3D Semantic Segmentation for Large-Scale Scene Understanding

Kiran Akadas and Shankar Gangisetty

KLE Technological University, Hubballi, India
akadask@gmail.com, shankar@kletech.ac.in

Abstract. 3D semantic segmentation is one of the most challenging events in the robotic vision tasks for detection and identification of various objects in a scene. In this paper, we solve the task of semantic segmentation to classify and assign every point in the scene with an associated label. We propose a lightweight semantic segmentation network for large-scale point clouds which consists of grid subsampling, dilated convolutions, and Gaussian error linear unit activation for gaining better performance. The dilated convolutions increase the receptive field while reducing the number of parameters, making proposed network faster and computationally more efficient with reduced number of parameters. Additionally, we use conditional random field as post processing method to boost the performance of proposed semantic segmentation network. We perform an exhaustive quantitative analysis of the proposed network on SOTA datasets, namely, SHREC 2020 street scenes dataset [1], S3DIS [2] and SemanticKITTI [3]. We show that proposed semantic segmentation network performs effectively and efficiently compared to SOTA methods.

1 Introduction

3D point clouds have attracted a lot of interest in recent years because of wide range of applications and the ability to preserve spatial information of objects and sceneries which makes point clouds efficient in capturing detailed information. With the advent of mobile 3D scanners and devices such as drones and mobile phones capable of capturing 3D information, there has been tremendous increase in the point cloud data availability.

Semantic segmentation is one of the most challenging tasks that assigns semantic labels to every point that belongs to the objects of interests. With the recent advances in deep learning, 3D semantic segmentation has become a very powerful tool with profound applicability in autonomous systems (mobile robots, autonomous driving), scene understanding, augmented reality and vegetation monitoring. The primary task of 3D semantic segmentation is to understand the constituents or the different objects present in the scene before performing further analysis. The raw point clouds acquired by 3D scanning devices are either sparse, irregularly sampled, unstructured, and unordered which makes segmentation task challenging. We thus need an efficient solution to accurately segment 3D point clouds.

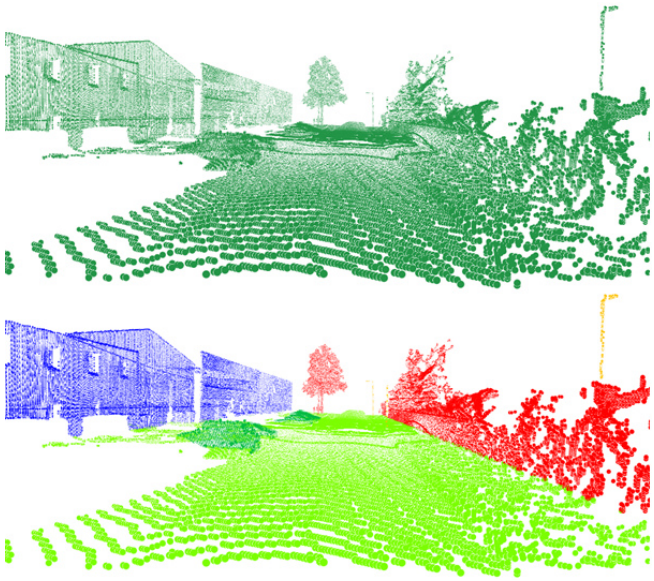

Fig. 1. 3D point cloud semantic segmentation of SHREC 2020 street scenes dataset [1]. Top row: Input point cloud scene Bottom row: Segmented point cloud scene using proposed approach

Initial works on segmentation [4–7] employed 2D convolutional neural networks (CNN) on range images. Next, voxel-based methods [8–12] harnessed the effectiveness of 3D convolutions for segmentation. However, the voxel segmentation is computationally expensive and the sparsity of point clouds renders the methods to be non-viable for large-scale point clouds. Additionally, the conversion of point clouds to voxels results in loss of intrinsic shape details. Finally, with the introduction of PointNet [13] came a new era of point-wise feature learning considering raw point clouds as input for 3D segmentation. The point-based works like [14–19] have shown good performance in processing and segmentation. However, it's very hard for these learning-based methods to train on large-scale point clouds. In this paper, we propose a 3D point cloud semantic segmentation network for large-scale street scenes by providing effective sampling and reduced parameters, that is computationally efficient and processes the point clouds directly. In this work, we extend our proposed GRanD-Net architecture [1] from Shape Retrieval Challenge (SHREC) 2020 track on 3D point cloud semantic segmentation for street scenes.

Inspired by RandLA-Net [20], our proposed network in this paper extends and improves RandLA-Net [20]. In the proposed network, we effectively use random sampling without lose of information while sampling through the use of local feature aggregation. We then adopt dilated/atrous convolutions in the network that helps to increase the receptive field without the loss of resolution and helps

in semantic segmentation. The dilated convolutions are proven to be effective on 2D semantic segmentation tasks [21]. Thus, we extend dilated convolutions use on 3D point clouds. We also use GeLU as the activation function to better learn complex functions. And optionally, use conditional random field (CRF) in post processing to boost the performance. We provide experimental analysis to demonstrate the efficiency and efficacy on various methods and datasets. The results of segmented regions like *building*, *car*, *ground*, *pole*, and *vegetation* from the large-scale outdoor scene [1] using proposed network shown in Figure 1.

