Complex & Intelligent Systems (2022) 8:3833–3845
https://doi.org/10.1007/s40747-021-00618-0

ORIGINAL ARTICLE

Semantic segmentation of large-scale point clouds based on dilated nearest neighbors graph

Lei Wang[1,2] · Jiaji Wu[3] · Xunyu Liu[4] · Xiaoliang Ma[4] · Jun Cheng[1,2]

Received: 21 August 2021 / Accepted: 3 December 2021 / Published online: 4 January 2022

Abstract

Three-dimensional (3D) semantic segmentation of point clouds is important in many scenarios, such as automatic driving, robotic navigation, while edge computing is indispensable in the devices. Deep learning methods based on point sampling prove to be computation and memory efficient to tackle large-scale point clouds (e.g. millions of points). However, some local features may be abandoned while sampling. In this paper, We present one end-to-end 3D semantic segmentation framework based on dilated nearest neighbor encoding. Instead of down-sampling point cloud directly, we propose a dilated nearest neighbor encoding module to broaden the network's receptive field to learn more 3D geometric information. Without increase of network parameters, our method is computation and memory efficient for large-scale point clouds. We have evaluated the dilated nearest neighbor encoding in two different networks. The first is the random sampling with local feature aggregation. The second is the Point Transformer. We have evaluated the quality of the semantic segmentation on the benchmark 3D dataset S3DIS, and demonstrate that the proposed dilated nearest neighbor encoding exhibited stable advantages over baseline and competing methods.

Keywords Semantic segmentation · Dilated neighborhood · Deep neural network

Introduction

Automatic driving and robotics have obtained rapid progress in recent years, and one important reason is the development of edge computing which makes real-time computation to be achievable. Automatic driving and robotic navigation always use LIDAR to collect point clouds for recognition of 3D objects. Point cloud is one kind of 3D geometrical data containing 3D coordinate on every point of the scanned object. Compared with two-dimensional (2D) images, point clouds have some advantages. First, they can represent 3D shapes or objects. Second, their 3D coordinates are not effected by climate or illuminations. Third, accurate distance can be calculated from the data. So point clouds have many latent applications including creation of 3D CAD models, metrology and quality inspection, visualization, animation, rendering.

The recognition and semantic segmentation of 3D point clouds plays an important role for scene understanding in intelligent systems, such as robotics [1–4], automatic driving for navigation [5–10] or interaction tasks in real-world environments. However, the processing of point clouds is challenging since they are unstructured, unordered, and contains a varying number of points.

Feature extraction is always the first step for recognition and segmentation. Traditional methods for point clouds include 3D Harris [11], intrinsic shape signature [12], point feature histograms [13], viewpoint feature histogram [14],

✉ Jun Cheng
jun.cheng@siat.ac.cn

Lei Wang
lei.wang1@siat.ac.cn

Jiaji Wu
wujj@mail.xidian.edu.cn

Xunyu Liu
2070276210@email.szu.edu.cn

Xiaoliang Ma
maxiaoliang@szu.edu.cn

[1] Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS), Shenzhen, China

[2] The Chinese University of Hong Kong, Hong Kong, China

[3] School of Electronic Engineering, Xidian University, Xi'an, China

[4] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

eigenvalues [15], subspace selection [16], etc. The effectiveness of these methods can be evaluated in respect of invariance of rigid transform, discriminative ability, robustness. But the above methods have some limitations since they are sensitive to the change of data mode and application scenarios.

Recently Deep Neural Networks (DNNs) have been used for processing point clouds [17] due to the powerful representation ability. Some different kinds of networks have been designed. The first is the voxel-based methods, such as VoxNet [18] and VoxelNet [19]. In the VoxNet [18], a volumetric occupancy grid representation for point cloud is integrated with a 3D Convolutional Neural Network (CNN) for 3D object recognition. In [19], a voxel feature extractor (VFE) network is proposed to transform voxels into fixed-dimensional feature vectors. Volumetric representation is constrained by its resolution due to the computation cost of 3D convolution. The second kind of methods directly take point cloud data as input, among which PointNet [20] is one of the most representative methods. It directly tackles point clouds and respects the permutation invariance of input points. The point-wise features are learned using shared multilayer perceptions. Although efficient and effective, it is unaware of context information because max-pooling operation aggregates a batch of features into one feature. PointNet++ [21] is introduced to extract local features from partitions of point clouds to deal with the problem so that it can be used in large scale scenes.

To deal with large-scale point clouds, sampling is performed first to reduce the redundant computation. However, there is still a problem that the point sampling methods used in these methods are either computationally expensive or memory inefficient. Hu et al. [22] proposed a solution using random point sampling instead of heuristic sampling or learning-based sampling methods. Accordingly, a local feature aggregation module is proposed, which uses local feature encoding, attention pooling and dilated residual block to extract the features of random sampling points. The shared multi-layer perception is used for up-sampling and decoding to obtain the final result of semantic segmentation. The backbone of the network is a typically encoding–decoding architecture. During the process of down sampling, K-Nearest Neighbor (KNN) is used for feature aggregation, which causes other points' features to be discarded which are outside of the nearest neighbors.

To utilize more points with geometric information, in this paper, we propose a method to increase the receptive field of neural network by dilated neighborhood with the same number of neural network parameters. We have verified the efficiency in two different frameworks. The first is based on the random sampling and local feature aggregation network (RandLA-Net [22]). The second is based on the Point Transformer [23]. Both frameworks have their special advantages. The RandLA-Net takes a fast sampling strategy and uses local feature aggregation to make up for the lost features. The Transformer does not use any CNN or RNN, and it is based on self-attention network, which have been successfully used in natural language processing by measuring the relationship between every word and others in the sentence. The self-attention is a set operator which is invariant to permutation and cardinality of the input elements, so it is appropriate for point clouds which are actually sets embedded in 3D space. In the experiments, we will show the efficiency of the proposed dilated nearest neighbor encoding in both frameworks.

The main contributions of this work are summarized as follows:

- A dilated nearest neighbor encoding is introduced to the point cloud sampling network to broaden the network's receptive field in the purpose of learning more 3D geometric information.
- We have designed one end-to-end framework based on random sampling and the dilated nearest neighbor encoding for 3D point cloud semantic segmentation to illustrate its efficiency. And we have also verified the effectiveness of the dilated nearest neighboring encoding in the framework of the Point Transformer.
- Better performance than state-of-the-art methods has been achieved on the large-scale benchmark datasets.

Related works

A number of methods have been proposed for feature extraction of point clouds, and based on this, 3D object classification and recognition methods have also been developed. Since we focus on semantic segmentation in this paper, we will introduce related works in the area of feature learning and semantic segmentation of point clouds.

