Journal of Network Intelligence ©2020 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **5**, Number **3**, August **2020**

Design of Neural Network Model for Lightweight 3D Point Cloud Classification

Chenxia Wang[1], Taibin Shi[2], LinLin Tang[3] and Yeh-Cheng Chen[4]

[1]School of Biomedical Engineering, Xinxiang Medical University
[2]Zhengzhou Preschool Education College
[3]Harbin Institute of Technology, Shenzhen
[4]Department of Computer Science, University of California, Davis, CA, USA
[1]stb2873314@163.com, [2]stb2873314@aliyun.com [3]hittang@126.com, [4]ycch@ucdavis.edu

Received May 2020; revised July 2020

Abstract. *In this paper, a lightweight point cloud classification model, LightSeDgCNN, is proposed to solve the problem of large number of basic parameters and long training time. Based on the latest DGCNN model, the structure of the model is optimized through experiments. Secondly, the influence of model classification power reduction caused by the reduction of lifting parameters is improved by introducing the SENeT model (which can automatically acquire the importance of each feature channel through learning). Finally, the experiment is carried out on the ModelNet40 data set. The experimental results show that LightSeDgCNN can achieve the same effect of point cloud classification and recognition with less model parameters and training time.*
Keywords: 3D Point Cloud Classification, Lightweight, DGCNN, Neural Network

1. **Introduction.** With the continuous development of 3D scanning technology and the development of deep learning, classification [1] and segmentation of point cloud data has become a research hotspot of scholars and researchers. However, due to the influence of 3D scanning equipment on objects in different positions and coordinate system, the point cloud data scanned by 3D scanning equipment is quite different, which brings great challenges to research of point cloud data processing problems for the point cloud data only containing 3D coordinate information (less information). With the emergence of PointNet [2] model, the research work of disordered point cloud has made a pioneering progress. For pointnet point cloud classification, only the global features of point cloud are considered. In recent years, researchers focus on new point cloud features combining global features and local features in order to get better results. Some researchers propose to consider distance relationship between point clouds, the distribution mode of point clouds, the direction between points and other ways are considered. Based on above analysis and consideration, a kind of network model structure that can run on small equipment or equipment with high real-time requirements is more suitable for practical application.

On the other hand, with the continuous development of deep learning technology, the research of 3D point cloud data has gradually shifted from the low-level manual feature extraction to the high-level semantic understanding (such as point cloud shape recognition, point cloud scene segmentation). Most of the manual feature extraction for point cloud is based on specific point cloud data processing tasks. And the manual feature of point cloud is based on the specific statistical features of the coding point cloud, which can be generally divided into intrinsic and non-intrinsic attributes [3]. Of course, it can also be divided into global and local features. At present, the deep learning of 3D point cloud data is mainly divided into the following three parts.

1. *Multiple Perspectives Based Model:* Su H[4] et al. obtained images of the object from different perspectives through 3D data of the object, took these two-dimensional images as training data and then trained the results in convolutional neural network through View-Pooling operation. They achieved some results. But it cannot be applied onto the 3D point cloud model classification and segmentation.
2. *Voxel Based Model:* Maturana D [5] et al. proposed to grid 3D point cloud data and then used convolution neural network to achieve good results. Zhi S et al. proposed a lightweight point cloud data model, which has less parameters but does not significantly reduce the model, and it is a lightweight point cloud processing model based on VoxNet.
3. *Point Cloud Based Model:* For the deep learning[16] neural network which is processed directly on the point cloud, Qi [2], a scholar from Stanford University, etc., proposed PointNet neural network, which can be seen as the pioneer of the point cloud deep learning neural network, which is the feature extraction directly on the point cloud data. It mainly uses a symmetric function to solve the disorder problem of point cloud, but it is a point by point process that cannot deal with the local characteristics of point cloud. Later. Based on the PointNet, Qi [6] and others put forward the PointNet + + neural network. PointNet + + obtains the local characteristics of each region sampled by the farthest point sampling algorithm and the ball query algorithm. It obtains better classification recognition and scene segmentation accuracy.

Chen LZ [7] and others designed a novel local space perception (LSA) layer based on the spatial distribution of point clouds. This layer can learn to generate spatial distribution weights based on the spatial relations of local areas, so as to carry out spatial independent operations, so as to establish the relationship between these operations and spatial distribution, and better aggregate the spatial information and its related characteristics of each layer of the network. This kind of neural network LSANet has achieved outstanding results.