The remainder of the paper is organized as follows: Section 2 briefly presents related work. The proposed semantic segmentation of 3D point clouds network is described in Section 3. Experiments and results are presented and discussed in Section 4. Finally, the paper is concluded in Section 5.

2 Related Works

With the recent availability of large scale point cloud datasets [1, 3, 22–24] has motivated researcher keenness in 3D semantic segmentation tasks. Based on the input data type, the semantic segmentation tasks can be categorized to range image-based, voxel-based and point-based methods.

2.1 Range image-based methods

Many works use the well-defined 2D CNNs for segmentation of range images which are a 2D representation of 3D point clouds. In SqueezeSeg [4], authors introduced FireModule and FireDeconv for efficient segmentation based on LIDAR point clouds represented as spherical range-image [25]. In Pointseg [5] and RIU-Net [6], authors used 2D CNNs for semantic segmentation on range images. In FuseSeg [7], authors extended SqueezeSeg [4] to fuse RGB features from ImageNet CNN [26]. However, the conversion of 3D objects to 2D image leads to the loss of contextual information. In 3D-MiniNet [27], a recent work, authors used a projection module to extract features from the spherical projection of point cloud for efficient segmentation of 2D image that is then projected back to 3D.

2.2 Voxel-based methods

The point clouds can be converted to 3D grids by voxelization and 3D CNNs are used to process these grids as in [10–12]. In 3D-FCNN [8], authors proposed to predict voxel-level semantic labels. The accuracy of segmentation depends on the resolution of voxels. In SEGCLOUD [9], authors extended 3D-FCNN [8] to obtain fine-grained results using tri-linear interpolation to point-wise labels and applied a fully connected CRF (FC-CRF), for training them jointly. These methods obtain good results for semantic segmentation, but are computationally expensive and thus, are not used for large scale point clouds.

2.3 Point-based methods

PointNet [13], was one of the first works to efficiently adopt convolutions for 3D point clouds using max-pooling to achieve permutation-invariance. PointNet introduced input transformations and feature transformations which ensured that the point clouds are invariant to geometric transformations. But, PointNet did not capture the local features which are essential for segmentation i.e., only global features are learned. In PointNet++ [14], authors extended PointNet [13] to capture local structure information by applying PointNet hierarchically that greatly improved segmentation scores. In PointCNN [15], authors proposed a hierarchical deep learning framework and introduced a X-Conv layer that elevated the points to a higher representation with rich features which are then propagated to pointwise feature using X-DeConv layers. But, an understanding of the operations of X-Conv are not well established. Although, all of these methods were extended to perform segmentation, they did not scale well for large-scale scene point clouds.

The graph-based methods like [16, 17] made use of graph-based structure and graph CNN for processing point clouds and assign semantic labels. In [28], authors introduced PointGCR, a plug and play module which uses graph convolutions to obtain a global contextual dependency. In ASIS [29] and JSIS3D [30], authors proposed the association of instance segmentation and semantic segmentation with partnerships to jointly solve segmentation tasks. The JSIS3D additionally used a multi-valued CRF as a post processing method. In [16], authors presented a method to enrich the point representations and introduced a graph PointNet module (GPM) to update features within local structures, and spatial-wise and channel-wise attention strategy to exploit the global information to obtain pointwise labeling. Recently, in RandLA-Net [20], authors proposed to directly predict per point semantics for large point clouds efficiently with a local feature aggregation module. The dilated residual blocks, local spatial encoding and attentive pooling are used to generate informative feature vectors. However, the method is computationally expensive. In our proposed network, we extend RandLA-Net [20] and address these issues to provide a solution by improving the semantic segmentation performance.

3 3D Point Cloud Semantic Segmentation Network

In this section, we describe our network for 3D point cloud semantic segmentation for large scale scenes. The proposed network, shown in Figure 2, is composed of the dilated residual blocks as the basic building blocks coupled with random sampling to down-sample the point cloud. We process raw point clouds as inputs, perform grid-subsampling on the large point clouds to bring it to a uniform size. We process these N points gradually by down-sampling the points using random-sampling while preserving the essential features required for segmentation through the use of dilated residual blocks that enhance the feature representation. We use dilated convolutions and GeLU as the activation function to process the point clouds in our dilated residual blocks. There are N labels

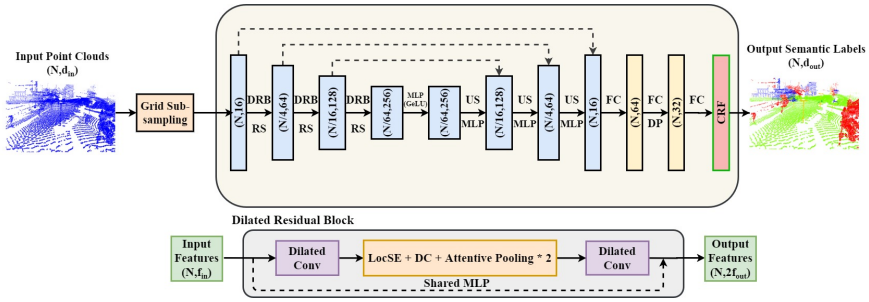

Fig. 2. The proposed 3D point cloud semantic segmentation network for large-scale scenes. DRB: Dilated Residual Block, RS: Random Sampling, US: Up-Sampling, FC: Fully Connected Layer, LocSE: Local Spatial Encoding, DC: Dilated Convolutions DP: Dropout, CRF: Conditional Random Field

obtained as output after up-sampling the results, one for each point indicating the category to which the point belongs.

