밀집한 신경망 그래프 기반점운의 분류

아메드 엘 카자리*, 이효종*
*전북대학교 컴퓨터공학과
E-mail : hlee@jbnu.ac.kr

Dense Neural Network Graph-based Point Cloud classification

Ahmed El Khazari[1], Hyo Jong lee[1]
[1]Dept. of Computer Science and Engineering, Chonbuk National University

Abstract

Point cloud is a flexible set of points that can provide a scalable geometric representation which can be applied in different computer graphic task. We propose a method based on EdgeConv and densely connected layers to aggregate the features for better classification. Our proposed approach shows significant performance improvement compared to the state-of-the-art deep neural network-based approaches.

1. Introduction

Point clouds are collections of points in 2D or 3D space. Point clouds can represent the objects well regardless the environment, they can exhibit less variation and can be recognized under strong lighting changes.

Several tasks can be done on the point clouds, i.e. classification, segmentation and registration. We consider the classification task. Traditional methods for solving such a task rely on handcrafted features to induce to obtain the geometric properties of the point clouds [1,2]. However, the huge success of the deep learning of 2D images has leaded to learn features on point clouds, ending up outperforming the traditional approaches.

The state-of-the-art deep neural networks are considering each point individually without taking into accounts its neighbors, this approach was pioneered by PointNet [3]. Various version of PointNet maintain neighbors of each points to have more understanding on the 3D model. However, these approaches failed to contain a geometric relationships among the points.

To address this problem, we come up with a simple approach based on EdgeConv [4], which generates features based on the edges that describes relationships between a point and its neighbors. Our key contribution resides on combining between the EdgeConv and skip connections inspired by the DenseNets [5]. The resulting network architecture achieves good performance on ModelNet40 datasets [6].

2. Related work

Point Net and its extensions is the first deep neural network based to manipulate raw 3D point clouds. They treat each point independently to learn a mapping from 3D to latent features, without considering local geometric structure. Whereas, PointNet++ [7] achieves the state-of-the-art results by leveraging local point set without taking into account relationships between the point sets.

However, DGCNN [4] come into existence to reply on the relationships among all the point by creating a dynamic graph by k-neighbors to learn the geometric structure as well as considering all the relationships that can occur between the points.

3. Proposed Method

We follow the same approach of DGCNN, yet we rely on the idea of the DenseNets, which is basically simple connectivity pattern among all the layers to ensure the maximum information flow and easily integrate the properties of identity mappings as shown in Figure 1.

After each EdgeConv layer the *i*-th vertex' output can be given by

$$x_i' = \sum_{j\epsilon N} h_\theta\ (x_i || x_j - x_i),$$

where h_θ denotes multi-layer perceptron (MLP).

This yields to the convolution operation.

The l^{th} layer gets the features of all proceeding layers, $x_0, x_1, \dots, x_{l-1}$ as input:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])\ ,$$

where $[x_0, x_1, \dots, x_{l-1}]$ refers to the concatenation of the features produced in EdgeConv layers. As shown in Figure 1, EdgeConv layer consists of k-nn graph alongside MLP layers.

The network architecture used for the classification task is illustrated in Figure 1.

4. Experiment

We conducted our experiment on the ModelNet40 classification task. The dataset contains 12,311 CAD models from 40 categories, 9,843 models are used for training and

