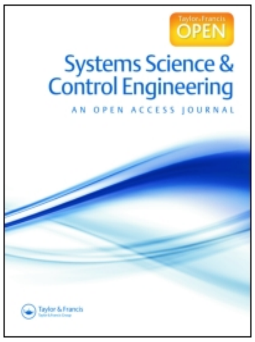

Taylor & Francis
Taylor & Francis Group

Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tssc20

An improved volumetric grid deep network model for point cloud segmentation

Xinliang Zhang , Chenlin Fu & Yunji Zhao

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL
https://doi.org/10.1080/21642583.2020.1826004

Taylor & Francis
Taylor & Francis Group

An improved volumetric grid deep network model for point cloud segmentation

Xinliang Zhang, Chenlin Fu and Yunji Zhao

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, People's Republic of China

ABSTRACT

Voxel grid is widely used in point cloud segmentation due to its regularity. However, the memory consumption caused by high resolution restricts the performance of voxel grid. This paper proposes an improved voxel grid deep network (IVDN) model to represent more comprehensive point cloud features at the same resolution, thus improving the segmentation performance of point cloud. Firstly, the point cloud data are structured within a voxel bounding box to correspond with the three-dimensional(3D) convolution kernel, and a fixed number of point coordinates are selected to generate the point feature vector. Then, in order to consider the distribution characteristics, the reliability coefficient is used as an equivalent descriptor of the point cloud distribution density. Finally, a corresponding deep network is constructed to deal with the above features. Experimental results show that the proposed IVDN model can improve the mean classification accuracy and segmentation index mIoU(Mean Intersection over Union) effectively, with a 0.45% and 0.3% improvement on Shape net16 dataset respectively.

ARTICLE HISTORY
Received 1 July 2020
Accepted 16 September 2020

KEYWORDS
3D point cloud; voxel grid; reliability coefficient; deep network; point cloud segmentation

1. Introduction

Convolutional Neural Network is a deep learning model framework based on multi-layer neural network for image classification and recognition. In recent years, a lot of in-depth research has done in the area of image classification, segmentation, detection, etc. (Chen et al., 2018; Krizhevsky et al., 2017; Zeng et al., 2016; Zeng et al., 2018; Zeng et al., 2019; Zeng et al., 2020). Meanwhile, the spatial three-dimensional(3D) point cloud data which obtained by 3D measurement techniques such as lidar/depth camera have been widely used in 3D reconstruction (Mescheder et al., 2019), unmanned vehicle, real-time positioning and mapping (SLAM) and other areas. The application of convolutional neural network to the classification and segmentation of 3D point clouds has a broad prospect (Yang et al., 2018).

Compared with two-dimensional(2D) images, 3D point cloud data has the advantages of efficient data processing, flexible structure and rich information description. However, because of the unstructured nature of point cloud (Thomas et al., 2019), deep learning method cannot be directly applied to point cloud, it is particularly critical to solve the unstructured problem of point cloud. Charles R (Charles et al., 2017) first obtained a transformation matrix by using neural network, then used the matrix to filter the point cloud to have structural characteristics for the usage of the deep learning operator. However, the learning process of this transformation matrix is much complicated and lacks generality. References (Maturana & Scherer, 2015; Zhou & Tuzel, 2018) put the point cloud in a fixed voxel frame by analogy with the matrix arrangement of pixel points in a two-dimensional image. In this way, the spatial 3D point is uniquely determined by the voxel index within the frame. In literature (Huang & You, 2016), the tagging of scenic spot clouds in real fields has been realized by the combination of the voxelized point clouds and 3D convolution operator. And it gives a good performance in large-scale point cloud segmentation. However, for the high-resolution point cloud data, its long processing time and the heavy memory consumption are considerable. PointGrid (Le & Duan, 2018) selects the coordinates of a fixed number of voxel internal points as the point feature to compensate the subtle information. This method can balance the resolution and memory consumption to some extent, but the fixed number of points lacks the ability to describe distribution characteristics. This paper uses voxel method to structure point cloud and describes local details of point cloud by selecting point coordinates in voxel. A reliability coefficient is constructed to highlight the difference of voxel components while describing the distribution density. The improved voxel grid deep network (IVDN) model can describe more complete/comprehensive point cloud features at a given resolution.

