatial distribution rules of standard polyhedron point clouds, enabling accurate classification and recognition of polyhedron point clouds.
Experimental results show that the proposed hybrid method integrates the advantages of efficient preprocessing from traditional clustering algorithms and high-precision feature learning from neural networks. Compared with single-model neural network classification methods, it greatly improves the computation speed and recognition accuracy of surface classification for standard polyhedron point clouds. It can provide effective technical support for rapid analysis, 3D reconstruction, and geometric feature detection of regular 3D geometric point clouds.

Keywords: K-means clustering; neural network; standard polyhedron; point cloud classification; 3D feature extraction

1. INTRODUCTION

With the rapid development of 3D laser scanning and depth sensing technologies, 3D point clouds have become a mainstream data carrier for acquiring spatial geometric information, and are widely used in the fields of 3D reconstruction, industrial precision detection, and intelligent modeling. As basic regular geometric models, standard polyhedrons require accurate point cloud classification and surface segmentation to support subsequent geometric detection and lightweight model processing.

Current point cloud classification methods have obvious drawbacks when processing regular polyhedron point clouds. Traditional machine learning algorithms rely on manual feature engineering, which cannot effectively distinguish similar smooth surfaces and suffer from low classification accuracy. While deep learning neural networks can automatically extract deep point cloud features and achieve high classification precision, they have redundant network parameters and low computational efficiency when processing regular point cloud data with single, uniform features. As a result, they cannot meet the real-time application requirements of industrial scenarios.

To address the aforementioned problems, this paper proposes a hybrid classification framework that combines K-means clustering and a neural network. Specifically, K-means unsupervised clustering is used for coarse point cloud segmentation and redundant data removal to complete lightweight preprocessing, and a lightweight neural network is then applied for further high-precision feature extraction and refined classification. The proposed method achieves an excellent balance between classification speed and accuracy. Experimental results verify that it outperforms both the single traditional clustering method and the single neural network model.

2. RESEARCH METHODS

2.1 K-means Point Cloud Preprocessing Principle

K-means clustering is an unsupervised clustering algorithm based on distance measurement. Its core idea is to iteratively update cluster centers to minimize the spatial distance of point clouds within the same category and maximize the distance between different categories. For standard polyhedron point clouds, the points are evenly distributed with strong geometric regularity, which makes them suitable for regional division by Euclidean distance features. Let the 3D point cloud dataset be $P = \{P_1, P_2, \dots, P_n\}$, and each point contains 3D coordinate information (x,y,z). The algorithm randomly selects initial cluster centers, and assigns each point to a corresponding category by calculating the Euclidean distance between each

International Journal of Science and Engineering Applications
Volume 15-Issue 07, 01 – 04, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1507.1001

point and the cluster center. The calculation formula is as follows，through the equation(2-1).

$$d(p_i, c_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (2\text{-}1)$$

where C_j represents the j-th cluster center. During the iteration process, cluster centers are continuously updated until the clustering error converges. Finally, the rapid division of point clouds on different surfaces of polyhedrons is completed, realizing noise reduction, regional segmentation and standardized preprocessing of point clouds.

2.2 Construction of Neural Network Classification Model

After K-means preprocessing, the disordered and redundant information is removed from the point cloud data, and the regional features demonstrate higher consistency. In this paper, we adopt a lightweight neural network model to conduct in-depth training on the structured point cloud features obtained through clustering.

The model mainly consists of a feature input layer, hidden feature extraction layers, and a classification output layer. The network takes three features of each clustered region as inputs: point cloud density, spatial mean value, and surface normal vector. Through multi-layer nonlinear transformation, the model automatically learns differentiated geometric features on polyhedron surfaces, and ultimately achieves accurate classification of different standard polyhedrons and their different surface regions.

Compared with the training mode that directly inputs raw disordered point clouds, the data preprocessed by K-means effectively reduces the network's input dimension and the model's training parameters, and significantly improves the convergence speed and classification stability.