Conventional methods

Feature learning of point cloud has been studied in the past decades, and handcrafted features have been always used in the conventional methods. Histogram can represent accumulated information, so this kind of methods have also been introduced for point clouds to learn their 3D geometric features, such as point feature histograms (PFHs) [24], fast point feature histograms (FPFHs) [13]. Various histogram-based methods have been proposed, and they have been compared in [25]. The 3D covariance matrix from the neighboring points' coordinates have been used for describing the local 3D structure [26], as well as covariance of angular measures and point distances [27]. Weinmann et al. [28] presented 2D and 3D point cloud features for automated large-scale scene analysis, including basic geometric properties (e.g.

absolute height, radius, local point density, local normal vector), 3D structure and shape features (general distribution, normalized eigenvalues, linearity, planarity, scattering, omnivariance, anisotropy, eigenentropy, local surface variation, etc.). One question in [28] is that they compute features at multiple scales, so this method is time-consuming. Hackel et al. [15] proposed a fast semantic segmentation method for 3D point clouds based on carefully handling of points' neighborhood relations. They first extract a rich and expressive set of features to capture the geometric properties of a point's neighborhood. The 3D features are based on neighboring points' covariance (sum, omnivariance, eigenentropy, anisotropy, planarity, linearity, surface variation, sphericity, verticality), moment (1st and 2nd order, 1st and 2nd axis) and height (vertical range, height below, height above). Based on these feature, a classifier is trained to predict class-conditional probabilities. One limitation of the handcrafted feature-based methods is that they are usually designed for specific tasks.

Projection-based networks

To use large-scale 2D image datasets and successful 2D convolutional networks, the irregular point data is projected to other forms suitable for 2D approaches. In [29], the point clouds are projected into 2D map position through the azimuth and elevation angle of viewing the point. In [30], 3D shapes features are learned by rendered views on 2D images, and a multi-view CNN is introduced for 3D shape in a single and compact shape descriptor. Point cloud is converted to a stacked pillar (vertical column, a voxel in the z direction) tensor and pillar index tensor in the work of [31]. For autonomous driving, the authors of [6] designed a network to take the bird's eye view and front view of LIDAR point cloud as well as an RGB image as input. The bird's eye view features include hight maps, density and intensity. The front view features include height, distance and intensity. And the front view is projected to a cylinder plane. In [7], PIXOR is proposed for 3D object localization in autonomous driving, which also takes bird's eye view representation as input. Their feature representation consists of a 3D occupancy tensor and a 2D reflectance image. One limitation of the projection-based methods is that the 3D geometric features may be lost.

Voxel-based networks

Since of the regulation of the voxel grids, some methods use the voxel as the representation of point clouds' features [18,19], which is referred as 3D voxelization. First, the irregular point clouds will be split to voxels. The voxel is encoded as fixed feature vectors, and then used as input for the following 3D convolution neural network to extract features further more. Submanifold sparse convolutional networks have been proposed for high-dimensional, sparse input data, and verified in semantic segmentation of 3D point clouds [32]. A voxel VAE network (VV-Net) is introduced for robust point segmentation in [33], in which a radial basis function based variational auto-encoder is used.

Basic point-based networks

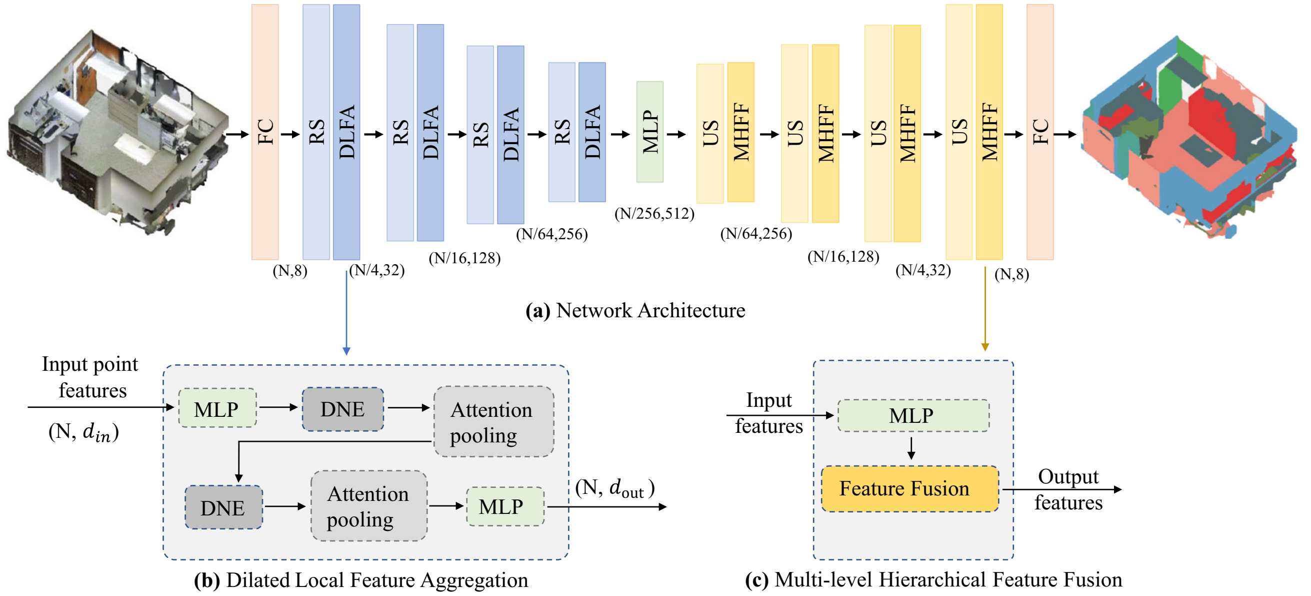
Some works directly use point clouds as input [34–39]. PointNet uses the shared multilayer perceptions (MLP) to learn per-point features, and uses max-pooling for global feature to solve the unordered data question [20]. PointNet++ learns hierarchical features in a metric space after furthest point sampling, with multi-scale and multi-resolution grouping [21]. Qi et al. [40] utilize PointNet as its basic feature extractor and use sliding frustums to construct mapping between 2D image and 3D point clouds. PointCNN is proposed in [41], and a transform is learned for the coordinates of points to weight and permute the input features, followed by a convolution, together as a basic building block for the framework. PointRCNN [42] generates proposals of bounding boxes directly from the segmented foreground point set, and then fine-tunes such proposals through transformation into canonical coordinates.

Based on a graph convolutional network, superpoint graph (SPG) is proposed in [43] to capture the organization of 3D point clouds with a compact and rich representation of contextual relationships between object parts. A graph attention convolution (GAC) is proposed in [36], the kernels of which can be dynamically carved into specific shapes to adapt to the structure of different objects. It has been used for fine-grained segmentation of point clouds. A grouping technique-based method is proposed in [44] to incorporate neighborhood information from the feature space and the world space, as well as a pairwise distance loss and a centroid loss.