CONTACT Xinliang Zhang ✉ zxldq@hpu.edu.cn

Our contributions are summerized as follows.

(1) The shortcomings of voxelized point cloud method in describing point cloud features were analysed.
(2) On the basis of selecting a fixed number of point coordinates to describe the detail information of point cloud, the reliability coefficient is constructed to describe the distribution information.
(3) An IVDN network model for point cloud segmentation is proposed.

The remainder of this paper is organized as follows. Section 2 summarizes the relevant literature and highlights the relevant issues. Section 3 details the input layer of the IVDN model and the reliability coefficient. Section 4 introduces the network structure of the model. Section 5 discusses the evaluation results of the model, and compares them with the existing network structure. Section 6 presents the conclusions and discusses possibilities for future work.

2. Problem statement

Point cloud data describes three-dimensional objects in the form of spatial coordinate sets. For point cloud data containing, $B(x_b, y_b, z_b)$ and $C(x_c, y_c, z_c)$, it can be described as the following matrix.

$$M = \begin{bmatrix} x_a & y_a & z_a \\ x_b & y_b & z_b \\ x_c & y_c & z_c \end{bmatrix} \quad (1)$$

The elementary row transformation of matrix M corresponds to different point cloud storage order. According to the elementary transformation property of the matrix, the transformed matrix is equivalent to the original matrix M. However, for the convolution operator, the same input corresponds to different output values. Therefore, point cloud data needs to be structured so that each point corresponds to its features, indexes, and coordinates.

For structuring point cloud methods, VoxNet (Maturana & Scherer, 2015) and its variants (Brock et al., 2016; Li et al., 2016; Mutz et al., 2016; Qi et al., 2016; Wang et al., 2017; Wang & Posner, 2015; Xiao et al., 2017) are the most direct way to convert 3D models into occupied voxel grids. Although this approach addresses the unstructured problem of point clouds, the main disadvantages of this method are the loss of too much 3D information. Figure 1 shows the descriptive performance of voxelized point cloud to the target. It can be seen from Figure 1 that the delicacy of voxel grid's description of the target depends on the resolution of the grid. In order to obtain more 3D information, the most intuitive method is to improve the resolution of the grid. But for three-dimensional point clouds, a small increase in resolution creates an exponentially growing amount of data. In addition, there must be a large number of empty voxels in regular voxel frames, which consume a large amount of unnecessary computing power.

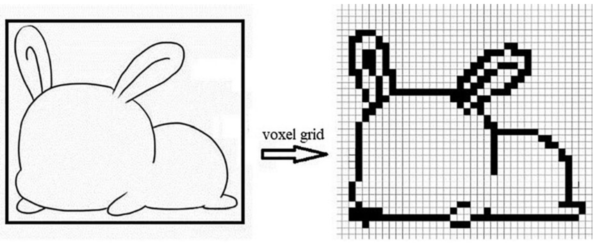

Figure 1. Voxelization description of 3D point cloud target.

In this paper, a fixed number of points in voxels are selected to describe the target details more accurately at the same resolution. In addition, the reliability coefficient highlighting voxel difference is added to the point feature vector to increase the descriptive ability of the target. The section III – input layer will introduce the pretreatment method in detail.

3. Input layer of IVDN

3.1. Generation of the point feature vector

Using voxel grid to structure point cloud data, the mapping relationship of 'point – voxel – index' is established. The index V_i to which point $I(x_i, y_i, z_i)$ belongs to the corresponding voxel can be described by the following formula.

$$V_i = (N \times x_i) \times N^2 + (N \times y_i) \times N + (N \times z_i) \quad (2)$$

In $A(x_a, y_a, z_a)$ the formula (2), $V_i \in [0, N^3 - 1]$, N-N-N are the voxel resolutions along the X-Y-Z axis respectively. The points of the same index are allocated to the same voxel unit, and the number of points within the voxel can be obtained by counting the number of the same index points.

