



# PointNet-Based Channel Attention VLAD Network

Rongrong Fan, Hui Shuai<sup>(✉)</sup>, and Qingshan Liu

Jiangsu Key Laboratory of Big Data Analysis Technology,  
Nanjing University of Information Science and Technology, Nanjing 211800, China  
{frr007, qslui}@nuist.edu.cn, huishuai13@163.com

**Abstract.** With the upgrading of application scenarios, computer vision is progressively expanded to 3D. Many methods that process point cloud directly provide a new paradigm for 3D understanding. Most of these methods employ maxpooling to handle the sparsity and disorder of point cloud. However, maxpooling layer extracts the global feature of the entire point cloud without learnable parameters, which is heuristics and insufficient. In this paper, we propose a VLAD enhanced Feature Aggregate Module to aggregate local features adaptively. In addition, a Channel Attention Module is applied to the features to reassemble the elements in high-dimension feature space. The experiments in both classification and segmentation demonstrate that the proposed method can improve the capacity of the baseline to extract more informative features. Specifically, we improve the accuracy from 88.5% to 89.8% for classification in ModelNet40 and improve the accuracy from 78.94% to 82.07% for semantic segmentation in S3DIS.

**Keywords:** Point cloud · Feature Aggregate Module · Channel Attention Module

## 1 Introduction

With the upgrading of application scenarios, 3D understanding has received a significant amount of attention in computer vision, especially for automatic driving and drone. Meanwhile, motivated by huge application demand, significant progress has been made in sensor technology and innumerable 3D data is generated by a depth camera, radar, and lidar. Consequently, 3D data has many formats such as voxels, meshes and point cloud owing to the diversity of sensor. Among these different 3D data, the point cloud is characterized by high accuracy and easy acquisition. The point cloud is a set of points with sparsity and disorder in 3D Euclidean space and the inherent irregular makes point cloud very different from 2D data. To enable UAVs [1] and unmanned driving [2] to perceive a 3D scene, high-level semantic understanding of 3D data is required. In common with 2D computer vision, the primary tasks for 3D understanding

---

This is a student paper.

are classification and segmentation. However, the input in 2D computer vision is usually images and videos, which are organized in a regular format. Although deep learning [3] has revolutionized many research fields in computer vision, conventional convolutional neural networks are not suitable for the point cloud. Therefore, many methods have been proposed to process point cloud and this paper will focus on the identification of point clouds based on convolutional neural networks. Recently, popular neural networks based methods for point cloud processing can be divided into four categories:

1. Voxel-based convolutional neural networks: These methods transform the point cloud into voxels and then employ 3D convolution neural networks on voxels, such as VoxNet [4] proposed by Maturaba and Scherer. However, the sparsity of the data causes a loss of details. FPNN [5] and Voted3D [6] proposed a special method to deal with the sparsity problem, but convolution is still limited to the sparse voxel. Besides, there are still huge challenges in dealing with large scenes. Some researchers have optimized the network in the data structure. For example, Klovov et al. proposed Kd-Net [7], Wang et al. [8] proposed O-CNN, and Riegler et al. [9] proposed Oct-Net. But sparse 3D data with 3D convolution kernels suffer from computation and memory cost.
2. Multi-view based Convolutional neural networks: Researchers try to process 3D data by referring to 2D data processing methods. For example, rendering 3D data into 2D images from different perspectives [10–12], and then using traditional 2D convolutional neural networks. This paradigm has achieved good results in classification and retrieval thanks to the abstract ability of deep learning. Among them, Su et al. [11] gather the information from multiple views of the 3D object together and turn them into a single compact shape descriptor, which is known as MVCNN. However, multi-view convolutional neural networks are difficult to extend to the segmentation in 3D data. When rendering 3D data, the choice of angle affects the final experimental results. In fact, how to select the angle in this method is also difficult. In addition, rendering the 3D data into 2D data may lose part of the 3D spatial position information, and the data processing process is relatively complicated.
3. Feature-based deep learning network: Fang et al. [13] and Guo et al. [14] convert traditional 3D data into corresponding feature description vectors and then use the fully connected network to obtain the result of classification. Because the features are manually designed, the quality of the features selected directly affects the performance of the network significantly, the process of selecting the original data features will be more complicated.
4. Point cloud based deep learning network: Qi et al. proposed PointNet [15] and PointNet++ [16] to directly deal with the unordered point cloud. These methods are not only convenient but also can preserve the integrity of the point cloud. All points are independently handled to extract local features, sharing multiple multi-layer perceptrons. Maxpooling layer is used to aggregate the global feature from local features because of its permutation invari-