Wang Y [8] et al. proposed a dynamic graph convolution neural network. The main idea is to extract local features of PointNet and PointNet + +, which do not use the relationship between points. By integrating the global features and local features, the convolution neural network is used to complete the task of point cloud classification and segmentation. The experimental results show that the performance is better than PointNet and PointNet + +.

Li [9] of Shandong University and others proposed PointCNN. This model mainly introduces a transformation method called X-Conv to complete regularization of unordered point cloud, so as to use convolution neural network [17] to effectively classify the input point cloud for the regularized point data. And it achieves good classification results.

For the problem of point cloud shape classification, we designs a lightweight neural network with less parameters and less computation. The model can run on miniaturized equipment or equipment with high real-time requirements, and the effectiveness can be comparable to the existing large-scale network. The section 2 will introduce some related work and our proposed algorithm will be given in section 3. Some experiments will be shown in section 4 and the conclusion will be in section 5.

2. **Related Work.** Point cloud input data needed in this paper is only the point containing 3D coordinate information:

$$\{p_i = \{x_i, y_i, z_i\} \, | i = 1, 2, ..., n\} \tag{1}$$

Here, p_i represents a point in the point cloud data, which contains three-dimensional coordinate information $\{x_i, y_i, z_i\}$, i represents the ith point in the input point cloud, and n represents the number of families of the input point cloud.

In this paper, we mainly concern the study of point cloud shape classification [15]. Therefore, we mainly study the design of a classification method f to transform the input point cloud into the probability distribution corresponding to each shape distribution p :

$$p = f(\{p_1, p_2, ..., p_n\}) \tag{2}$$

Firstly, the model in this paper is based on the model of DGCNN [8]. Some ideas are from the model designed in lightweight point cloud classification model combined with the latest ideas of ShuffleNetV2 model proposed by Ma N [10].

In the DGCNN [8] model, a T-Net neural network is used to calibrate the point cloud to ensure the transformation invariance (rotation and translation transformation). However, the addition of T-Net makes the amount of parameter data of whole model nearly double. For this, we can add appropriate rotation and translation transformation to the training data. After experimental analysis, the classification accuracy of the model is not significantly reduced after removing T-Net. The underlined data represents the model obtained in the experimental environment of this paper, and is given by the underlined way.

Table 1. Time Efficiency of Matching

Model	Accuracy	Number of parameters (10^6)
DGCNN[7]	<u>91.56</u>	<u>1.84</u>
DGCNN(no T-Net)	91.23	1.03

Through the above experiments, it can be shown that after removing T-Net, parameters of the model are reduced by nearly half. The accuracy has not decreased a lot, which provides a good basis for further design. Part of the reason can be obtained from literature [11]. Here we have a simple analysis and discussion. Because the output of T-Net neural network is a correction matrix R of 3×3 , the correction matrix is used to correct the input point transport data:

$$P_{offset} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3}$$

In the above formula, r_{ij} is the element in row i and column j of the 3×3 correction matrix R obtained from the T-Net network, in addition to the 3D coordinates of the ith point p_{ij} of the input point cloud data are x_i, y_i, z_i.

If we transform the process of MLP updating point cloud features into matrix form, we can find that the process of updating hidden MLP feature matrix is as follows.

$$h(p) = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1C'} \\ w_{21} & w_{22} & \cdots & w_{2C'} \\ w_{31} & w_{32} & \cdots & w_{3C'} \end{bmatrix} + [b_1 \;\; b_2 \;\; \ldots \;\; b_{C'}] \tag{4}$$

In the formula, $h(p)$ is the feature processing function of MLP in a certain layer. $w_{iC'}$ and $b_{C'}$ are the parameters can be trained, represent the number of output channels.

Compared with the above two formulas, we can find the expressions of the two formulas are very similar. Their main difference is that correction matrix obtained through T-NET is a special correction matrix for each point cloud, and the MLP process can be seen as a static transformation for all point clouds.

3. **Our Proposed Method.** In addition to T-Net increasing the number of parameters of the whole model, the parameters of the whole network are basically concentrated in the full connection layer. We hope to change the number of full connection layers and the number of nodes in the full connection layer. In this paper, we will consider and explore the impact of the number of nodes in the full connection layer on the classification effect. Here is a study based on the model shown in the following figure.