On the basis of voxelazation point cloud, the point coordinates within voxel are selected to be extended into feature vectors to better reflect the target contour details. Figure 2 shows the descriptive performance of the point information. Using placeholder filling method, the number of points in the voxel selected by the design is K. If the number of points contained in voxel is greater than or equal to K, K points are randomly selected. If the number of points in the precursor element is less than K, the insufficient points are filled with $(0,0,0)$. If the voxel does not contain any true points, K zeros $(0,0,0)$ are used to fill it. This method selects K points whose coordinates span vector $K \times 3$ to represent each voxel.

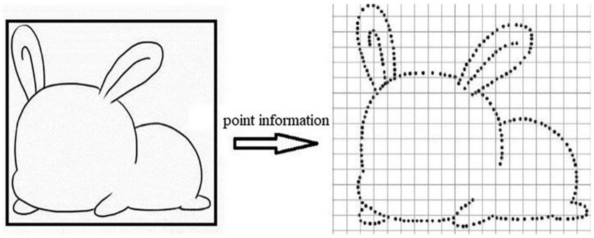

Figure 2. Point information description of the point cloud target.

The method above selects a fixed number of points, but each voxel is different. In order to highlight the differences between voxels and enhance the descriptive ability of the model, the corresponding quantization parameters are constructed in this paper.

3.2. The reliability coefficient

Define the reliability coefficient of voxel data:

$$C = P/K \tag{3}$$

where P is the number of real points in voxel, and K is the number of selected points in voxel.

When $P = 0$, the reliability coefficient is 0. That is, the K points used in the design are filled with placeholders instead of real points.

When $0 < P < K$, then $0 < C < 1$. And the *k-p* points are placeholders.

When $P \geq K$, the credibility coefficient $C \geq 1$ and K points are all true and credible points. The reliability coefficient C represents how dense the number of true points P in voxel is relative to the number K selected.

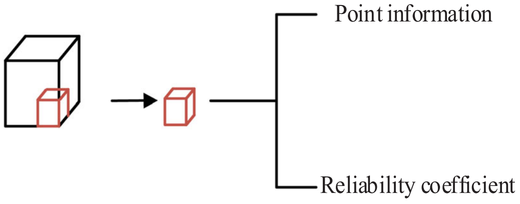

Figure 3. The feature composition of the input layer.

At the same resolution, the voxel detail information varies linearly with the selected points. The reliability coefficient is not only a measurement index of data reliability, but also defined by it. After the parameter is linearly amplified by K times, the distribution density information of point cloud can be just represented. In other words, the higher the density, the higher the reliability. Therefore, the amplified confidence parameter is used as the density to introduce the feature vector of voxel to make the voxel descriptors into the network more complete. Figure 3 shows the feature composition of the input layer.

Deep network has great advantages in feature extraction. The purpose of structured point cloud is to make point cloud data suitable for deep network. The main work of this paper is to improve the ability of feature descriptors to describe local details on the basis of structured point cloud and construct the credibility coefficient to supplement the distribution information. Therefore, any deep network that can handle structured point clouds is applicable to the method in this paper.

4. The network structure of IVDN

For point clouds with structural information, 3D convolution operator can be used for feature extraction. The network structure of the IVDN used is shown in Figure 4. The forward network is adopted to extract point cloud features at different levels through convolution and pooling operation (Max pooling). Finally, classification prediction is completed through full connection.

The output of 3D convolution operation in the l layer is described as follows.

$$v_{lm}^{xyz} = b_{lm} + \sum_{q}\sum_{i=0}^{f-1}\sum_{j=0}^{f-1}\sum_{k=0}^{f-1} w_{lmq}^{ijk} v_{(l-1)q}^{(x+i)(y+j)(z+k)} \tag{4}$$