Extended deep neural networks

Wang et al. [45] proposed a framework of Associatively Segmenting Instances and Semantics (ASIS). This framework associates instance and semantic segmentation together based on the consideration that two tasks can benefit from each other to boost respective performance. Specifically, instance segmentation is boosted by learning semantic-aware point-level instance embedding, while semantic segmentation is boosted by fusing the semantic features of the points belonging to the same instance.

A Joint instance semantic Segmentation neural Network (JSNet) is proposed in [46], which includes a shared feature encoder, two parallel branch decoder, a feature fusion module for each decoder, and a joint segmentation module. The joint instance and semantic segmentation module transforms semantic (instance) features into instance (semantic)



(a) Network Architecture

(b) Dilated Local Feature Aggregation

(c) Multi-level Hierarchical Feature Fusion

Fig. 1 Framework of the proposed method. *FC* fully connected layer, *RS* random sampling, *DNFE* dilated neighbor-hood feature extraction, *MLP* multilayer perception, *US* up sampling, *MHFF* multi-level hierarchical feature fusion, *DNE* dilated nearest neighbor encoding

embedding space by a 1D convolution and then the transformed features are fused with instance (semantic) features to facilitate instance (semantic) segmentation.

Fuzzy3DSeg [47] is proposed based on fuzzy mathematical methods to integrate the learning of the fuzzy neighborhood feature of each point for the fine-grained local feature missing problem. Both spatial information (coordinates) and other features (colors) are used for feature learning.

A fuzzy mechanism in spherical convolutional kernel is introduced for 3D point clouds, as well as a graph convolutional network (SegGCN) for semantic segmentation [48]. The fuzzy kernel will be robust to boundary effects in feature extraction since it avoids splits along the radial direction.

Our method

Problem statement

A point cloud is a set of 3D points, which can be represented as

$$\mathbf{P} = \{\boldsymbol{p}_i \| i = 1, \ldots, N_p\}, \tag{1}$$

where each point $\boldsymbol{p}_i$ represents a vector of its (x, y, z) coordinate in our work. The semantic segmentation of the point cloud is to predict $N_p \times N_c$ scores to indicate their semantic categories.

Overview of the proposed approach

The framework of the proposed network is shown in Fig. 1. The network follows the widely used encoding-decoding structure, and is based on the backbone of RandLA-Net [22]. First, several encoding layers which consist of random sampling (RS) layers and dilated local feature aggregation (DLFA) are used to learn the features of each sampled point. Then, the features of each level of down sampling are upsampled, and the features belonging to the same layer are concatenated. Finally, three full connection layers and one Dropout (DP) layer are used to predict the semantic tags of each point.

We will introduce the sampling strategy of point clouds, dilated local feature aggregation, multi-level hierarchical feature fusion, and point cloud data augmentation in the following.

Sampling of point clouds

Various sampling methods have been designed for large-scale point clouds to reduce the computational complexity. Due to the properties of point cloud data such as disorder, irregularity and large volume, it is necessary to find an efficient point sampling method. Farthest point sampling (FDS) is widely used in many classical methods, such as PointNet [20] and PointNet++ [21]. However it has very high computation complexity due to the calculation of the distance between each point. Inverse Density Importance Sampling (IDIS) selects the top K points according to the density of each point. Compared with FPS, IDIS has a great decrease of computation complexity, but it is still not suitable for large-scale point clouds.

Different from the above methods that need to preprocess the data, *Random Sampling (RS)* directly select K points from the original point clouds. It has a low computation complexity, so we use this method in the sampling of point clouds.

Dilated local feature aggregation

To extract local features of sampled point clouds, we designed the dilated local feature aggregation module, which consists of three components: dilated nearest neighbor encoding, attention pooling, and dilated residual block.

Dilated nearest neighbor encoding

As shown in Fig. 2, the input of the dilated nearest neighbor encoding is point cloud data. N is the number of points, the dimension of the point space coordinates (x, y, z) is 3, and d is the dimension of point feature f obtained by the previous network layer (fully connected layer).

Inspired by the dilated convolutional networks, we aim to increase the model's reception field for 3D point clouds. Based on the K-nearest neighbor (KNN) algorithm, we first find $2K$ neighborhood points of the ith point, and then 50% are randomly selected as key points for subsequent calculation. In other words, for each sampling point, the receptive field is expanded by twice so that the features of the obtained neighborhood points are more representative.

For the K points $\{p_i^1 ... p_i^k ... p_i^K\}$ of each center point p_i, we use an augmented matrix of their relative position and feature aggregation as their feature representation, which can be described as

$$l_i^k = \mathrm{MLP}(p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus \|p_i - p_i^k\|), \tag{2}$$

where l_i^k is the relative position encoding, $\oplus$ represents a concatenation operation, and $\|\cdot\|$ calculates the Euclidean distance. Then we augment the encoded relative point positions l_i^k of each neighboring point with its corresponding features f_i^k, so the augmented features can be described as

$$\widehat{F_i} = \{\widehat{f_i^1} ... \widehat{f_i^k} ... \widehat{f_i^K}\}. \tag{3}$$

Attention pooling

After obtaining features of neighboring point features $\widehat{F_i}$, we use attention pooling to aggregate a set of features. First, we use function $g()$ to calculate the attention score of each feature $\widehat{f_i^k}$, which acts as a mask and is a MLP with shared parameters. W represents weight parameters of MLP. Then the mask is formulated as

$$m_i^k = g(\widehat{f_i^k}, W). \tag{4}$$

Finally, the feature of point p_i can be calculated as

$$\tilde{f_i} = \sum_{k=1}^{K} (\widehat{f_i^k} \cdot m_i^k). \tag{5}$$

Dilated residual block

Due to the above two steps, the point cloud data is significantly down-sampled, and a lot of details will be lost. Therefore, it is necessary to expand the reception field of each point so that the geometric details of the input point cloud can be retained as much as possible. An extended residual block is composed of two dilated nearest neighbor encoding and attention pooling units.

After the first *Dilated Neighborhood Encoding* and *Attention Pooling* operation, its receiving field includes K neighboring points, and after the second operation, its receiving field is expanded to K^2 points.

Dilated point transformer

To verify the efficiency of the dilated nearest neighbor strategy, we designed another semantic segmentation framework for point clouds, named as dilated point transformer. The backbone is based on the work in [23], which is motivated by the success of the Transformer in natural language processing (NLP), and its core component is the self-attention mechanism. The self-attention operator is invariant to permutation and cardinality of input elements, so it is very suitable to process point clouds which are 'sets' embedded in 3D space with the properties of irregularity.