3.1 Data Preparation

Let $P = \{p_i | i = 1, 2, ..., N\}$ be the point cloud with N points. To efficiently process data of large-scale 3D scenes, we sub-sample the point clouds each with N points using grid-subsampling from KPConv [18] to bring it to a uniform size. In order to get back the original number of semantic labels from the predictions, we index projections for up-sampling the point clouds, one for each point indicating the category to which the point belongs. The training dataset is augmented by scaling and rotation.

3.2 Data Loading

To load the data in batches, we generate the data flow for each batch. For a given batch size n and the steps in each epoch s, $(n \times s)$ point clouds are reserved for each epoch. To avoid ordered learning by the proposed network, we feed the data randomly. The k-Nearest Neighbours (NN) algorithm is used with a pre-defined set of k neighbours being selected of all the sub-sampled points. If the sampled points are less than the given pre-defined k points, we pick the points with replacement. To prepare a batch of point clouds, we generate the neighbour indices for every point in a point cloud. These are used to get the relative point features. We then randomly sample 25% of points to be reduced in the next phase while down-sampling and simultaneously track the indices for up-sampling. The pooling indices are obtained using k-NN search for every sampled point.

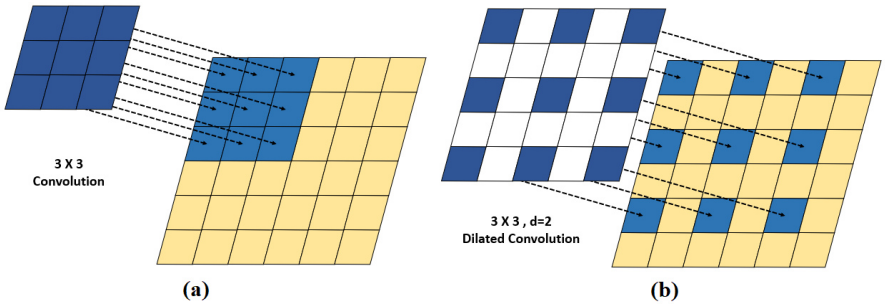

Fig. 3. Convolutions: (a) 3×3 regular convolution used in CNNs and (b) 3×3 dilated convolution with $d = 2$ covering an area of 5×5 used in our proposed model results in reduced parameters.

3.3 Background for Model Building

In our next step of model building, dilated convolutions and GeLU activation function are essential components. Let us have a brief overview before discussing on adaptation in proposed network.

Dilated Convolutions The convolutional layer in deep learning architecture extracts low-level features in the initial layers and higher-level features deeper in the network. Dilated convolutions [31], shown in Figure 3, brings an extra parameter d that controls the receptive area of the kernel which convolves around the input with a gap difference shown Figure 3(b). The parameter d determines the size of the hole in the kernel. Thus, without increasing the number of parameters, it increases the receptive area of the kernel over the input making the convolutions more efficient and faster.

Consider $F : Z^2 \rightarrow R$ to be a discrete function (i.e., a region shown in Figure 3) and $\Omega_r = [-r, r]^2 \cap Z^2$. If $k : \Omega_r \rightarrow R$ is a kernel of size $(2r+1)^2$ (3×3 shown in Figure 3), then the dilated convolution on an input point p with a dilation factor of d is given as,

$$(F * k)(p) = \sum_{s+dt=q} F(s)k(t) \tag{1}$$

The number of parameters remain same as we do not increase the size of units in kernel that are used to calculate the result. In our proposed network, dilated convolutions help process areas with redundant data faster while preserving the required features. Compared to normal convolutions, dilated convolutions are faster, efficient and better at semantic segmentation.

GeLU The Gaussian error linear unit (GeLU) [32] is a nonlinear high-performing neural network activation function [33]. GeLU considers not only the sign as in

ReLU [34], but the magnitude of the input and considered to be more effective than other activation functions. Instead of multiplying the input by one or zero deterministically, the GeLU determines the value stochastically based on the input and multiplies with the input. This helps in better learning of complex functions. If point p is the input then, GeLU is given as,

$$GELU(p) = 0.5p(1 + \tanh[\sqrt{\frac{2}{\pi}}(p + 0.044715p^3)]) \tag{2}$$

The GeLU activation is applied over the outputs of the dilated convolutions and others in the proposed network.