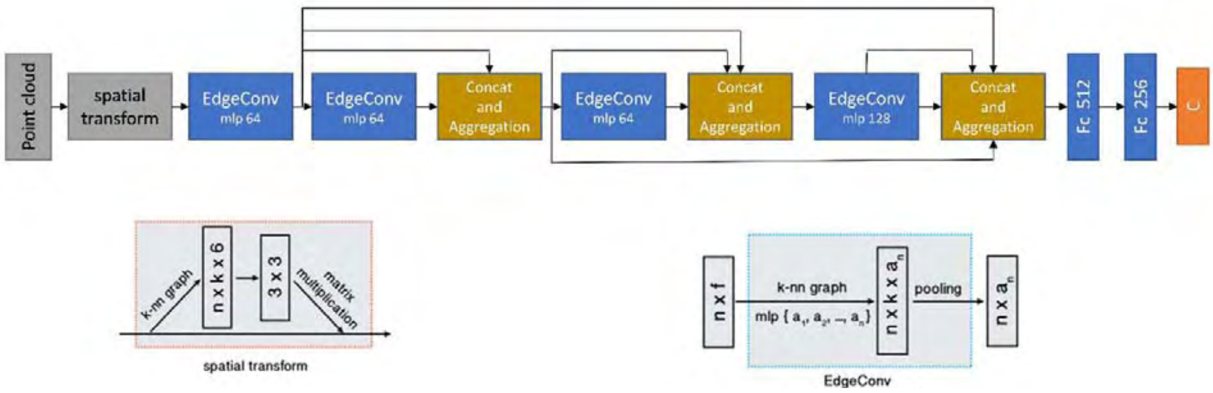

Figure 1. **Model architecture:** takes as input a point cloud and outputs the classification score by obtaining the features using EdgeConv and concatenating and aggregating them. **Point cloud transform block:** is meant to align the point cloud by assessing 3x3 matrix. **EdgeConv block**: takes as input a tensor and computes features by applying mlp then pooling among neighboring edge features.

2,468 models are for testing. The dataset was sampled into 1,024 points.

The input point cloud passes through the spatial transformer network in order to align it. Then followed by 4 EdgeConv layers. The first layer consists of 64 fully connected layer, as well as the second and third layer, whereas, the last layer consists of 128 fully connected layer. The skip connections are also included from every previous layer to keep the information flow and aggregate globally to form 1D descriptor to the next layer. The number of k is 20 in this experiment. Then, and max pooling is used to obtain the features, after that 2 fully connected layers of 512 and 256 respectively are used to transform the features to classify.

5. Results

In Table 1, we show the results of the classification task. Our model achieves some competitive results compared to the state-of-the-art.

Table1. Classification results on ModelNet40

	Mean class Accuracy	Overall Accuracy
3DShapeNets [6]	77.3	84.7
VoxNet [8]	83.0	85.9
SubVolume [9]	86.0	89.2
ECC [10]	83.2	87.4
PointNet [3]	86.0	89.2
PointNet++ [7]	-	90.7
KD-Net (Depth15) [11]	-	91.8
DGCNN (baseline) [4]	88.8	91.2
Ours	**89.3**	**91.7**

6. Implementation Details

We train using a batch size of 32, and use the Adam optimizer with leaning rate of 1e-3. We find that more epochs are needed to obtain good performance (approximately 400 epochs). We implement the network architecture in Tensorflow and train using Nvidia GeForce GTX Titan X.

7. Conclusion

In this paper we explored the effectiveness of using dense connections that leaded to better accuracy. We showed also the combination of the dense graph-based approach could achieve a better classification in the benchmark dataset.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (GR 2016R1D1A3B03931911).

References

[1] Rusu, Radu Bogdan, Nico Blodow, and Michael Beetz. "Fast point feature histograms (FPFH) for 3D registration." 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009.

[2] Rusu, Radu Bogdan, et al. "Aligning point cloud views using persistent feature histograms." 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008.

[3] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[4] Wang, Yue, et al. "Dynamic graph cnn for learning on point clouds." arXiv preprint arXiv:1801.07829 (2018).

[5] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[6] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proc. CVPR, 2015.

[7] Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in Neural Information Processing Systems. 2017.

[8] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proc. IROS, 2015.

[9] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proc. CVPR, 2016.

[10] M. Simonovsky and N. Komodakis. Dynamic edgeconditioned filters in convolutional neural networks on graphs. In Proc. CVPR, 2017.

[11] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. 2017.