The point transformer layer can be formulated as

$$y_i = \sum_{p_j \in \mathcal{P}(i)} \sigma(\gamma(\varphi(p_i) - \psi(p_i) + \delta)) \odot (\alpha(p_j) + \delta) \tag{6}$$

where σ is a normalization function (e.g. *softmax*), γ is the attention vector, φ, ψ and α are pointwise feature transformation. δ is a position encoding function. The point transformer modules are depicted in Fig. 3 including point transformer block, feature encoding and decoding modules. The framework of the dilated point transformer is depicted in Fig. 4, which has a similar architecture with that in Fig. 1 but different in encoding. The subset $\mathcal{P}(i) \subseteq \mathcal{P}$ is a set of points in a local neighborhood of p_i. In our dilated point transformer, we changed the subset in a form of dilated nearest neighbor graph $\mathcal{P}(i) \rightarrow \widetilde{\mathcal{P}}(i)$.

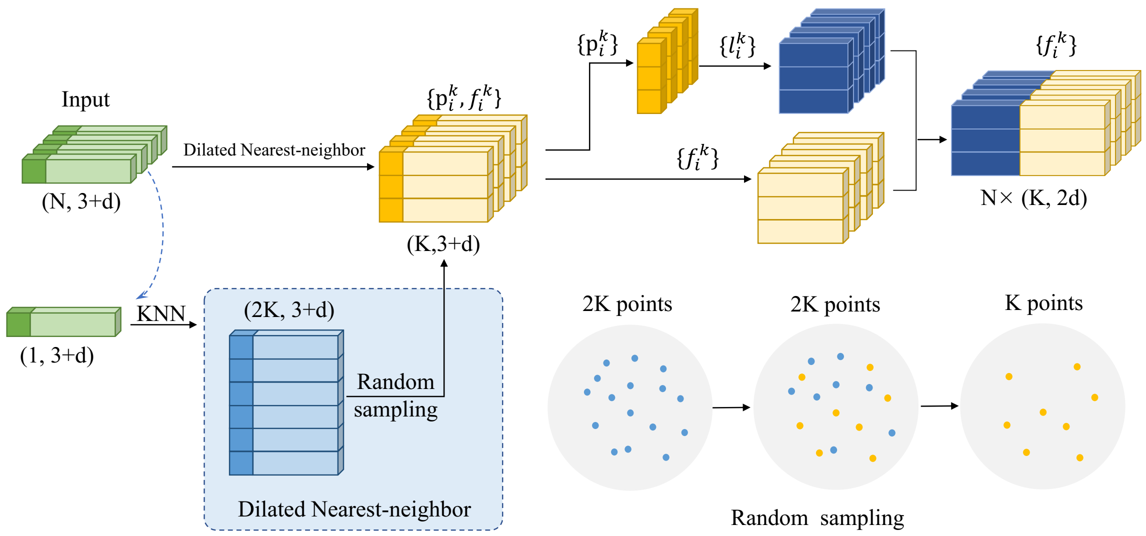
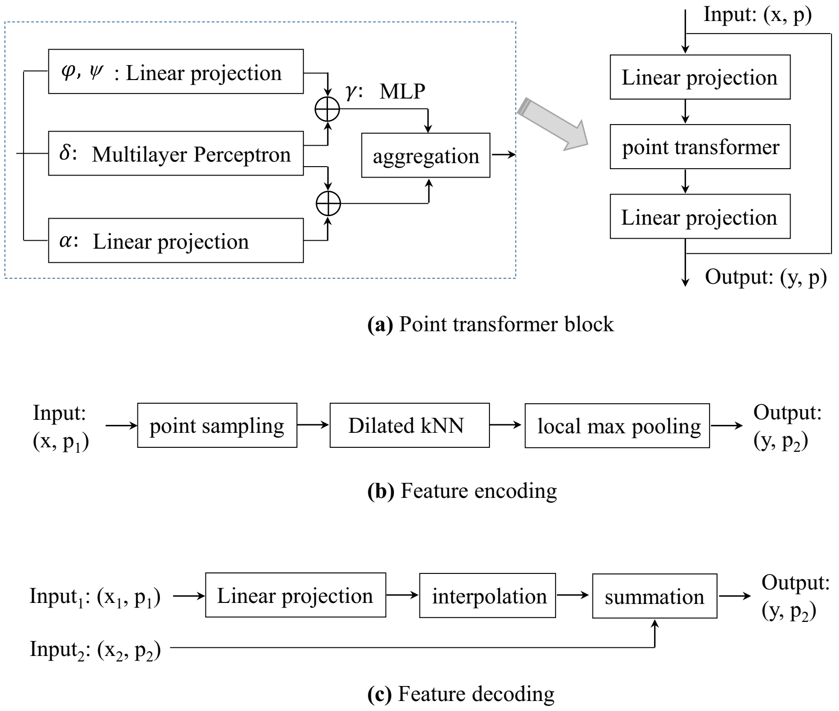


Fig. 2 Dilated nearest neighbor encoding

Fig. 3 Point transformer modules



(a) Point transformer block

(b) Feature encoding

(c) Feature decoding

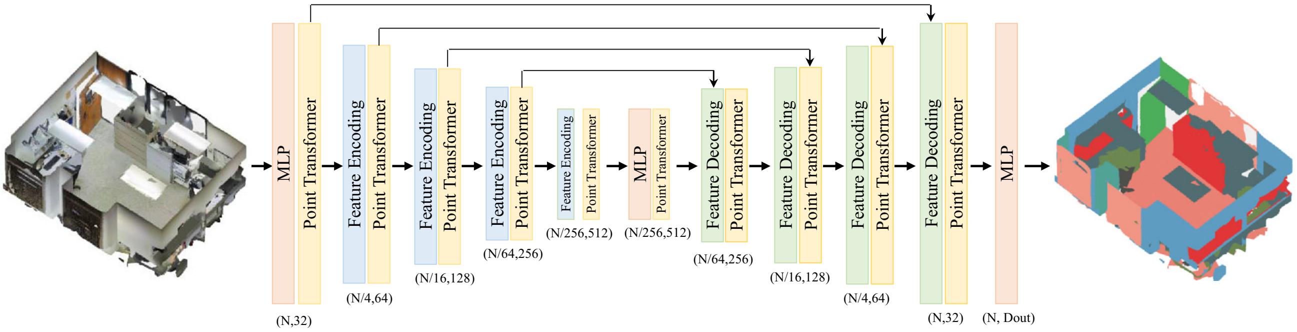


Fig. 4 Dilated point transformer framework

Experiments

Experiments will be introduced in this section to illustrate the efficiency of our proposed method. First, the experiment settings and datasets will be presented. Then we evaluate our method and compare with state-of-the-art methods.