3.4 Model Building

We train the sampled point clouds over several batches of data. The loaded point clouds of dimensions (N, d_{in}), where N is the number of points in the point cloud and d_{in} is the number of features associated with each point p in the point cloud, are processed using the dilated residual blocks (DRBs) shown in Figure 2. Each of the DRB includes multiple units of local spatial encoding (LocSE) and attentive pooling stacks. The DRBs are connected through skip-connections as proposed in RandLA-Net [20]. Unlike the convolutions used in RandLA-Net [20], we use dilated convolutions [31] to implement the DRB in order to increase the receptivity of the filters without affecting the resolution and gain better efficiency. The dilated convolutions incorporate multi-scale features, essential for semantic segmentation. These dilated convolutions make our network faster and more efficient since we are increasing the area of filter coverage without increasing the parameters and affecting the feature learning, thus reducing the number of convolutions.

The LocSE in DRB uses the centre points and their k-NN neighbours to encode the point cloud using relative positional information. At each step, we apply RS with DRB using the points we loaded earlier to reduce the size of input point cloud to 25%. The attentive pooling is used as a replacement to max-pooling in order to compute the attention score for every feature which is further aggregated to avoid loss of information and learn important local information. We use GeLU [32] as the activation layer in our proposed network for better learning of non-linear features. The GeLU prevents strong negative activations which may affect the model. The curvature and non-monotonicity of GeLU is used to learn complex functions much better compared to ReLU and leaky ReLU [34]. The output of the stacked DRBs is up-sampled and passed through multi-layer perceptrons (MLP) followed by fully connected layers. The use of skip-connections and MLP while up-sampling ensures that the labeling is accurate. Our network follows an all-inclusive up-sampling approach that refines labels gradually unlike simple interpolation, which would result in a single label for a group of points ignoring the demarcation of classes. The output of interpolation can be refined by the use of a post-processing technique (i.e., CRF).

3.5 Conditional Random Field

The CRF refines the labels based on the position of the input point and the neighboring point's label. An energy function is defined for label assignment which acts as a cost function. The minimization of the energy function leads to refinement of labels and increase in the accuracy.

Finally, the predicted semantic labels for every point are obtained as the output of the network with dimensions (N, d_{out}), where d_{out} is the number of labels (i.e., classes) in the dataset.

4 Experiments and Results

4.1 Implementation Details

We train our proposed network using the Adam optimizer with a learning rate of 0.01 and a decay rate of 0.05. A grid size of 0.06 is fixed for grid-subsampling while training and we select $k = 16$ NN to be queried. To train, we sample a fixed number (N) of $65,536$ points from each point cloud as the input and use a batch size of 4 with 500 steps per epoch. We train our proposed network for 50 epochs with a train-validation split of $3:1$. A four layered network is used with feature sizes of $16, 64, 128$, and 256. The network is trained on a 15 GB CPU with a single NVIDIA Tesla T4 GPU. The code of our network is released here: https://github.com/KiranAkadas/GRanDNet.

4.2 Datasets

The proposed network is experimented on three datasets, namely, SHREC 2020 street scenes [1], S3DIS [2] and SemanticKITTI [3].

SHREC 2020 Street Scenes [1] dataset: The dataset contains 80 large-scale 3D point clouds for street scene which are captured by a LIDAR sensor mounted on a car and manually labeled using open source software Cloud Compare [35]. Each point cloud represents a street scene and contains a group of

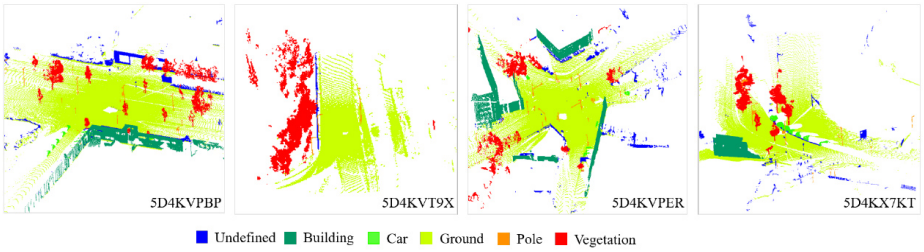

Fig. 4. Sample point clouds of SHREC 2020 Street Scenes [1] dataset.

objects labeled into 5 meaningful classes and an extra *undefined* class which is not used for evaluating the results. The distribution of points in each of the 6 classes, namely, *undefined*, *building*, *car*, *ground*, *pole*, and *vegetation* are $8.37\%, 17.05\%, 2.81\%, 54.64\%, 0.47\%$, and 16.64%. The 80 point clouds are randomly divided into training and testing sets with 60:20. The average number of points in training and testing dataset is approximately 2 to 4 million points per point cloud. We train our proposed network for the 5 classes leaving the *undefined* class. The sample point clouds of the dataset are visualized in Figure 4.

S3DIS [2] and SemanticKITTI [3] datasets: The S3DIS [2] is a large-scale 3D scene dataset of indoor spaces. The dataset contains scans of 271 rooms, each provided as a separate point cloud belonging to 6 large areas. The points are classified into 13 categories of object. The SemanticKITTI dataset [3] consists of 21 sequences with 00 to 10 as training set with the sequence 08 used as validation set, and 11 to 21 as test set. There are a total of 23,201 full 3D scans for training and 20,351 scans for testing. A total of 19 categories are considered to evaluate the dataset.