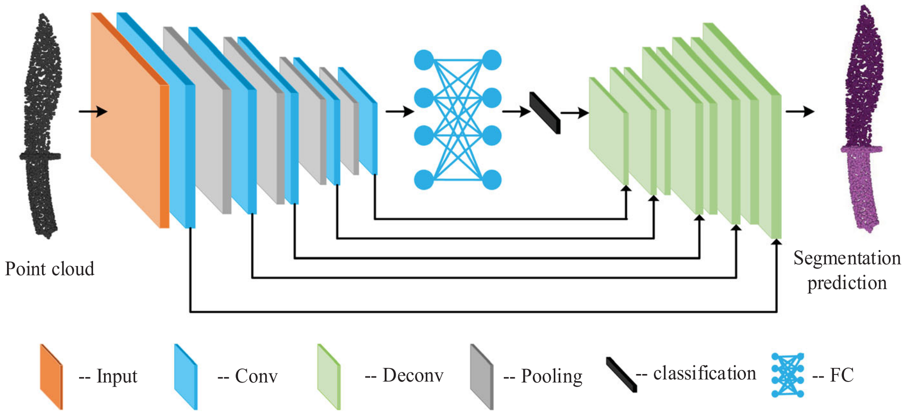

Figure 4. The network structure of IVDN.

where v_{lm}^{xyz} represents the m-th sub-block of the output feature map of layer l with the starting position of (x, y, z) and size of $f \times f \times f$. b_{lm} is the threshold deviation of the corresponding sub-feature block. $v_{(l-1)q}^{(x+i)(y+j)(z+k)}$ is the traversal of the convolution output feature map of the layer $l-1$. w_{lmq}^{ijk} is the weight of the q-th kernel at (i, j, k) in $v_{(l-1)q}^{xyz}$. f is the size of the feature map

$$f_i = \frac{f_{i-1} - F}{s} + 1 \tag{5}$$

where f_i is the size of the feature map at layer i, F is the size of the convolution kernel, and the stride is s.

The pooling output of layer l-th is

$$p_{lm}^{xyz} = \max_{i,j,k \in \{0,1,\ldots,g-1\}} p_{(l-1)m}^{(gx+i)(gy+j)(gz+k)} \tag{6}$$

where $p_{(l-1)m}^{(gx+i)(gy+j)(gz+k)}$ donates the parameter value traversal of the m-th feature map subblock at position (x, y, z). The size of the pooling layer kernel is $g \times g \times g$ and p_{lm}^{xyz} is the maximum output.

Each convolutional layer in the IVDN includes a 3 × 3 × 3 convolution kernel with stride 1, a batch normalization and a rectified linear unit. The first block use 32-filter convolutions and they are doubled in each successive block. IVDN has two fully connected layers, each of which contain a ReLU activation and a Dropout layer. The last fully connected layer followed by a softmax to regress to the probability of each category. The number of data set categories determines the number of nodes in this layer.

Forward network through training can get the weights and deviations of each convolution layer and the result of classification. Backward segmentation network relies on the classification prediction results of Forward network. And in turn, the forward features are connected to the segmentation network. Then the point cloud segmentation is realized by deconvolution operation. The deconvolution operation can be realized by reversing the positive and negative traversal of the convolution. For single label point cloud in the model, it only needs forward network to complete classification and segmentation, while for multi-label point cloud, it needs backward network to predict point-by-point label.

The segmentation task.is completed by mapping the predicted point tag to all points in the point cloud model. The mapping method corresponds to the point feature construction. If the number of fixed points selected is K, the number of output channels of the segmentation network is $K+1$. K channels correspond to the point label prediction of K points; The other channel is the total label of voxel. And the label with the most votes among K points determined by voting.

K points are selected from the voxel to span the eigenvectors. And Correspondingly, the decoding prediction results are mapped to each point when the segmentation task is completed. For empty voxels, the label is empty; For the number of points is less than or equal to K of voxels, each point corresponds to the predicted point label. And if the number of points is greater than K, K points correspond to K channel prediction tags. Finally, all other points are covered with voxel total tags to complete the tag prediction of all points.

5. Evaluation

The Model net40 (Wu et al., 2015) dataset and Shape net16 (Yi et al., 2016) dataset are used to verify the effectiveness of the proposed algorithm.

The Model net40 dataset is a point cloud deep learning dataset provided by Stanford university, mainly for verifying the classification of point clouds. This dataset contains point clouds of 40 categories of objects, with a total of 12,311 CAD models. Virtual scanner is used to obtain surface point cloud data, and each point cloud model is fixed with 10,000 points. The Shape net16 dataset is a subset of the Shape net model which contains point cloud data and point-by-point labels. There are 16,881 point cloud models including 16 kinds of objects, which are divided into 50 parts. And each model is divided into 2–6 parts. In this dataset, the gaussian function with mean value of zero and standard deviation of 0.02 is used to construct the rotation matrix, and the random jitter point cloud is enhanced. This paper uses the enhanced point cloud dataset and randomly selects 80% of the models for training and 20% for testing.