Experiment settings

We use the point clouds data with semantic segmentation labels. The data is preprocessed and the sampling rate is 4% of the original data. The feature dimension is from $(N, 8)$ to $(N/128, 512)$ with a sub-sampling ratio of 4, where N is 40,960 in this experiment. The results of the network with four layers and five layers are tested, respectively. The network training environment is Ubuntu 16.04.3 with 2 NVIDIA Titan XP GPUs.

Datasets

In the experiments, we have used the benchmark **S3DIS** dataset [49], which is obtained by scanning 271 rooms in indoor areas of large buildings, and it consists of over 215 million points covering 6000 square meters. The scanned areas include various architectures including offices, conference rooms, restrooms, lobbies, stairways, and hallways. Twelve semantic elements have been labeled in the dataset including structural elements (ceiling, floor, wall, beam, column, window, and door), and some furniture (table, chair, sofa, bookcase, and board).

Experimental results

We have compared our method with the state-of-the-art methods including PointNet++ [21], PointCNN [41], DGCNN [50], 3P-RNN [51], SPG [43], JSNet [46], and RandLA-Net [22]. The comparison results are listed in Table 1. Two metrics have been used as criterion to evaluate the methods' performance, i.e. overall accuracy (OA) and mean IoU (mIoU). From Table 1, it can be seen that our proposed method has a better performance than others on both metrics. Compared with the baseline of RandLA-Net [22], our method has achieved an improvement of 1.7 and 1.6% for OA and mIOU, respectively. Compared with one most recent method, JSNet [46], our method performs slightly better on the overall accuracy, but has an improvement of 7.1% on the mean IOU.

Table 1 Quantitative results of different approaches on the S3DIS dataset

Methods	OA (%)	mIoU (%)
PointNet++, *NeurIPS2017* [21]	78.6	47.6
DGCNN, *NN2018* [50]	84.1	56.1
SPG, *CVPR2018* [43]	85.5	62.1
3P-RNN, *ECCV2018* [51]	86.9	56.3
PointCNN, *NeurIPS2018* [41]	88.1	65.4
$N_F F_W$, *ECCV2019* [44]	83.9	58.3
ASIS, *CVPR2019* [45]	86.2	59.3
RandLA-Net, *CVPR2020* [22]	87.2	67.2
JSNet, *AAAI2020* [46]	88.7	61.7
Ours	**88.9**	**68.8**

To further demonstrate the advantages of our method compared with RandLA-Net, more results have been given in Table 2. In this table, we have listed the mIOU of segmentation methods on 6 areas of S3DIS dataset in all the categories (e.g. ceiling, floor, wall, etc.). RandLA-Net-4 and RandLA-Net-5 represent results of RandLA-Net with four-layer and five-layer networks respectively, and Ours-4 represents the results of the four-layer network of our method. The results show that our method with four-layer networks performs better in most cases, even compared with RandLA-Net of 5-layers, since according to the mean value in Table 2, our method performs better on 5 areas, Area 1(2.6% ↑), Area 2(2.8% ↑), Area 4(1.2% ↑), Area 5(0.9% ↑), Area 6(1.2% ↑). Specifically, our method is better on 4 areas for the category of 'wall', 'window', 'chair', 'bookcase', 'board', and better on 3 areas for other categories.

We have also evaluated our designed dilated point transformer, and the results have been listed in Table 3, from which it can be seen that the dilated point transformer performs better on all the areas. This demonstrated the effectiveness of the dilated nearest neighboring encoding. Specifically, the dilated point transformer has obtained an increase of absolute 2.8% for Area 1, 4.7% for Area 2, 1.6% for Area 3, 0.3% for Area 4, 2.5% for Area 5, and 2.2% for Area 6. In our experiments, the dilated Point Transformer need about one day for training the network (24 epochs), and six minutes for inference one area of the S3DIS, at the same level with the RandLA-Net.

Some qualitative results of semantic segmentation have been given in Figs. 5 and 6, in which we also give the full RGB input point cloud and corresponding ground truth for illustration. Figures 5 and 6 show that our method achieves satisfactory semantic segmentation quality.

Conclusions

In this paper, we proposed a 3D dilated nearest neighbor encoding method, which proves to be efficient to leverage the semantic segmentation of large-scale point clouds. We have verified its effectiveness in two different frameworks. The first is based on random sampling and encoding–decoding structure. The second is based on the Point Transformer.

Table 2 Comparison of mIoU results between RandLA-Net and ours on six areas of S3DIS dataset

	Mean	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter
Area 1														
RandLA-Net-4	72.2	96.5	95.1	76.0	49.8	46.8	79.1	82.4	72.2	80.0	68.4	60.2	64.6	67.3
RandLA-Net-5	71.9	96.6	95.4	74.8	54.2	49.4	78.2	80.9	66.8	78.6	71.9	56.7	67.4	63.4
Ours-4	**74.8**	96.5	**95.5**	**78.7**	**70.4**	**53.7**	**80.0**	**83.4**	71.0	**81.7**	64.3	59.5	**70.9**	66.3
Area 2														
RandLA-Net-4	53.8	90.0	95.1	77.9	22.0	34.6	43.5	64.3	42.3	48.7	48.9	44.8	44.5	43.6
RandLA-Net-5	52.9	88.5	95.0	80.0	20.1	40.7	44.0	69.7	41.8	37.7	47.5	48.7	31.9	42.6
Ours-4	**56.6**	88.2	**95.3**	79.2	20.5	**41.7**	**44.4**	64.5	**47.0**	**66.0**	**54.6**	44.3	43.9	**46.5**
Area 3														
RandLA-Net-4	76.1	95.8	98.3	78.1	67.7	24.9	62.6	87.0	73.0	84.7	86.8	70.3	88.6	71.0
RandLA-Net-5	**77.1**	95.3	98.2	80.3	65.8	31.7	71.6	88.4	71.8	84.5	84.7	70.7	90.2	69.8
Ours-4	76.4	**96.0**	98.2	**80.5**	**68.9**	28.3	63.3	84.7	71.9	**85.5**	85.7	**70.9**	87.4	**71.7**
Area 4														
RandLA-Net-4	57.8	93.9	97.0	76.6	27.6	34.8	31.4	56.2	64.3	75.3	49.7	44.71	43.2	57.2
RandLA-Net-5	60.6	94.3	96.7	77.3	55.6	40.2	31.7	59.3	61.0	71.4	49.8	45.69	41.5	60.7
Ours-4	**61.8**	**95.5**	**97.6**	**78.6**	**58.8**	39.7	**32.8**	**62.4**	62.0	70.7	**52.1**	**48.99**	**45.5**	58.7
Area 5														
RandLA-Net-4	61.6	89.2	95.1	79.4	0	22.2	59.3	43.1	77.4	86.1	66.9	69.9	62.2	50.3
RandLA-Net-5	61.3	92.7	97.9	78.5	0	24.9	55.6	41.3	73.6	84.2	69.8	66.4	63.6	48.2
Ours-4	**62.5**	91.9	96.6	**79.9**	0	**26.5**	**61.5**	39.0	77.3	**87.3**	59.2	**71.3**	**68.3**	**53.4**
Area 6														
RandLA-Net-4	78.9	96.3	97.5	81.8	80.4	68.4	81.1	85.8	75.8	84.9	61.0	69.2	75.9	68.4
RandLA-Net-5	79.3	96.1	97.6	84.4	79.2	75.8	77.7	86.6	76.5	85.3	60.5	66.7	75.5	69.3
Ours-4	**80.5**	**96.5**	97.5	83.5	79.9	73.2	78.4	**87.1**	76.4	83.9	**69.4**	**74.8**	**76.7**	69.0