4.3 Evaluation

We adopt the evaluation criteria that have been widely applied in 3D semantic segmentation tasks, Overall Accuracy (OA) and mean Intersection over Union (mIoU). Generally, OA reports the percent of points in the data set which are correctly classified. And mIoU is the average of per-class IoU. The IoU of class i is defined as,

$$IoU_i = \frac{TP_i}{GT_i + Pred_i + TP_i} \tag{3}$$

where TP_i, GT_i, $Pred_i$ denote the correctly classified number of points, the ground truth point number, and predicted point number for class i, respectively.

4.4 Results and Discussion

In this section, we evaluate the overall efficiency of our proposed network on large-scale point clouds for semantic segmentation. We compare the performance of proposed network with RandLA-Net [20]. We are the first to evaluate the performance on the SHREC 2020 street scenes [1] dataset. For a fair comparison, we use the same parameters and same number of input points in our network and RandLA-Net and train for 5 classes. The best performing model is frozen with a mIoU of 84.11%. The frozen model is used to predict segments for the 20 test point clouds that contain a total of $7,27,53,747$ points. The resulting mIoU is 86.4% with an OA of 97.83% for 5 classes shown in Table 1. Our network achieves superior performance on four of the classes, except *pole*. We observe that the resulting IoU of *ground*, *vegetation*, *building*, and *car* classes are segmented accurately as the dataset distribution in these classes is high and learnt better. We also performed a qualitative analysis of our network on test set shown in

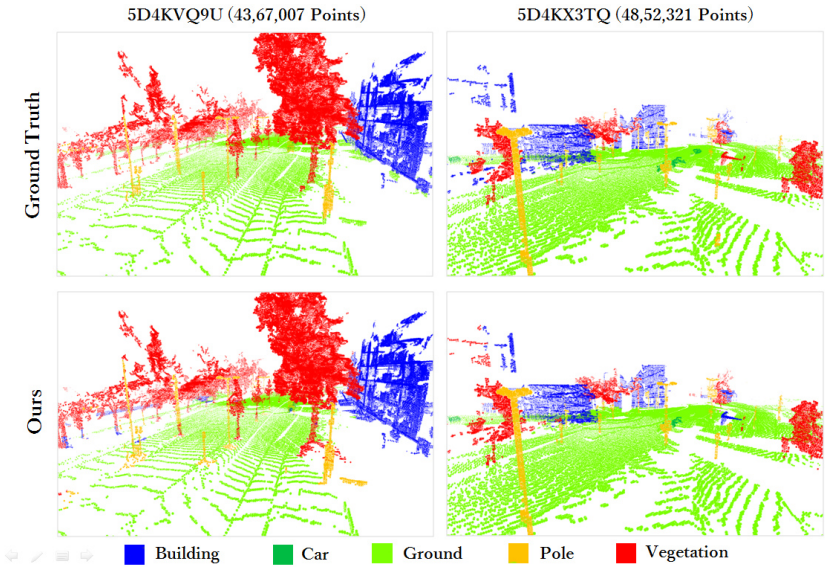

Fig. 5. Qualitative results of our proposed network on the test set (5D4KVQ9U and 5D4KX3TQ point clouds) of Street Scenes [1] dataset.

Figure 5. Visual inspection shows that our network performance is good and close to ground truth. We also compared our results with RandLA-Net for the 5 classes as shown in Table 1. We observe that our network outperforms RandLA-Net by a good margin.

As the *pole* class IoU is low, to estimate out the misclassification we plotted the confusion matrix for the proposed network shown in Figure 6. Based on the confusion matrix and visual inspection, we observe that few instances of the *pole* class are mislabeled as *building* shown in Figure 7, due to their proximity to the *building* points and also the insufficiency of the *pole* training points.

4.5 Time and Space Complexity

We compute the average time taken to complete an epoch during the training of the models and additionally calculate the total number of trainable parameters

Table 1. Quantitative results: The OA, mIoU, and IoU for each of the five classes in the test data on our proposed network.

Methods	OA (%)	mIoU (%)	Building	Car	Ground	Pole	Vegetation
RandLA-Net [20]	95.92	84.3	91.76	80.72	96.31	59.80	93.23
Ours	**97.83**	**86.40**	**93.66**	**83.92**	**98.10**	**61.79**	**94.55**

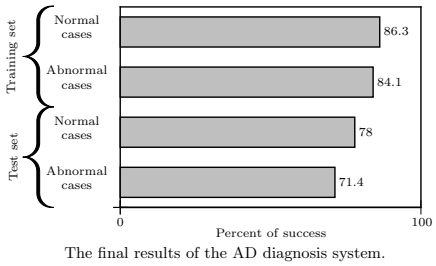

Fig. 6. Confusion matrix for 5 classes of Street Scenes dataset [1] using our model.

Table 2. Time and space complexity of our network and RandLA-Net [20] on Street Scenes [1] dataset.