The experimental system comprised a Nvidia1080ti graphic processing unit, and a software environment of ubuntu16.04, cuda9.0, and cuDNN7.05. The deep learning framework was tensorflow1.8. The structured grid resolution of point cloud is 16 × 16 × 16, and select a fixed number of points $K = 4$. The convolution layer is composed of 5 × 5 × 5, 3 × 3 × 3, 1 × 1 × 1 convolution kernel, batch normalization and linear rectifying function (*RELU*). And the convolution kernel step size is 1. The parameter settings were as follows: batch-size = 32, momentum = 0.5, clipping = 0.99, initial learning rate = 1×10^{-4}.

MIoU (Mean Intersection over Union) was used to evaluate the segmentation effect, which is described below.

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{7}$$

where p_{ij} is the number of points that belong to class i but are predicted to be class j. In the same way, p_{ji} is the number of points that belong to class j but are predicted to be class i. P_{ii} is the number of correct points predicted.

Table 1. Mean classification accuracy on the model net40 dataset.

Method	Mean classification accuracy
PointGrid	0.852632
Ours	**0.857085**

After 200 iterations, the forward training model of segmentation network was obtained. In the test under this Model, the average classification accuracy of Model net40 dataset is 85.7% while the mean classification accuracy of Shape net16 dataset is 98.6%, and the *MIoU* is 77.3%.

In order to compare the performance of the IVDN algorithm in this paper, the PointGrid algorithm was used to classify and segment on the Model net40 and Shape net16 dataset. Table 1 shows the comparison results of two methods with the mean classification accuracy of Model net40 dataset. As the results show, the average classification accuracy of IVDN is 0.45% higher than that of PointGrid on single label M40 dataset compared with PointGrid.

Furthermore, the comparison results of the mean classification accuracy and *MIoU* of the two methods in the dataset of Shape net16 are shown in Table 2. And Table 3 shows the comparison results of classification accuracy and Iou of each category of the two methods in Shape Net16 dataset.

According to Tables 2 and 3, the mean classification accuracy of IVDN on Shape net16 dataset is improved by 0.45%, and the *MIoU* is improved by 0.3%. Moreover, the improvement of the two types of Earphone and Knife is especially obvious when the average accuracy and *MIoU* are improved. For Earphone, the classification accuracy increased by 20%, and the *IoU* increased by 0.38%. For Knife, the classification accuracy increased by 12.8%, and the *IoU* increased by 14.5%.

Table 2. Mean classification accuracy and *MIou* on Shape net16 dataset.

Method	Mean classification accuracy	*MIoU*
PointGrid	0.981148	0.770092
Ours	**0.985650**	**0.773123**

Table 3. Each category classification accuracy and *IoU* of Shape net16 dataset.

Class	Classification accuracy		*IoU*	
	PointGrid	Ours	PointGrid	Ours
Airplane	0.990706	0.998141	0.715490	0.712007
Bag	0.875000	0.875000	0.674935	0.668899
Cap	0.800000	0.800000	0.705827	0.705620
Car	0.997253	0.997253	0.687558	0.673497
Chair	0.994638	0.994638	0.854608	0.850345
Earphone	0.600000	**0.800000**	0.434958	**0.438768**
Guitar	0.993631	0.993631	0.855467	0.864712
Knife	0.846154	**0.974359**	0.648448	**0.793500**
Lamp	0.977346	0.951456	0.734876	0.727419
Laptop	0.988764	0.988764	0.921619	0.932248
Motorbike	1.000000	0.975000	0.515085	0.516928
Mug	0.944444	0.944444	0.850096	0.865511
Pistol	1.000000	1.000000	0.790120	0.806478
Rocket	0.846154	0.846154	0.408658	0.395426
Skateboard	0.966667	0.966667	0.638374	0.643128
Table	0.981956	0.988604	0.782993	0.790648