Table 3 Comparison of mIoU results between point-transformer and our dilated point-transformer on six areas of S3DIS dataset

	Mean	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter
Area 1														
Point-transformer[a]	74.26	98.35	98.72	92.37	57.08	13.68	94.24	85.87	82.32	86.31	67.56	42.47	66.84	79.58
Dilated point-transformer	**77.08**	98.28	98.44	87.54	58.28	45.28	92.64	84.07	85.92	85.98	64.38	56.41	66.62	78.25
Area 2														
Point-transformer	52.90	95.18	92.69	92.65	13.10	0.00	56.08	65.33	44.19	76.91	6.38	45.06	31.32	68.86
Dilated point-transformer	**57.61**	94.92	96.89	91.47	28.16	0.12	52.07	69.78	55.65	72.21	17.53	59.14	39.73	71.27
Area 3														
Point-transformer	71.05	97.99	99.44	89.34	88.91	0.00	51.79	87.37	81.53	76.10	41.78	52.26	85.34	71.78
Dilated point-transformer	**72.68**	98.13	99.67	87.32	89.70	14.34	66.42	88.25	86.86	76.43	44.11	63.84	64.01	65.75
Area 4														
Point-transformer	57.91	97.43	99.74	83.64	9.12	9.53	18.49	72.00	70.77	76.15	30.85	56.27	62.23	66.55
Dilated point-transformer	**58.23**	97.10	99.70	82.68	2.67	1.35	19.50	80.78	72.87	71.96	42.40	59.22	60.85	65.91
Area 5														
Point-transformer	58.36	96.49	99.19	93.53	0.00	3.39	43.47	33.76	78.56	87.58	29.69	60.24	60.77	72.00
Dilated point-transformer	**60.85**	95.75	99.56	92.13	0.00	3.64	42.99	55.20	73.63	86.91	43.34	67.68	58.58	71.69
Area 6														
Point-transformer	76.41	98.64	98.61	87.96	83.89	16.47	66.36	92.09	85.21	84.49	72.07	55.59	73.74	78.27
Dilated point-transformer	**78.62**	99.09	98.48	87.79	81.18	46.94	67.81	94.14	85.23	83.44	68.17	59.26	70.87	79.69

[a] Since the code of point-transformer is not available, the results are obtained by our simulation, and not as good as that in their paper

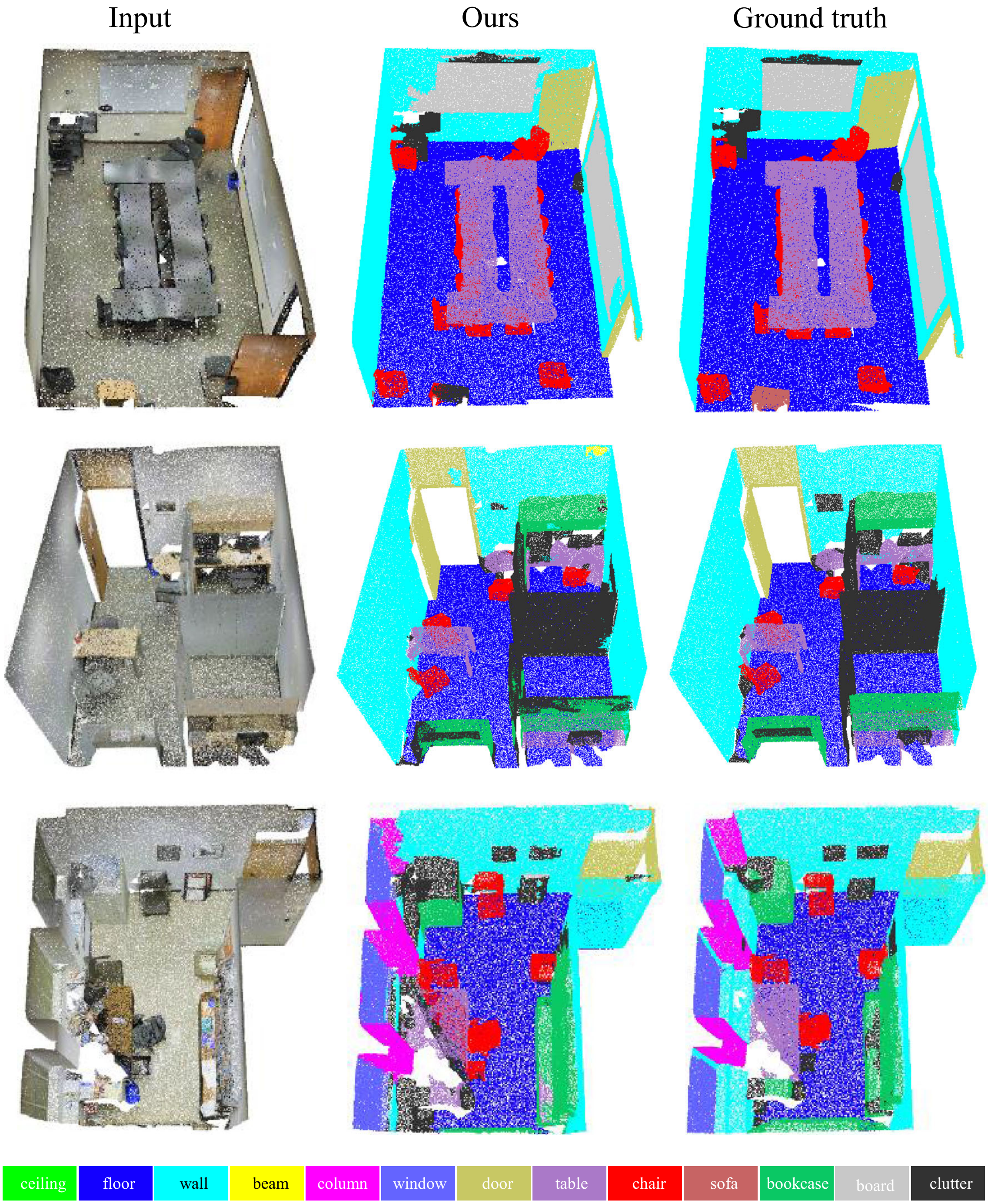


Fig. 5 Visualization of semantic segmentation results of test split on the S3DIS dataset. Left: full RGB input point cloud; middle: predicted labels; right: ground truth