	Average training time per epoch (seconds)	# Trainable parameters ('M' stands for Million)
RandLA-Net[20]	18,070	1.24
Ours	**15,960**	**0.99**

to estimate the memory consumption shown in Table 2. We observe that there is a significant decrease in the number of trainable parameters and the average training time in our network. Unlike RandLA-Net, the use of dilated convolutions in our network helps us increase the receptive area across every input resulting in faster training and lesser number of trainable parameters.

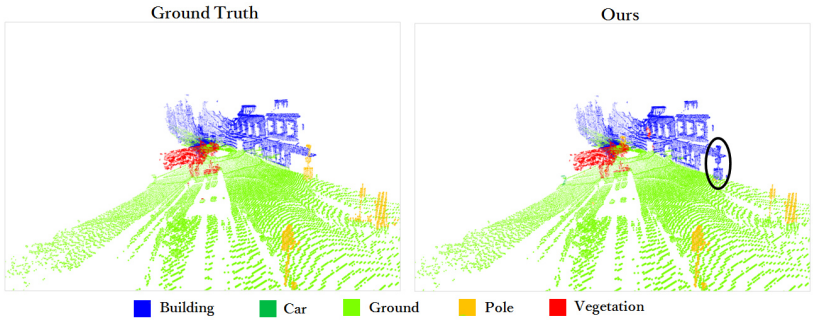

Fig. 7. Qualitative results of our proposed network on the test dataset of Street Scenes [1]. The black circle shows the *pole* mislabeled as *building* class.

4.6 Comparison with other datasets

In this section, we compare the results of our network with existing segmentation methods on other datasets.

S3DIS [2] The proposed network achieves comparably better performance to the SOTA methods on S3DIS [2] dataset shown in Table 3. Most of the SOTA methods are computationally expensive and operate on blocks of point clouds. Our network takes the entire large-scale scene as input and processes the output labels for every point in a single pass while being more efficient and faster. The qualitative results are shown in Figure 8.

SemanticKITTI [3] The quantitative results of our network compared to SOTA methods shown in Table 4. We observe that our network achieves the best mIoU over other SOTA methods on SemanticKITTI [3] dataset.

Table 3. Quantitative results of our network and SOTA methods on S3DIS [2] dataset.

	OA(%)	mAcc(%)	mIoU(%)	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet [13]	78.6	66.2	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
RSNet [36]	-	66.5	56.5	92.5	92.8	78.6	32.8	34.4	51.6	68.1	59.7	60.1	16.4	50.2	44.9	52.0
3P-RNN [37]	86.9	-	56.3	92.9	93.8	73.1	42.5	25.9	47.6	59.2	60.4	66.7	24.8	57.0	36.7	51.6
SPG [19]	86.4	73.0	62.1	89.9	95.1	76.4	62.8	47.1	55.3	68.4	**73.5**	69.2	63.2	45.9	8.7	52.9
PointCNN [15]	88.1	75.6	65.4	**94.8**	97.3	75.8	63.3	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
PointWeb [38]	87.3	76.2	66.7	93.5	94.2	80.8	52.4	41.3	64.9	68.1	71.4	67.1	50.3	62.7	62.2	58.5
ShellNet [39]	87.1	-	66.8	90.2	93.6	79.9	60.4	44.1	64.9	52.9	71.6	**84.7**	53.8	64.6	48.6	59.4
KPConv [18]	-	79.1	70.6	93.6	92.4	**83.1**	**63.9**	**54.3**	**66.1**	**76.6**	57.8	64.0	**69.3**	**74.9**	61.3	60.3
RandLA-Net [20]	88.0	82.0	70.0	93.1	96.1	80.6	62.4	48.0	64.4	69.4	69.4	76.4	60.0	64.2	65.9	60.1
Ours	**88.3**	**82.3**	**71.0**	94.5	**97.7**	80.7	59.8	51.5	64.8	69.9	70.4	75.4	64.2	66.1	**67.7**	**60.5**

Table 4. Quantitative results of our network and SOTA methods on SemanticKITTI [3] dataset.

Methods	Size	mIoU(%)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
PointNet [13]		14.6	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPG [19]		17.4	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [40]	50K pts	18.4	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [14]		20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
TangentConv [41]		40.9	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5
SqueezeSeg [4]		29.5	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5
SqueezeSegV2 [42]		39.7	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
DarkNet21Seg [3]		47.4	91.4	74.0	57.0	26.4	81.9	85.4	18.6	26.2	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0
DarkNet53Seg [3]		49.9	**91.8**	74.6	64.8	**27.9**	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
RangeNet53++ [43]	64*2048	52.2	**91.8**	**75.2**	**65.0**	27.8	87.4	91.4	25.7	25.7	**34.4**	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9
LatticeNet [44]	pixels	52.2	88.8	73.8	64.6	25.6	86.9	88.6	43.3	12.0	20.8	24.8	76.4	57.9	54.7	34.2	39.9	**60.9**	55.2	41.5	42.7
SalsaNext [45]		54.5	90.9	74.0	58.1	27.8	87.9	90.9	21.7	36.4	29.5	19.9	81.8	61.7	66.3	**52.0**	**52.7**	16.0	58.2	**51.7**	**58.0**
SqueezeSegV3 [46]		54.5	90.9	74.0	58.1	27.8	87.9	90.9	21.7	36.4	29.5	19.9	81.8	61.7	66.3	**52.0**	**52.7**	16.0	58.2	**51.7**	**58.0**
RandLA-Net [20]	50K pts	55.9	90.5	74.0	61.8	24.5	89.7	94.2	43.9	**47.4**	32.2	39.1	**83.8**	63.6	**68.6**	48.4	47.4	9.4	60.4	51.0	50.7
Ours		**56.0**	90.6	74.0	61.4	24.1	**89.8**	**94.5**	**44.6**	30.9	29.6	**40.3**	83.2	**63.9**	**68.6**	48.7	47.8	9.9	**60.7**	51.5	50.0