ance. But maxpooling layer has no learnable parameters, which makes this process heuristics and insufficient.

In order to alleviate the weakness caused by the insufficiency of maxpooling layer, we propose the VLAD enhanced Feature Aggregate Module to extract more sufficient global feature and Channel Attention Module to reassemble the elements in high-dimension feature space. The VLAD enhanced Feature Aggregate Module is robust to the order of input points and stores the residuals for each point to the centers in a trainable manner. The Channel Attention Module strengthens the representational power of convolutional layers by enhancing the spatial encoding throughout its feature hierarchy. The architecture of our network is illustrated in Fig. 1 and our contributions are as follows:

1. We develop a convolutional neural network with VLAD enhanced Feature Aggregate Module and Channel Attention modules to extract more informative global feature for 3D point cloud processing in an end-to-end manner.
2. We demonstrate that the limitations of maxpooling layer can be alleviated with some learnable feature aggregate modules robust to the order of points, while the theoretical analysis about the VLAD is provided.
3. We improve the accuracy from 88.5% to 89.8% for classification in ModelNet40 and improve the accuracy from 78.94% to 82.07% for semantic segmentation in S3DIS, which verifies the effectiveness of the proposed method.

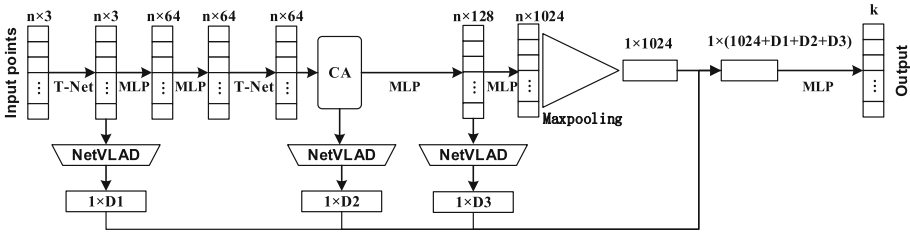


Fig. 1. The architecture of proposed method

## 2 Related Work

The structure of PointNet is shown in Fig. 2, which can serve as the classification network and segmentation network. In the pointnet framework, multi-layer perceptron (MLP) transforms the 3D coordinate into high-dimensional feature space. Due to the independence of point-wise transform, the point cloud is easy to apply the rigid or affine transformation. Therefore, the T-Net [17] is used for transforming the points adaptively.

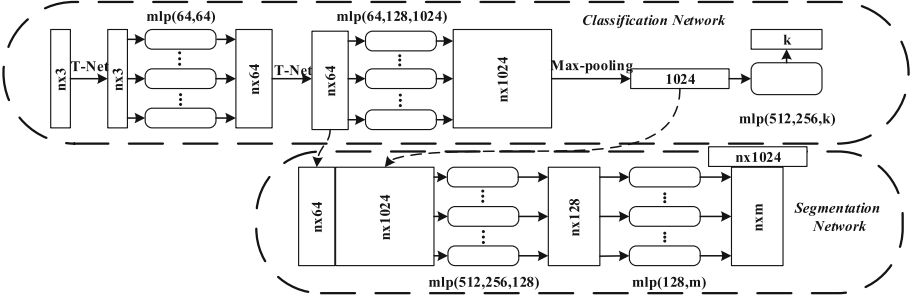


Fig. 2. Structure of PointNet

Formally, given an unordered point set, where an aggregate function can be defined as follows:

$$f(x_1, x_2, \dots, x_n) = \gamma(\max_{i=1, \dots, n} h(x_i)) \quad (1)$$

Where  $\gamma$  and  $h$  usually refer to MLPs to transform the features. It can be proved that any continuous aggregate function can be arbitrarily approximated. In this way, points in 3D are transformed into more informative high-dimensional features and the aggregated global feature is robust to the disorder of point cloud. However, there are two problems in the PointNet. (1) When projecting the low dimensional features to high-dimensional features, the surrounding context of the point is not used. Due to this, the network can't capture the contextual features. (2) When using the maxpooling operation, the feature components of different points are used directly to replace the features of the entire input point cloud, resulting in the loss of surface information.

## 3 Method

### 3.1 Channel Attention Module

Generally, the importance of different feature components varies a great deal for the final decision. Taking images understanding as an example, an important feature is usually a region where are corners, edges. In PointNet, features are transformed into high-dimension space via MLP while MLP is usually implemented with  $1 \times 1$  convolution operation. The amount of convolution kernels determines the dimension of the target feature and the components of this feature are supposed to be reassembled for better expression capacity. Inspired by this, we designed a channel-based attention mechanism, called Channel Attention module, referred to as CA. CA module is data-driven processing that enhances representative features and suppresses weaker features. Given a corresponding input, the CA module can be formulated as:

$$X \xrightarrow{CA} Y, X \in R^{N \times D \times C}, Y \in R^{N \times D \times C} \quad (2)$$

Where  $X$  is the input,  $Y$  is the output,  $N$  is the size of the point cloud,  $D$  is 1.  $C$  is the number of channels. The size of input and output is identical, so the CA module can be embedded into any network easily. The specific operation is shown in Fig. 3. Referring to the idea of Qi et al., in the CA module, we employ the maxpooling operation to retain the most effective features. To obtain the most important channel information, we use a fully connected network for further dimension reducing. The feature information is compressed so that the reserved channel features are more significant. After that, the nonlinearity of the network is increased by the ReLU. Then we use another fully connected network to recover the dimension of the channel with Sigmoid as the activate function. So the number of the channel is the same as the input. Finally, we do the channel weighting and fuse the weighted channel feature with the original features.

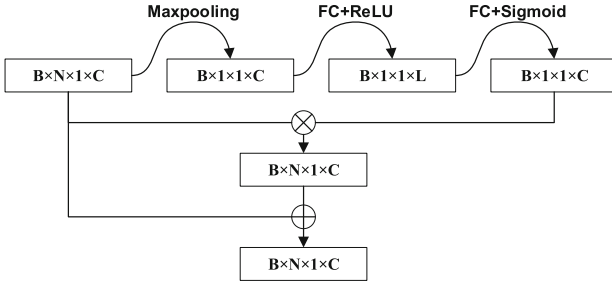


Fig. 3. Channel Attention module

The CA module can rank the importance of the components in the feature and reassemble them, which is an implementation of feature selection in deep learning. In addition, due to the presence of maxpooling in CA module, the global information is fused with local features in an early stage. It ultimately enhances the capability of the network to extract more informative global features.

### 3.2 NetVLAD Module

Jegou et al. [18] first proposed a local aggregation descriptor vector (VLAD), which is regarded as a simplification of the Fisher kernel. Fisher kernel captures statistical information about the local descriptors aggregated on the image, while VLAD stores the sum of the residuals of each descriptor. Formally,  $N$  local image descriptors of  $\{x_i\}$  with  $D$  given dimension is taken as input, and there are  $K$  cluster centers.  $\{c_k\}$  are the parameters of VLAD. The description vector  $V$  of the output VLAD for the entire image is  $D \times K$ . For convenience, the vector is written as a  $D \times K$  matrix. When used as an image representation, the matrix needs to be converted to a vector and normalized. The  $(j, k)$  element of  $V$  can be expressed as:

$$V(j, k) = \sum_{i=1}^N \alpha_k(x_i) (x_i(j) - c_k(j)) \quad (3)$$

Where,  $x_i(j)$  represents the  $j$ -th dimension of the  $i$ -th descriptor,  $c_k(j)$  represents the  $j$ -th dimension of the  $k$ -th cluster center.  $\alpha_k(x_i)$  represents the relationship between  $x_i$  and  $k$ . Specifically,  $\alpha_k(x_i) = 1$  if the cluster is closest to the descriptor; otherwise,  $\alpha_k(x_i) = 0$ . Intuitively, the  $D$  dimension in column  $k$  of the vector represents the sum of the descriptor residuals  $(x_i - c_k)$  assigned to the cluster  $c_k$ . Then, the matrix  $V$  is regularized according to the column, converted to a vector, and then regularized.

Inspired by the local aggregate descriptor vector (VLAD) representation, Arandjelovic et al. [19] proposed a new end-to-end convolutional neural network structure that can be used for scene recognition. The main components of this neural network is NetVLAD. NetVLAD is a new universal VLAD layer that excels in image retrieval and location recognition. This network structure can be easily embedded in any CNN framework and can be trained through backpropagation.

The VLAD is discontinuous because of the hard assignment of the descriptors while training through back-propagation requires the module to be differentiable. The problem lies in making VLAD differentiable and Arandjelovic et al. handled this by replacing the hard assignment of descriptors with the soft assignment of descriptors:

$$\bar{\alpha}_k(x_i) = \frac{e^{-\alpha\|x_i - c_k\|^2}}{\sum_{k'} e^{-\alpha\|x_i - c_{k'}\|^2}} \quad (4)$$

The former equation is equivalent to the proximity of other cluster centers, and the weight of the descriptor is assigned to the cluster whose proximity is proportional. The range of  $\bar{\alpha}_k(x_i)$  is between 0 and 1, with the highest weight assigned to the nearest cluster center.  $\alpha$  is a positive constant that controls the magnitude of the attenuation of the response. It can be noted that this setting is the same as the original VLAD.

By extending the square of equation, the  $e^{-\alpha\|x_i\|^2}$  in denominator and the intermolecular can be eliminated:

$$\bar{\alpha}_k(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} \quad (5)$$

Among them, vector  $w_k = 2\alpha c_k$ , scalar  $b_k = -\alpha\|c_k\|^2$ . Substituting Eq. (5) into Eq. (3), the final form of NetVLAD can be obtained:

$$V(j, k) = \sum_{i=1}^N \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \quad (6)$$

Where  $\{w_k\}$ ,  $\{b_k\}$  and  $\{c_k\}$  are the set of parameters that can be trained in each cluster. Similar to the original VLAD descriptor, the NetVLAD layer aggregates the first-order statistic of the residuals in different parts of the descriptor space, which is weighted by the soft assignment of the descriptors to the corresponding cluster. It is worth noting that the NetVLAD layer has three sets of independent parameters  $\{w_k\}$ ,  $\{b_k\}$  and  $\{c_k\}$  compared to  $\{c_k\}$  of the original VLAD,

which is more flexible than the original VLAD. And all parameters of NetVLAD can be obtained automatically. The NetVLAD layer was originally designed to aggregate the local image features known by VGG and AlexNet into the VLAD global descriptor. By sending the local feature descriptor of the point cloud into the neural network, the global representation can also be generated. Descriptor vector can be viewed as a supplement to the max-pooling operation. Besides, it allows end-to-end training and reasoning and can extract global descriptors from a given 3D point cloud. Because of the disorder of the point cloud, the NetVLAD layer needs to be insensitive to the order of the point cloud. In the following proof, it can be concluded that NetVLAD is a symmetric function, that is, it can be applied in the local features to generate global features with permutation invariance.