Figure 1 .Methodological framework of the study

2.3 Research Process

The overall workflow of the hybrid algorithm proposed in this paper is divided into three steps. First, the original standard polyhedron point cloud dataset is read and normalized to eliminate differences in coordinate scales. Second, the K-means clustering algorithm is applied to perform unsupervised clustering on the point clouds, so as to complete coarse classification of polyhedron surface areas and remove redundant points. Third, the clustered feature dataset is input into a neural network model for training, verification and testing, to realize refined classification and recognition of polyhedron point clouds.

3. EXPERIMENTAL SETUP

3.1 Dataset Introduction

In this experiment, a public standard polyhedron point cloud dataset is used. The dataset contains point cloud samples of four common regular polyhedrons: cube, cuboid, regular tetrahedron and regular octahedron. The point cloud density of each model is unified, and the number of points in a single model is controlled between 8000 and 12000. The point clouds are evenly distributed in space, retain the standard geometric features of edges, corners and surfaces, and are free from massive noise interference, which makes it suitable for verifying classification algorithms for regular geometric point clouds.

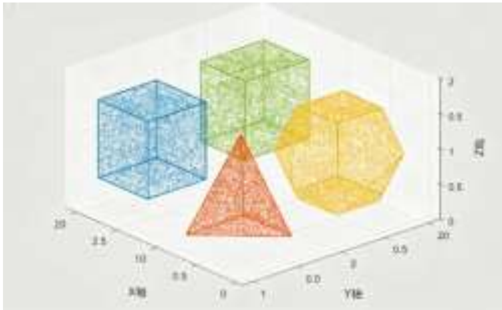


Figure 2 .Image of a standard cube point cloud dataset in a three-dimensional coordinate system

The dataset contains a total of 1200 samples, which are randomly divided into a training set and a test set at an 8:2 ratio. Specifically, 960 samples are allocated to the training set for model parameter fitting, and 240 samples to the test set for verifying classification performance.

During the division process, the number of samples for the four types of polyhedrons is evenly distributed to avoid the interference of imbalanced samples on the experimental results.

3.2 Experimental Environment and Parameter Settings

All experiments were implemented in the Python programming language, using the Open3D and PyTorch frameworks. For K-means clustering, the number of cluster centers is set to match the number of polyhedron surfaces, the maximum number of iterations is set to 300, and the convergence threshold is set to 0.001. The neural network uses an adaptive learning rate strategy, with an initial learning rate of 0.001, a batch size of 32, and a total of 100 training epochs. All parameters were kept consistent throughout all experiments to ensure the authenticity and validity of the comparison results.

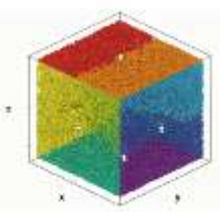
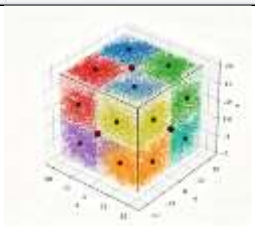
3.3 Training Process

The overall training process consists of two stages: K-means clustering preprocessing and iterative neural network training. First, K-means clustering is applied to the original unordered point cloud, using Euclidean distance as the classification criterion. Through multiple rounds of iteration, the cluster centers are optimized, causing scattered surface points to gradually converge toward their respective centers, thereby achieving automatic segmentation of polyhedral surfaces. The number of clusters *K* strictly matches the number of geometric surfaces, which ensures that each segmented region corresponds one-to-one with actual physical surfaces. In this experiment, the maximum number of iterations was set to 300, and iteration stops when the clustering error meets the convergence criteria.

After clustering, the unordered raw point cloud is divided into multiple locally distinct regions with independent characteristics, while redundant points and invalid noise are filtered out. Table 1 shows the visualization results of point cloud clustering during training on a standard cube, clearly revealing the spatial distribution patterns of the partitioned point clouds in a three-dimensional coordinate system.

When such structured feature data is fed into a lightweight neural network, the network learns based on the spatial distribution, density, normal vectors, and other features of each region. Compared to directly training with raw point clouds, this preprocessing method reduces redundant computations in the network, accelerates model convergence, and enhances the model's perception of polyhedral geometric features, effectively addressing the issues of low training efficiency and insufficient feature utilization in single neural networks.