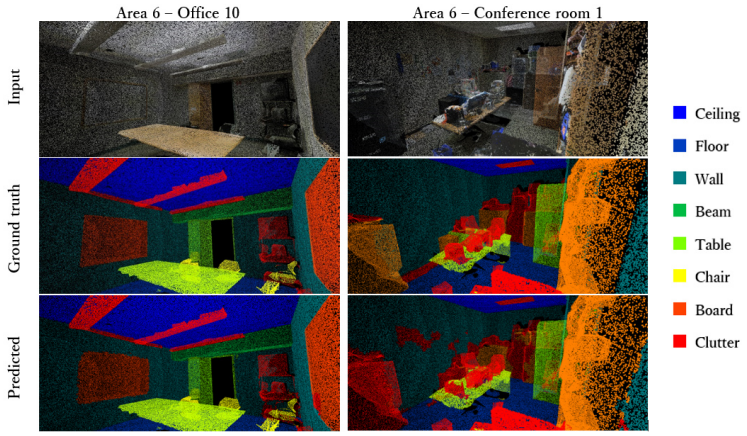

Fig. 8. Qualitative results of our network on the test set of S3DIS [2] dataset.

Table 5. Comparison of results for our model with and without CRF on Street Scenes dataset [1].

Methods	OA (%)	mIoU (%)	Building	Car	Ground	Pole	Vegetation
Ours (without CRF)	97.83	86.41	93.66	83.92	98.10	61.79	94.55
Ours (with CRF)	97.91	86.53	93.78	84.04	98.21	61.93	94.67

4.7 Proposed Network with CRF

A CRF explicitly designed for point clouds is used after we get the pointwise labels from our network. The parameters for the CRF are selected using grid search to obtain best set of parameters. The comparison results of our proposed network with and without CRF on the street scenes [1] dataset are shown in Table 5. We observe that there is a incremental boost in the classwise IoU performance of all 5 classes. This increase in the IoU is attributed to the correction of mislabeled points near the demarcation of two classes. The CRF based post-processing takes around 30 seconds per point cloud (with approximately 3 million points) and is hence very efficient. Considering the scale of the dataset, we state that our CRF is able to refine large set of points and gives a performance which is much better.

4.8 Ablation Study

To verify the effectiveness of dilated convolutions and the GeLU activation layer in DRBs, we conduct the ablation studies on street scenes [1] dataset. Using normal 2D convolution in proposed network instead of dilated convolutions within the DRBs gave an mIoU of 87.8% which is slightly better but on S3DIS we

get an mIoU of 68.3%, which is 3% less. Moreover, the network without dilated convolution takes 20% more time and space. Further, when the GeLU activation layer is replaced with it's predecessor leaky ReLU in the DRBs and also in the layers preceding the encoding layer, we obtain a lower mIoU of 86.1%. Thus, we observe that our proposed network architecture is effective and efficient for 3D point cloud semantic segmentation of large-scale scene understanding.

5 Conclusions and Future Work

In this paper, we proposed a 3D semantic segmentation network for assigning pointwise labels to large-scale 3D scenes. We use dilated convolutions as an essential unit to our building blocks of DRBs which coupled with random sampling unlike other sampling strategies helps our network to reduce the computational cost and preserve important features. We used GeLU as our activation function to learn complex functions. We also used an optional post processing module, CRF that helps refine labels assigned to the points at the boundaries of different classes. The resulting mIoU of our proposed network is 86.41% with an OA of 97.83% for 5 classes on street scenes dataset [1]. Additionally, our network achieves superior performance compared to SOTA methods on other large scale point cloud datasets, namely, S3DIS [2] and SemanticKITTI [3]. In the future, we plan to extend our network to perform instance and hierarchical semantic segmentation for scene understanding.

Acknowledgement

This research work is partly supported (DST/ICPS/IHDS/2018) under the Indian Heritage in Digital Space (IHDS) of Interdisciplinary Cyber Physical Systems (ICPS) Programme of the Department of Science and Technology (DST), Government of India.