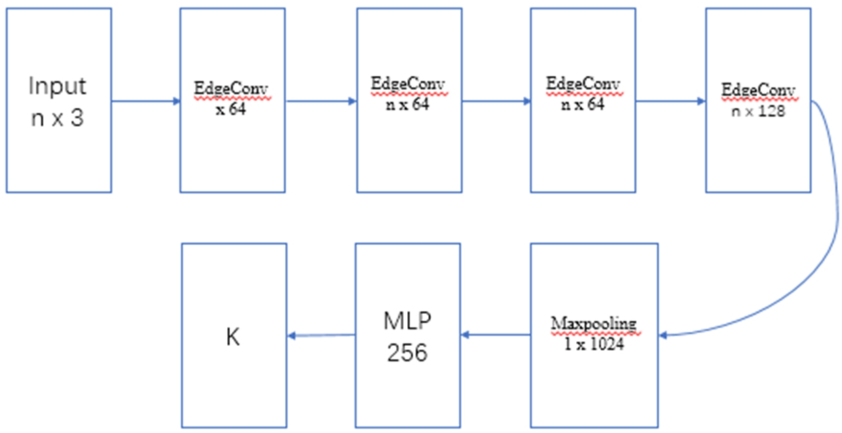

Figure 1. Basic Model

Because there is only one full connection layer at this time, we change the number of nodes in the full connection layer in the figure, and take the number of full connection layers as 128256512 and 1024 for training, which is verified on the test set. The experimental results are as follows:

Table 2. Matching Time Efficiency

Number of nodes	Classification accuracy
1024	88.31
512	89.19
256	89.14
128	87.52

At this time, after the number of nodes of full connection has been determined to be 256, a layer of full connection layer 128 will be connected. Through experiments, the impact of number of layers of full connection on effect will be analyzed. The experimental results are shown in the following table 3.

The experimental results in the above table show that although a full connection layer is added However, it does not improve the performance much, but reduces the classification effect. The possible reason is that the point cloud data only contains three-dimensional coordinates, which contains too little information. Therefore, the later design still uses one layer of full connection, and the full connection parameter nodes are set to 256.

Table 3. Matching Time Efficiency

Number of layers	Classification accuracy
256	89.14
256+128	89.08

As shown in the figure below, the lightweight point cloud classification network model designed in this paper is shown in the figure below. We call this lightweight point cloud classification network LightSeDgCNN.

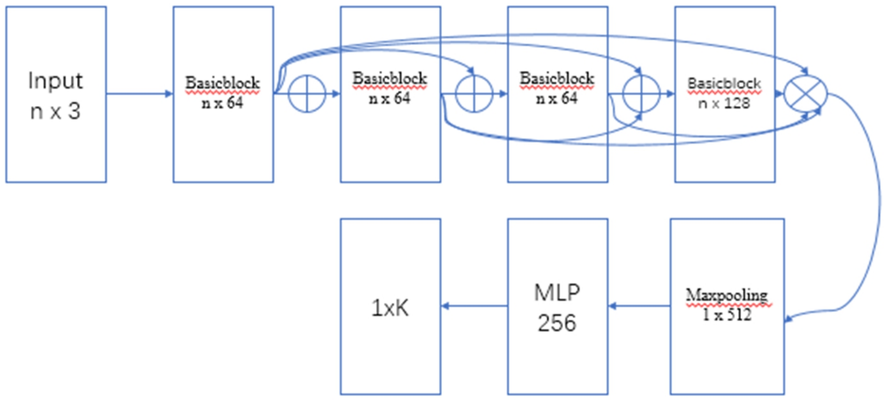

Figure 2. Model Diagram

As shown above, our model LightSedGCNN consists of six layers (excluding the input layer and output layer): the point cloud data input by the input layer is composed of n points, each point is composed of three-dimensional coordinates (x, y, z), which is a $n{\times}3$ tensor. It also contains four basicblock modules: the first basicblock is connected with the other two modules through residual, and the second is connected with the third through residual. And finally, the output results of the 4 basicblock modules are concatenated. The pooling layer pools the results of the previous concat to get the tensor of $1{\times}1024$. Finally, results of the pooling layer are further learned through a full connection layer to complete the whole model.

In this paper, ModelNet40 of Princeton ModelNet, which is the same as PointNet is selected as benchmark to complete the model test, and experimental results are compared with other methods. There are total 12311 models in 40 shapes in ModelNet40. The experimental data are obtained under the experimental environment of this paper, and they are given in the following table by underlining.