The comparison of the segmentation results of the two methods after visual rendering is shown in Figure 5. Figure 5(a–d) respectively shows the original Point cloud, ground truth, the segmentation effect of PointGrid and the segmentation effect of IVDN in this paper. The different colours in Figure 5 are only used to distinguish different parts. The segmentation of point clouds can be understood as the point-by-point classification of point clouds, so the IoU's promotion indicates that there are a number of points that predict the right label. The IoU of guitar in PointGrid is 0.855467 and that of guitar in IVDN is 0.864712. Figure 5 gives a rendering of the guitar

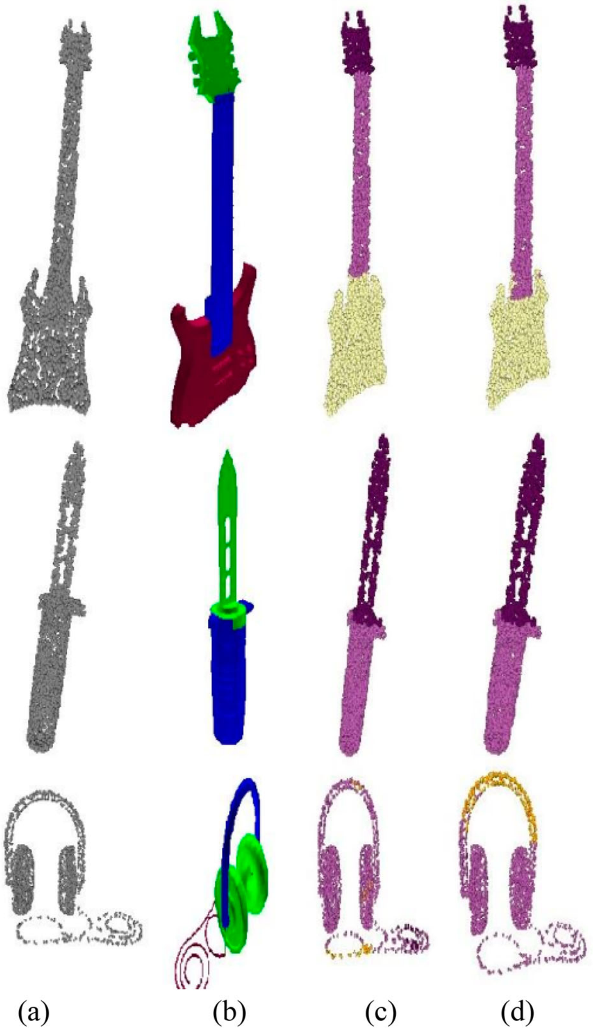

Figure 5. Contrast of segmentation effect.

under the above segmentation effect, it can be seen that the two methods have little difference in the segmentation effect of the head part of the guitar, while there are obvious differences in the neck and body part of the guitar. The IoU of knife in PointGrid is 0.648448 and that of guitar in IVDN is 0.793500, it can be seen that the middle point segmentation difference is greater. For earphones with three parts, the IoU in PointGrid is 0.434958 and that in IVDN is 0.438768, there are obvious differences in headphone, body and machine wire. The above is the visualization result of segmentation effect sampling. The detailed segmentation effect is shown in Table 3.

6. Conclusion

This paper selects a fixed number of points in voxel to achieve a more accurate description of the target details. In addition, the reliability coefficient highlighting voxel difference is added to the point feature vector to increase the descriptive ability of the target. Based on the voxel descriptor, an improved volumetric grid deep network model for point cloud segmentation was constructed to complete the point cloud segmentation task. According to the experimental results, the IVDN model can improve the classification accuracy and MIoU, and significantly improve the two categories of Earphone and Knife in Shape net16 dataset.

The proposed IVDN can provide more abundant information description at a certain resolution and construct a complete depth network with feature description. Other network optimization algorithms can be directly applied.

Acknowledgements

This research was partially supported by the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (CXTD2016054), National Natural Science Foundation of China (U1404612, 61973105), Key Scientific Research Projects of Universities in Henan Province (21A120004) and Practice and innovation training program for college students in Henan Province(2019
10460032).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Innovation Scientists and Technicians Troop Construction Projects of Henan Province: [grant number CXTD2016054]; National Natural Science Foundation of China: [grant number U1404612]; Key Scientific Research Projects of Universities in Henan Province: [grant number 21A120004]; Practice and innovation training program for college students in Henan Province: [grant number 201910460032].