References

1. Ku, T., Veltkamp, R.C., Boom, B., Duque-Arias, D., Velasco-Forero, S., Deschaud, J.E., Goulette, F., Marcotegui, B., Ortega, S., Trujillo, A., Suárez, J.P., Santana, J.M., Ramírez, C., Akadas, K., Gangisetty, S.: Shrec 2020: 3d point cloud semantic segmentation for street scenes. Computers & Graphics **93** (2020) 13 – 24
2. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. CoRR **abs/1702.01105** (2017)
3. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. IEEE ICCV (2019) 9296–9306
4. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. IEEE ICRA (2018) 1887–1893

5. Wang, Y., Shi, T., Yun, P., Tai, L., Liu, M.: Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. CoRR **abs/1807.06288** (2018)
6. Biasutti, P., Bugeau, A., Aujol, J.F., Brédif, M.: Riu-net: Embarrassingly simple semantic segmentation of 3d lidar point cloud. ArXiv **abs/1905.08748** (2019)
7. Krispel, G., Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Fuseseg: Lidar point cloud segmentation fusing multi-modal data. IEEE WACV (2020) 1863–1872
8. Huang, J., You, S.: Point cloud labeling using 3d convolutional neural network. ICPR (2016) 2670–2675
9. Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J., Savarese, S.: Segcloud: Semantic segmentation of 3d point clouds. 3DV (2017) 537–547
10. Choy, C.B., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. IEEE CVPR (2019) 3070–3079
11. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. IEEE CVPR (2018) 9224–9232
12. Meng, H.Y., Gao, L., Lai, Y.K., Manocha, D.: Vv-net: Voxel vae net with group convolutions for point cloud segmentation. IEEE ICCV (2019) 8499–8507
13. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. IEEE CVPR (2017) 77–85
14. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. (2017) 5099–5108
15. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: NeurIPS. (2018) 828–838
16. Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J.: Graph attention convolution for point cloud semantic segmentation. In: IEEE CVPR. (2019) 10296–10305
17. Wang, C., Samari, B., Siddiqi, K.: Local spectral graph convolution for point set feature learning. In: ECCV (4). Volume 11208 of LNCS., Springer (2018) 56–71
18. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. IEEE ICCV (2019) 6410–6419
19. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. IEEE CVPR (2018) 4558–4567
20. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. IEEE CVPR (2020) 11105–11114
21. Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., Hikosaka, S.: Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In: IEEE WACV. (2018) 1442–1450
22. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: Semantic3d.net: A new large-scale point cloud classification benchmark. CoRR **abs/1704.03847** (2017)
23. Roynard, X., Deschaud, J., Goulette, F.: Paris-lille-3d: A point cloud dataset for urban scene segmentation and classification. In: CVPR Workshops. (2018) 2027–2030
24. Zolanvari, S.M.I., Ruano, S., Rana, A., Cummins, A., da Silva, R.E., Rahbar, M., Smolic, A.: Dublincity: Annotated lidar point cloud and its applications. In: BMVC, BMVA Press (2019) 44
25. Biasutti, P., Aujol, J.F., Brédif, M., Bugeau, A.: Range-image: Incorporating sensor topology for lidar point cloud processing. Photogrammetric Engineering and Remote Sensing **84** (2018) 367–375
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60** (2017) 84–90

27. Alonso, I., Riazuelo, L., Montesano, L., Murillo, A.C.: 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. (2020)
28. Ma, Y., Guo, Y., Liu, H., Lei, Y., Wen, G.: Global context reasoning for semantic segmentation of 3d point clouds. In: IEEE WACV. (2020) 2920–2929
29. Wang, X., Liu, S., Shen, X., Shen, C., Jia, J.: Associatively segmenting instances and semantics in point clouds. IEEE CVPR (2019) 4091–4100
30. Pham, Q.H., Nguyen, T., Hua, B.S., Roig, G., Yeung, S.K.: Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: IEEE CVPR. (2019) 8827–8836
31. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (Poster). (2016)
32. Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR **abs/1606.08415** (2016)
33. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR **abs/1312.4400** (2014)
34. Agarap, A.F.: Deep learning using rectified linear units (relu). CoRR **abs/1803.08375** (2018)
35. CloudCompare: 3d point cloud and mesh processing software open source project (2020)
36. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3d segmentation of point clouds. IEEE CVPR (2018) 2626–2635
37. Ye, X., Li, J., Huang, H., Du, L., Zhang, X.: 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In: ECCV (7). Volume 11211 of LNCS., Springer (2018) 415–430
38. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: Pointweb: Enhancing local neighborhood features for point cloud processing. IEEE CVPR (2019) 5560–5568
39. Zhang, Z., Hua, B.S., Yeung, S.K.: Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. IEEE ICCV (2019) 1607–1616
40. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. IEEE CVPR (2018) 2530–2539
41. Tatarchenko, M., Park, J., Koltun, V., Zhou, Q.Y.: Tangent convolutions for dense prediction in 3d. IEEE CVPR (2018) 3887–3896
42. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: ICRA. (2019) 4376–4382
43. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet ++: Fast and accurate lidar semantic segmentation. IEEE IROS (2019) 4213–4220
44. Rosu, R.A., Schütt, P., Quenzel, J., Behnke, S.: Latticenet: Fast point cloud segmentation using permutohedral lattices. CoRR **abs/1912.05905** (2019)
45. Cortinhal, T., Tzelepis, G., Aksoy, E.E.: Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving (2020)
46. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. CoRR **abs/2004.01803** (2020)