As shown in Fig. 4, the input of the NetVLAD layer is a high-dimensional feature of the point cloud. It can be obtained by projecting the features with the MLPs. The output is the VLAD descriptor of the input feature. However, the VLAD descriptor is a high-dimensional vector, i.e., a  $(D \times K)$  dimensional vector. To alleviate resource conservation, a fully connected layer can be used to compress the  $(D \times K)$  vector into a more compact output feature vector, which is then quadraticized to generate the final global descriptor vector.

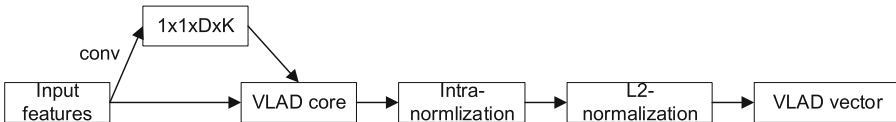


Fig. 4. NetVLAD layer structure

### 3.3 Proof of Symmetry of NetVLAD

Pixels of an image have a fixed spatial position, so there is no need to consider the order of input pixels when using filters. However, when it comes to point clouds, the order of points matters. The output of traditional convolutional neural network varies when the order of point cloud changed. Therefore, methods processing points directly must characteristic with permutation invariance. In other words, the points in different orders should produce the same output. In this paper, we use the NetVLAD architecture to get the features of the point cloud because it's symmetrical. The invariance of the NetVLAD layer for the point cloud order is demonstrated below. Given the input point cloud, the MLP independently transforms the input features to another feature space. To prove that NetVLAD is symmetrical, it means the output of the result is irrelevant with the order of the input point cloud.

*Proof.* Assuming that the characteristics of the input point cloud  $P$  are expressed as  $\{p'_1, p'_2, \dots, p'_N\}$ , the output of the NetVLAD is  $V = [V_1, V_2, \dots, V_k]$ , for  $\forall k$ , we have

$$V_k = h_k(p'_1) + h_k(p'_2) + \dots + h_k(p'_N) = \sum_{t=1}^n h_k(p'_t) \quad (7)$$

where  $V_k(p')$  satisfying

$$V_k(p') = \frac{e^{w_k^T p' + b_k}}{\sum_{k'} e^{w_{k'}^T p' + b_{k'}}} (p' - c_k)$$

Suppose there is another point cloud  $\tilde{P} = \{p_1, \dots, p_{i-1}, p_j, p_{i+1}, \dots, p_{j-1}, p_i, p_{j+1}, \dots, p_N\}$ , when  $\tilde{P}$  are the same as  $P$  except for the order of  $p_i$  and  $p_j$ . So for  $\forall k$ , we have

$$\begin{aligned} \tilde{V}_k &= h_k(p'_1) + \dots + h_k(p'_{i-1}) + h_k(p'_j) + h_k(p'_{i+1}) + \dots \\ &+ h_k(p'_{j-1}) + h_k(p'_i) + h_k(p'_{j+1}) = \sum_{t=1}^n h_k(p'_t) = V_k \end{aligned} \quad (8)$$

From the former equation, we can draw the conclusion that NetVLAD is symmetrical. Therefore we can use the NetVLAD module to enhance the global feature of the point cloud.

## 4 Experiments

In this paper, we incorporate the CA module and NetVLAD module into the original PointNet network. The corresponding classification network and segmentation network are designed respectively. The data set of the classification experiment is ModelNet40 [20], and the data set used in the segmentation experiment is S3DIS [21]. The proposed framework is effective both in classification and in segmentation.