Table 4. Experimental Results of ModelNet40

Input type	Methods	Classification Accuracy	Parameter Quantityx10^6
Muti-view	MVCNN[4]	90.10	138.00
Volumetric	VoxNet[5]	83.00	0.92
Volumetric	3DShapeNets[13]	77.32	38.00
Volumetric	LightNet[14]	86.9	∼0.30
P_C	PointNet[12]	89.20	3.50
P_C	PointNet++[6]	90.70	1.47
P_C	DGCNN[8]	<u>91.56</u>	<u>1.84</u>
P_C	LightSeDgCNN	**89.81**	**0.34**

As shown in the above table, the classification results of various methods indicated in lines 2 to 8 of the table are on modelnet40. The classification accuracy of LightSedGCNN on ModelNet40 dataset is 89.81%, and the parameter quantity is 0.34×10^6.

The following table is the experiment of several latest typical point cloud neural network models on the ModelNet40. The experimental results are given in the following table 4.

TABLE 5. Time Performance Analysis

Method	Forward time(ms)	Training Time
PointNet[12]	0.80	~3.5h
PointNet++[6]	1.40	-
DGCNN[8]	3.10	~4.5h
LightSeDgCNN	**0.71**	**~2.3h**

In terms of the forward propagation time of the model, the LightSeDgCNN proposed by us is only 22.9% of DgCNN, 50.7% of PointNet++, 88.75% of PointNet, which has considerable performance advantages. Moreover, in terms of the training time of the model, LightSeDgCNN is about 2.3 hours, PointNet is about 3.5 hours, and DGCNN is about 4.5 hours, it obviously also has certain advantages in the training time.

4. **Conclusions.** LightSeDGCNN proposed in this paper has achieved good results in point cloud model classification, and is superior to the methods mentioned in the paper in model parameters, forward propagation time and training time. It is believed that this work will have a very wide range of scene applications in unmanned driving, virtual reality or other miniaturized equipment and other scenes with high real-time requirements for point cloud processing.

Acknowledgment. This work was supported by Shenzhen Science and Technology Plan Fundamental Research Funding JCYJ20180306171938767 and Shenzhen Foundational Research Funding JCYJ20180507183527919.

REFERENCES

[1] Xiaosong Zhao, Rui Zhang, Jheng-Long Wu and Pei-Chann Chang A Deep Recurrent Neural Network for Air Quality Classification, *Journal of Information Hiding and Multimedia Signal Processing*, 9(2): 346-354, March 2018.

[2] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652-660, 2017.

[3] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration, *2009 IEEE International Conference on Robotics and Automation. IEEE*, pp. 3212-3217, 2009.

[4] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3d shape recognition, *Proceedings of the IEEE international conference on computer vision*, pp. 945-953, 2015.

[5] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition, *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE*, pp. 922-928, 2015.

[6] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Advances in neural information processing systems*, pp. 5099-5108, 2017.

[7] Chen L Z, Li X Y, Fan D P, et al. LSANet: Feature Learning on Point Sets by Local Spatial Attention[J]. arXiv preprint arXiv:1905.05442, 2019.

[8] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. *ACM Transactions on Graphics (TOG)*, 38(5): 146, 2019.

[9] Li Y, Bu R, Sun M, et al. Pointcnn: Convolution on x-transformed points, *Advances in Neural Information Processing Systems*, pp. 820-830, 2018.

[10] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116-131, 2018.

[11] Zhang K, Hao M, Wang J, et al. Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features[J]. arXiv preprint arXiv:1904.10014, 2019.

[12] Chen L Z, Li X Y, Fan D P, et al. LSANet: Feature Learning on Point Sets by Local Spatial Attention[J]. arXiv preprint arXiv:1905.05442, 2019.

[13] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912-1920, 2015.

[14] Zhi S, Liu Y, Li X, et al. Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning[J]. *Computers & Graphics*, 71: 199-207. 3, 2018.

[15] Junli Zhao, Zhenkuan Pan, Fuqing Duan, Zhihan Lv, Jinhua Li, Qiang Zhou, Xiangang Shang, Jie Sun and Pingping Wang, A Survey on 3D Face Recognition based on Geodesics, *Journal of Information Hiding and Multimedia Signal Processing*, 10(2): 368-383, March 2019.

[16] EK Wang, CM Chen, MM Hassan, A Almogren, A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain, *Future Generation Computer Systems 108*, 135-144, 2020.

[17] Eric Ke Wang, Ship Peng Xu, Chien-Ming Chen and Neeraj Kumar, Neural-architecture-search-based multiobjective cognitive automation system, *IEEE Systems Journal*, 2020.