References

Brock, A., Lim, T., Ritchie, J. M., & Weston, N. (2016). *Generative and discriminative voxel modeling with convolutional neural networks*. Computer ence arXiv:1608.04236 [cs.CV] (or arXiv:1608.04236v2 [cs.CV] for this version).

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184.

Huang, J., & You, S. (2016). Point cloud labeling using 3D convolutional neural network. *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2670–2675. https://doi.org/10.1109/ICPR.2016.7900038

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of The ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386.

Le, T., & Duan, Y. (2018). Pointgrid: A deep network for 3D Shape understanding. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9204–9214. https://doi.org/10.1109/CVPR.2018.00959

Li, Y., Pirk, S., Su, H., Qi, C., & Guibas, L. J. (2016). FPNN: Field probing neural networks for 3D data. *Advances in Neural Information Processing Systems*, 307–315. arXiv:1605.06240.

Maturana, D., & Scherer, S. (2015). Voxnet: A 3D convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 922–928. https://doi.org/10.1109/IROS.2015.7353481

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4460–4470. arXiv:1812.03828.

Mutz, F., et al. (2016). Large-scale mapping in complex field scenarios using an autonomous car. *Expert Systems With Applications*, *46*, 439–462. https://doi.org/10.1016/j.eswa.2015.10.045

Qi, C.R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3D classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–85. https://doi.org/10.1109/CVPR.2017.16.

Qi, C. R., Su, H., NieBner, M., Dai, A., Yan, M., & Guibas, L. J. (2016). Volumetric and multi-view CNNs for object classification on 3D data. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5648–5656. doi:10.1109/CVPR.2016.609.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. (2019). KPConv: Flexible and deformable convolution for point clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6410–6419. https://doi.org/10.1109/ICCV.2019.00651.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., & Tong, X. (2017). O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics*, *36*(4), 72. https://doi.org/10.1145/3072959.3073608.

Wang, D. Z., & Posner, I. (2015). Voting for voting in online point cloud object detection. *Robotics: Science and Systems*, *11*. doi:10.15607/RSS.2015.XI.035.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3D ShapeNets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1912–1920. doi:10.1109/CVPR.2015.7298801.

Xiao, W., Vallet, B., Xiao, Y., Mills, J., & Paparoditis, N. (2017). Occupancy modelling for moving object detection from lidar point clouds: A comparative study. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 171–178. https://doi.org/10.5194/isprs-annals-IV-2-W4-171-2017.

Yang, B., Luo, W., & Urtasun, R. (2018). PIXOR: Real-time 3D Object detection from point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7652–7660. https://doi.org/10.1109/CVPR.2018.00798.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., & Guibas, L. (2016). A scalable active framework for region annotation in 3D shape collections. *International Conference on Computer Graphics and Interactive Techniques*, *35*(6), 210. https://doi.org/10.1145/2980179.2980238.

Zeng, N., Li, H., Wang, Z., Liu, W., Liu, S., Alsaadi, F. E., & Liu, X. (2020). Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip. *Neurocomputing*, https://doi.org/10.1016/j.neucom.2020.04.001

Zeng, N., Wang, Z., Zhang, H., Kim, K.-E., Li, Y., & Liu, X. (2019). An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips. *IEEE Transactions on Nanotechnology*, *18*(1), 819–829. https://doi.org/10.1109/TNANO.2019.2932271

Zeng, N., Wang, Z., Zhang, H., Liu, W., & Alsaadi, F. E. (2016). Deep belief networks for quantitative analysis of gold immunochromatographic strip. *Cognitive Computation*, *8*(4), 684–692. https://doi.org/10.1007/s12559-016-9404-x

Zeng, N., Zhang, H., Song, B., Liu, W., Li, Y., & Abdullah, D. (2018). Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, *273*, 643–649. https://doi.org/10.1016/j.neucom.2017.08.043

Zhou, Y., & Tuzel, O. (2018). VoxelNet: End-to-end learning for point cloud based 3D object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4490–4499. doi:10.1109/CVPR.2018.00472