### 4.1 3D Object Classification

The dataset for classification is ModelNet40. It includes 12,311 CAD models, of which 9843 are for training and 2,468 are for testing. The same data used for the PointNet 3D target classification is to evenly sample 2048 points on the mesh surface and normalize them to a unit sphere. During training, training data is augmented by rotating the upper axis and dithering the points by Gaussian noise with zero mean and 0.02 standard deviation. The experimental is conducted on Ubuntu 14.04, and the framework is Tensorflow. Same as PointNet, each experiment has a batch size of 32, the number of input points is 1024, the initial learning rate is 0.001, the learning rate attenuation parameter is 0.7, the step size is 20000 and the optimizer is Adam. PointNet consists of five convolution



layers and one maximum pooling layer. In order to verify the validity of the CA module proposed, we embed CA modules in different locations. The results are shown in Table 1:

**Table 1.** Classification results for different CA module locations

Location	Accuracy avg. class	Accuracy overall
PointNet	85.5	88.5
CONV_1	<b>86.4</b>	89.3
CONV_2	85.3	88.4
CONV_3	<b>86.4</b>	89.0
CONV_4	85.9	88.9
CONV_5	85.9	88.5
MAX_POOLING	86.2	88.6

When testing the CA module, the data set used is ModelNet40, the setting is consistent with PointNet, the number of input points is 1024, and the dimension reduction factor of the CA channel is 4. It shows that the CA module improves the performance when embedded into most convolutional layers especially the first and third convolutional layer. It can be concluded that the CA module is effective, but the embedding location is sensitive.

**Table 2.** Classification results for ModelNet40

Methods	Input	# views	Accuracy avg. class	Accuracy overall
SPH	mesh	–	68.2	–
VoxNet	voxel	12	83.0	85.9
Subvolume	voxel	20	86.0	89.2
LFD	image	10	75.5	–
MVCNN	image	80	90.1	–
PointNet (vanilla)	point	–	83.6	87.4
PointNet (baseline)	point	–	85.5	88.5
CA-VLADNet	point	–	86.5	89.8

As we can see in Table 2, compared with previous works, the proposed method achieves better performance. However, there is still a certain gap between the proposed method and the multi-view based method (MVCNN) owing to the information loss in the sampling process. In preprocessing, only 1024 points are sampled from point cloud as the input in the proposed method while a large number of images can be obtained by rendering in MVCNN. It is the lack of geometric details that results in this gap.

## 4.2 3D Semantic Segmentation

The dataset for semantic segmentation is S3DIS data set. It is a large-scale semantic 3D dataset constructed by Armeni et al. of Stanford University. The data set detected 13 semantic elements, including structural elements (ceiling, floor, wall, beam, pillar, window, and door), common items and furniture (tables, chairs, sofas, bookcases, and planks), and finally type of clutter, each point in the scan is labeled with one of them. The dataset is divided into rooms, and the room is divided into areas of 1 m by 1 m. Each of these points is represented by a 9-dimensional vector from the three-dimensional coordinates XYZ, the color information RGB, and the normalized position of the opposing room (from 0 to 1). During training, 4096 points are randomly extracted from each block randomly, in testing, all points are tested. As mentioned in Armeni et al. [21], training and testing were performed in the k-fold strategy. The batch size is 24, the learning rate is set to 0.001, the learning rate attenuation parameter is 0.5, and the optimizer is Adam. According to Qi et al. [15], the S3DIS data set is divided into 6 regions, and the method of six-fold cross-validation is used. Table 3 shows the six-fold cross-validation results on the S3DIS.

**Table 3.** Semantic segmentation results for S3DIS

Region	Evaluation	PointNet (baseline)	AC-VLADNet
Region one	IOU	52.86	57.06
	Accuracy	80.82	83.57
Region two	IOU	28.92	34.51
	Accuracy	64.23	73.70
Region three	IOU	54.76	59.20
	Accuracy	83.59	85.99
Region four	IOU	40.06	42.90
	Accuracy	78.55	80.48
Region five	IOU	41.98	43.74
	Accuracy	80.03	80.76
Region six	IOU	47.23	50.99
	Accuracy	78.94	82.07
Average	IOU	47.23	50.99
	Accuracy	78.94	82.07

It can be seen that the IOU and accuracy of each region in this model are higher than that of PointNet. The average IOU of the six regions is about 3.76% higher than the baseline, and the accuracy rate is increased by 3.13%. The result demonstrates that the proposed method is feasible to extract more informative features for semantic segmentation while semantic segmentation rely more on the detail context of the point cloud.