Table 1 .Partial demonstration of the training process

Training Process 1	Training Process 2
	

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Evaluation Metrics and Classification Accuracy Comparison

To quantitatively evaluate the performance of different methods in standard polyhedron point cloud classification, a confusion matrix is used to count classification results, and the relevant indicators are defined as follows:

TP (True Positive): The number of samples that are actually positive and correctly predicted as positive.

FP (False Positive): The number of samples that are actually negative but incorrectly predicted as positive.

FN (False Negative): The number of samples that are actually positive but incorrectly predicted as negative.

TN (True Negative): The number of samples that are actually negative and correctly predicted as negative.

Based on the above statistical values,, the following evaluations (3-1)， （3-2）， （3-3）and（3-4）are selected:

$$ACC=\frac{TP+TN}{TP+TN+FP+FN} \qquad (3\text{-}1)$$

$$Precision=\frac{TP}{TP+FP} \qquad (3\text{-}2)$$

$$Recall=\frac{TP}{TP+FN} \qquad (3\text{-}3)$$

$$F1=\frac{2\times Precision\times Recall}{Precision+Recall} \qquad (3\text{-}4)$$

The above metrics are calculated respectively for each category (cube, cuboid, regular tetrahedron, regular octahedron), and the Macro-Average is taken as the final basis for performance comparison.

Table 2.Comparison of classification performance across different methods on standard polyhedral point cloud datasets

Method	ACC (%)	Precision (%)	Recall (%)	F1-score (%)
Traditional K-means Clustering	72.4	70.8	68.2	69.5
Single Neural Network	88.6	87.9	86.5	87.2
Proposed K-means + Neural Network	94.2	93.5	92.8	93.1

4.2 Efficiency Analysis

In terms of computational efficiency, K-means preprocessing simplifies the unordered features of the original point clouds and reduces the learning difficulty for the network, which greatly accelerates the model's convergence speed. Compared with neural network models trained directly on original point clouds, the inference speed of the proposed method is significantly improved under the same experimental environment. Additionally, it effectively solves the problem of redundant network learning caused by single features of regular polyhedron point clouds.

5. CONCLUSION

A hybrid classification framework combining K-means clustering and neural network is proposed to solve the problems of severe feature homogenization, low classification efficiency and limited recognition accuracy in standard polyhedron point cloud analysis. As a preprocessing module, K-means clustering segments point clouds by Euclidean distance based on geometric surface characteristics, which removes redundant points and invalid noise, and simplifies input features for subsequent models. On this basis, the lightweight neural network further learns the spatial distribution and topological features of point cloud data, and implements fine-grained classification of different types of polyhedrons.

Quantitative experimental results show that the proposed method achieves better performance than single K-means clustering and standalone neural network across all evaluation metrics. This framework avoids the tedious manual feature engineering required by traditional machine learning methods, and effectively reduces redundant computation of deep learning models. The research results can provide a feasible technical solution for rapid analysis, detection and lightweight processing of regular 3D geometric point clouds in engineering applications.

In future work, we will focus on developing an automatic K-value selection algorithm to eliminate the dependence on prior geometric knowledge of polyhedrons. Furthermore, we will integrate multi-dimensional geometric features including surface curvature and normal vectors to improve segmentation quality. We also plan to test the algorithm under complex noisy environments and mixed polyhedron scenarios, so as to enhance its robustness and practical generalization ability for industrial field applications.

6. REFERENCES

[1] Wang X L, Chen G, Zhou H R. Research on 3D Point Cloud Classification Method Based on Deep Learning[J]. IJRES, 2023, 11(4): 426-431.

[2] Weinmann M, Jutzi B, Hinz S, et al. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2015, 105: 286-304.

[3] Zhang L, Li Y, Wang H. 3D Point Cloud Segmentation and Classification Method Fusing Multi-scale Features[J]. Laser & Optoelectronics Progress, 2022, 59(18): 181002.

[4] Qi C R, Su H, Mo K, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 652-660.

[5] Liu X, Chen M, Zhao Y. Research on Lightweight 3D Point Cloud Classification Neural Network[J]. Computer Engineering and Applications, 2024, 60(8): 192-199.

[6] Li J, Wu F. Fast 3D Point Cloud Segmentation Algorithm Based on K-means Clustering[J]. Application Research of Computers, 2021, 38(S1): 156-158.