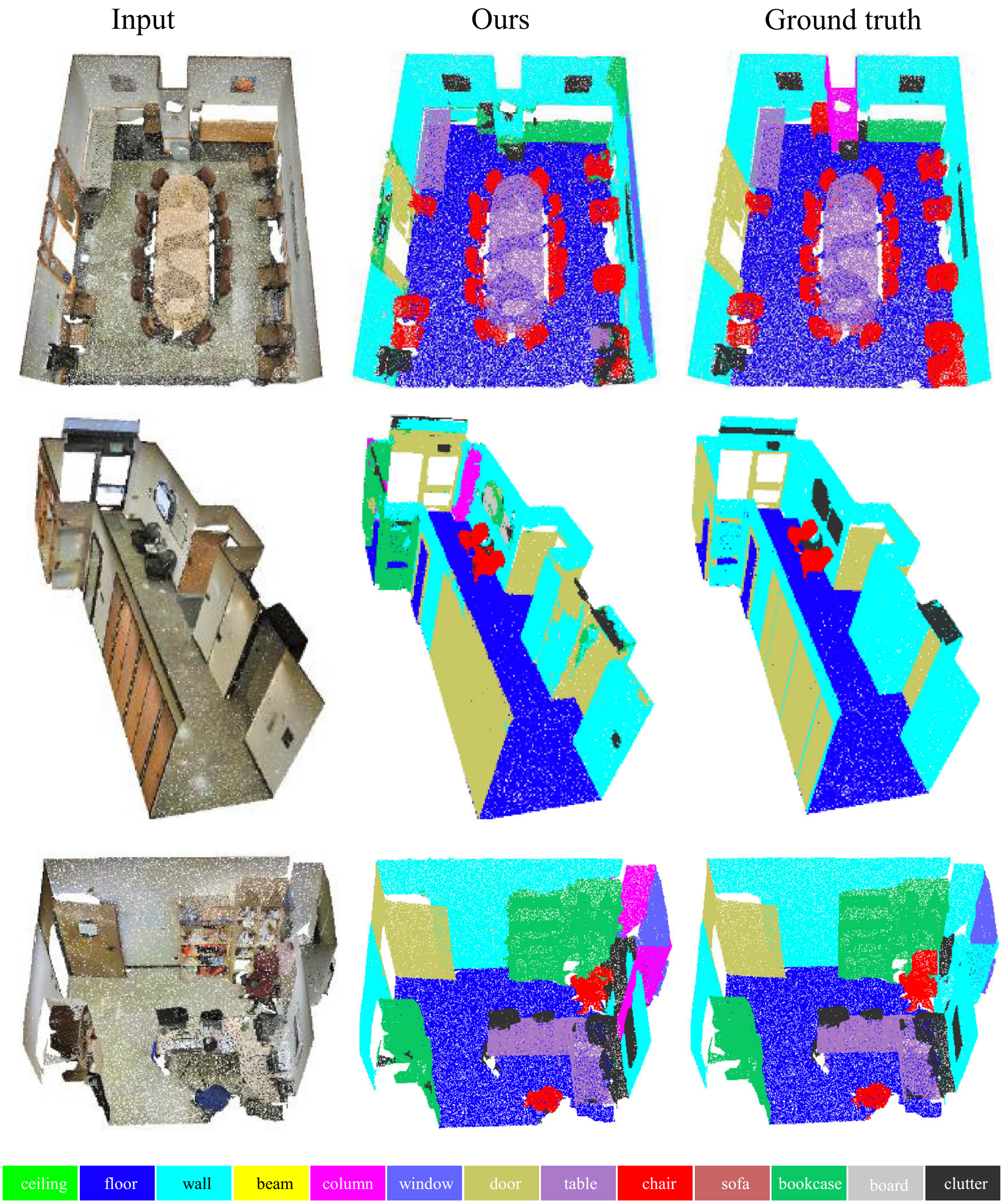


Fig. 6 Visualization of semantic segmentation results of test split on the S3DIS dataset. Left: full RGB input point cloud; middle: predicted labels; right: ground truth

Experiments on the benchmark dataset show that our model has achieved better performance than state-of-the-art methods.

Edge computing has been widely used in many applications for real-time processing of large-scale IOT big data, especially with the development of autonomous driving and robotics. Point clouds semantic segmentation is one important task in these areas. In the future, we will learn features of both point cloud data and RGB images so that we can utilize multi-modality information for prediction to further improve the performance.

Acknowledgements This work was supported in part by the National Key R&D Program of China (2018YFB1308000), the National Natural Science Foundation of China (61772508, U1713213, 61976143, 61775175), Guangdong Basic and Applied Basic Research Foundation (2021A1515012637), CAS Key Technology Talent Program, Guangdong Technology Program (2016B010108010, 2016B010125003, 2017B010110007), Shenzhen Engineering Laboratory for 3D Content Generating Technologies (No. [2017] 476), CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (2014DP173025), Guangdong–Hong Kong–Macao Joint Laboratory of Human–Machine Intelligence-Synergy Systems (2019B121205007).

Declarations

Conflict of interest All authors declare that there are no conflicts of interest.