## 5 Conclusion

We presented a VLAD enhanced PointNet equipped with Channel Attention module for 3D point cloud processing. Both VLAD enhanced Feature Aggregate Module and Channel Attention Module are readily pluggable into any convolutional neural network and trained in an end-to-end manner. Most remarkable of all is that the proposed method aggregate global features with learnable parameters while keeping the robustness to the order of points. The experiments in classification and segmentation verify the effectiveness of the proposed method and the necessity of improving maxpooling to aggregate more informative global features.

## References

1. Mozaffari, M., Saad, W., Bennis, M., et al.: A tutorial on UAVs for wireless networks: applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **21**, 2334–2360 (2019)
2. Zhang, X., Gao, H., Guo, M., et al.: A study on key technologies of unmanned driving. *CAAI Trans. Intell. Technol.* **1**(1), 4–13 (2016)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436 (2015)
4. Maturana, D., Scherer, S.: VoxNet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928. IEEE (2015)
5. Li, Y., Pirk, S., Su, H., et al.: FPNN: field probing neural networks for 3D data. In: *Advances in Neural Information Processing Systems*, pp. 307–315 (2016)
6. Wang, D.Z., Posner, I.: Voting for voting in online point cloud object detection. In: *Robotics: Science and Systems*, vol. 1, no. 3, p. 10.15607 (2015)
7. Klovov, R., Lempitsky, V.: Escape from cells: deep Kd-networks for the recognition of 3D point cloud models. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 863–872 (2017)
8. Wang, P.S., Liu, Y., Guo, Y.X., et al.: O-CNN: octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph. (TOG)* **36**(4), 72 (2017)
9. Perdomo, O., Otálora, S., González, F.A., et al.: OCT-NET: a convolutional network for automatic classification of normal and diabetic macular edema using SD-OCT volumes. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 1423–1426. IEEE (2018)
10. Qi, C.R., Su, H., Nießner, M., et al.: Volumetric and multi-view CNNs for object classification on 3D data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5648–5656 (2016)
11. Su, H., Maji, S., Kalogerakis, E., et al.: Multi-view convolutional neural networks for 3D shape recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945–953 (2015)
12. Su, H., Wang, F., Yi, E., et al.: 3D-assisted feature synthesis for novel views of an object. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2677–2685 (2015)
13. Fang, Y., Xie, J., Dai, G., et al.: 3D deep shape descriptor. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2319–2328 (2015)

14. Guo, K., Zou, D., Chen, X.: 3D mesh labeling via deep convolutional neural networks. *ACM Trans. Graph. (TOG)* **35**(1), 3 (2015)
15. Qi, C.R., Su, H., Mo, K., et al.: PointNet: deep learning on point sets for 3D classification and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660 (2017)
16. Qi, C.R., Yi, L., Su, H., et al.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems*, pp. 5099–5108 (2017)
17. Jaderberg, M., Simonyan, K., Zisserman, A.: Spatial transformer networks. In: *Advances in Neural Information Processing Systems*, pp. 2017–2025 (2015)
18. Jegou, H., Perronnin, F., Douze, M., et al.: Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(9), 1704–1716 (2012)
19. Arandjelovic, R., Gronat, P., Torii, A., et al.: NetVLAD: CNN architecture for weakly supervised place recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307 (2016)
20. Wu, Z., Song, S., Khosla, A., et al.: 3D ShapeNets: a deep representation for volumetric shapes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920 (2015)
21. Armeni, I., Sener, O., Zamir, A.R., et al.: 3D semantic parsing of large-scale indoor spaces. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543 (2016)