References

1. Johnson-Roberson M, Bohg J, Björkman M, Kragic D (2010) Attention-based active 3D point cloud segmentation. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, Taipei, Taiwan, pp 1165–1170
2. Liu M (2016) Robotic online path planning on point cloud. IEEE Trans Cybern 46(5):1217–1228
3. Asif U, Bennamoun M, Sohel FA (2017) RGB-D object recognition and grasp detection using hierarchical cascaded forests. IEEE Trans Rob 33(3):547–564
4. Chen J, Cho YK, Kira Z (2019) Multi-view incremental segmentation of 3-D point clouds for mobile robots. InIEEE Robot Autom Lett 4(2):1240–1246
5. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition, Providence, RI, USA, pp 3354–3361
6. Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3D object detection network for autonomous driving. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6526–6534
7. Yang B, Luo W, Urtasun R (2018) PIXOR: real-time 3D object detection from point clouds. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, Salt Lake City, UT, USA, pp 7652–7660
8. Wang L, Fan X, Chen J, Cheng J, Tan J, Ma X (2020) 3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. Sustain Cities Soc 54:1–10
9. Zeng Y et al (2018) RT3D: real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving. IEEE Robot Autom Lett 3(4):3434–3440
10. Wang BH, Chao W, Wang Y, Hariharan B, Weinberger KQ, Campbell M (2019) LDLS: 3-D object segmentation through label diffusion from 2-D images. IEEE Robot Autom Lett 4(3):2902–2909
11. Sipiran I, Bustos B (2011) Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes. Vis Comput 27:963–976
12. Zhong Y (2009) Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: 2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops, Kyoto, Japan, pp 689–696
13. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE international conference on robotics and automation, Kobe, Japan, pp 3212–3217
14. Rusu RB, Bradski G, Thibaux R, Hsu J (2010) Fast 3D recognition and pose using the viewpoint feature histogram. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, Taipei, Taiwan, pp 2155–2162
15. Hackel T, Wegner JD, Schindler K (2016) Fast semantic segmentation of 3D point clouds with strongly varying density. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci 3(3):177–184
16. Tao D, Cheng J, Lin X, Yu J (2015) Local structure preserving discriminative projections for RGB-D sensor-based scene classification. Inf Sci 320:383–394
17. Bobkov D, Chen S, Jian R, Iqbal MZ, Steinbach E (2018) Noise-resistant deep learning for object classification in three-dimensional point clouds using a point pair descriptor. IEEE Robot Autom Lett 3(2):865–872
18. Maturana D, Scherer S (2015) VoxNet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), Hamburg, Germany, pp 922–928
19. Zhou Y, Tuzel O (2018) VoxelNet: end-to-end learning for point cloud based 3D object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4490–4499
20. Qi CR, Su H, Mo K, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 652–660
21. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems, pp 5099–5108
22. Hu Q, Yang B, Xie L, Rosa S, Guo Y, Wang Z, Trigoni N, Markham A (2020) RandLA-Net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 11108–11117
23. Hengshuang Z, Li J, Jiaya J, Philip T, Vladlen K (2020) Point transformer. arXiv:2012.09164 [cs.CV]
24. Rusu RB, Marton ZC, Blodow N, Beetz M (2008) Persistent point feature histograms for 3D point clouds. In: Proceedings of the International Conference on Intelligent Autonomous Systems, pp. 119–128
25. Behley J, Steinhage V, Cremers AB (2012) Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In: 2012 IEEE international conference on robotics and automation, pp 4391–4398
26. Jutzi B, Gross H (2009) Nearest neighbour classification on laser point clouds to gain object structures from buildings. Int Arch Photogramm Remote Sens Spat Inf Sci XXXVIII-1-4-7/W5
27. Fehr D, Cherian A, Sivalingam R, Nickolay S, Morellas V, Papanikolopoulos N (2012) Compact covariance descriptors in 3D point clouds for object recognition. In: 2012 IEEE international conference on robotics and automation, pp 1793–1798

28. Weinmann M, Urban S, Hinz S, Jutzi B, Mallet C (2015) Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Comput Graph 49:47–57
29. Li B, Zhang T, Xia T (2016) Vehicle detection from 3D lidar using fully convolutional network. In: Robotics science and systems
30. Su H, Maji S, Kalogerakis E, Learned-Miller E (2015) Multi-view convolutional neural networks for 3D shape recognition. In: IEEE international conference on computer vision (ICCV), pp 945–953
31. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) PointPillars: fast encoders for object detection from point clouds. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 12689–12697
32. Graham B, Engelcke M, Maaten LVD (2018) 3D semantic segmentation with submanifold sparse convolutional networks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, Salt Lake City, UT, pp 9224–9232
33. Meng H, Gao L, Lai Y, Manocha D (2019) VV-Net: voxel VAE net with group convolutions for point cloud segmentation. In: 2019 IEEE/CVF international conference on computer vision (ICCV), Seoul, Korea (South), pp 8499–8507
34. Huang Q, Wang W, Neumann U (2018) Recurrent slice networks for 3D segmentation of point clouds. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 2626–2635
35. Zhao H, Jiang L, Fu CW, Jia J (2019) Pointweb: enhancing local neighborhood features for point cloud processing. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5560–5568
36. Wang L, Huang Y, Hou Y, Zhang S, Shan J (2019) Graph attention convolution for point cloud semantic segmentation. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10288–10297
37. Chen C, Li G, Xu R, Chen T, Wang M, Lin L (2019) Clusternet: deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 4989–4997
38. Jiang L, Zhao H, Liu S, Shen X, Fu CW, Jia J (2019) Hierarchical point-edge interaction network for point cloud semantic segmentation. In: IEEE/CVF international conference on computer vision (ICCV), Seoul, Korea (South), pp 10432–10440
39. Wu W, Qi Z, Fuxin L (2019) Pointconv: deep convolutional networks on 3D point clouds. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9613–9622
40. Qi CR, Liu W, Wu C, Su H, Guibas LJ (2018) Frustum pointnets for 3D object detection from RGB-D data, In: 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 918–927
41. Li Y, Bu R, Sun M, Wu W, Di X, Chen B (2018) PointCNN: convolution on x-transformed points. In: Advances in neural information processing systems, pp 820–830
42. Shi S, Wang X, Li H (2019) Pointrcnn: 3D object proposal generation and detection from point cloud. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 770–779
43. Landrieu L, Simonovsky M (2018) Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4558–4567
44. Engelmann F, Kontogianni T, Schult J, Leibe B (2018) Know what your neighbors do: 3D semantic segmentation of point clouds. European conference on computer vision (ECCV), workshops. pp 395–409
45. Wang X, Liu S, Shen X, Shen C, Jia J (2019) Associatively segmenting instances and semantics in point clouds. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), Long Beach, CA, USA, pp 4091–4100
46. Zhao L, Tao W (2020) JSNet: joint instance and semantic segmentation of 3D point clouds. In: The thirty-fourth AAAI conference on artificial intelligence, pp 12951–12958
47. Zhong M, Li C, Liu L, Wen J, Ma J, Yu X (2020) Fuzzy neighborhood learning for deep 3-D segmentation of point cloud. IEEE Trans Fuzzy Syst 28(12):3181–3192
48. Lei H, Akhtar N, Mian A (2020) SegGCN: efficient 3D point cloud segmentation with fuzzy spherical kernel. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), Seattle, WA, USA, pp 11608–11617
49. Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3D semantic parsing of large-scale indoor spaces. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, pp 1534–1543
50. Phan AV, Nguyen ML, Nguyen YLH, Bui LT (2018) DGCNN: a convolutional neural network over large-scale labeled graphs. Neural Netw 108:533–543
51. Ye X, Li J, Huang H, Du L, Zhang X (2018) 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 403–417

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.