INVESTIGATION OF POINTNET FOR SEMANTIC SEGMENTATION OF LARGE-SCALE OUTDOOR POINT CLOUDS

A. Nurunnabi[1] *, F. N. Teferle[1], J. Li[2], R. C. Lindenbergh[3], S. Parvaz[1]

[1] Geodesy and Geospatial Engineering, Faculty of Science, Technology and Medicine, University of Luxembourg, 6, rue Richard Codenhove-Kalergi, L-1359 Luxembourg − (abdul.nurunnabi, norman.teferle)@uni.lu, shahoriar.parvaz.001@student.uni.lu
[2] Geography and Environmental Management, University of Waterloo, Waterloo ON N2L 3G1,Canada − junli@uwaterloo.ca
[3] Geosciences and Remote Sensing, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN Delft, The Netherlands − r.c.lindenbergh@tudelft.nl

KEY WORDS: Airborne Laser Scanning, City Modelling, Deep Learning, LiDAR, Machine Learning, Scene Understanding

ABSTRACT:

Semantic segmentation of point clouds is indispensable for 3D scene understanding. Point clouds have credibility for capturing geometry of objects including shape, size, and orientation. Deep learning (DL) has been recognized as the most successful approach for image semantic segmentation. Applied to point clouds, performance of the many DL algorithms degrades, because point clouds are often sparse and have irregular data format. As a result, point clouds are regularly first transformed into voxel grids or image collections. PointNet was the first promising algorithm that feeds point clouds directly into the DL architecture. Although PointNet achieved remarkable performance on indoor point clouds, its performance has not been extensively studied in large-scale outdoor point clouds. So far, we know, no study on large-scale aerial point clouds investigates the sensitivity of the hyper-parameters used in the PointNet. This paper evaluates PointNet's performance for semantic segmentation through three large-scale Airborne Laser Scanning (ALS) point clouds of urban environments. Reported results show that PointNet has potential in large-scale outdoor scene semantic segmentation. A remarkable limitation of PointNet is that it does not consider local structure induced by the metric space made by its local neighbors. Experiments exhibit PointNet is expressively sensitive to the hyper-parameters like batch-size, block partition and the number of points in a block. For an ALS dataset, we get significant difference between overall accuracies of 67.5% and 72.8%, for the block sizes of 5m×5m and 10m×10m, respectively. Results also discover that the performance of PointNet depends on the selection of input vectors.

1. INTRODUCTION

Pointwise classification, also known as semantic segmentation is a higher-level task in object recognition: detection, classification and segmentation. This task, which is crucial for scene understanding and 3D data visualization, has many applications, which include 3D city modeling (Agoub et al., 2019), building information modeling (Romero-Jaren et al., 2021), location-based services (Li et al., 2019), infrastructure monitoring (Li et al., 2021), road inventory (Ma et al., 2020), augmented reality (Ko and Lee, 2020), autonomous driving (Gong et al., 2020), and urban planning (Guo et al., 2018).

Point clouds have been widely used for semantic segmentation. They can be generated by using remote sensing technologies such as LiDAR (Light Detection and Ranging), SAR (Synthetic Aperture Radar), and as an intermediary product of photogrammetry or structure from motion. A major source of large-scale outdoor point clouds is Airborne Laser Scanning (ALS) using LiDAR systems. Point clouds are usually unorganized, sparse, incomplete, noisy, and occluded. They also have inconsistent point density, irregular data format, arbitrary surface shape, sharp features and contaminated with outliers (Nurunnabi et al., 2014, 2015). Due to the presence of a large number of object types; ranging from small scale spatial neighborhoods (e.g., wires on utility power poles) to large scale spatial neighborhoods (e.g., buildings), and many overlapping and closely related complex shape objects; automatic point labelling (i.e., semantic segmentation) of each point in ALS point clouds is challenging in urban environment. However, point clouds can provide a more precise 3D representation than images as they have the ability to capture 3D geometric details (e.g., shape, size and orientation) of objects. They are also of unified structures that can avoid the combinatorial irregularities and complexities of meshes, and thus are easier to learn from (Qi et al., 2017a).

Many methods have been developed over the years for semantic segmentation in relevant research fields such as computer vision (Guo et al., 2020; Minaee et al., 2021), photogrammetry (Nurunnabi et al., 2016; Yu et al., 2021), and machine learning (ML; Chehata et al., 2009; Zhang et al., 2013). In recent years, deep learning (DL) has been recognized as the most powerful approach in object detection, classification, and segmentation (Qi et al., 2017a, b; Thomas et al., 2019; Hu et al., 2020; Boulch, 2020; Guo et al., 2020; Jing et al., 2021). Convolutional neural networks (CNNs) (LeCun et al., 1989) have achieved unprecedented success in structured data (e.g., image) analysis (Krizhevsky et al., 2012). However, using CNNs for point clouds processing is challenging because CNNs typically require regular data in order to perform weight sharing. Because of the unstructured nature of point clouds, many researchers transform such point clouds into regular formats like voxel grids or multiple image collections that can loss data information (Qi et al., 2017a). They also transform the raw data into useful features, and develop feature-based DL methods (Zhang et al., 2018; Nurunnabi et al., 2021). Some methods reformulate the CNN architecture to consider the unstructured nature of point clouds (Boulch, 2020). Recently, Qi et al. (2017a) developed a

* Corresponding author

revolutionary approach, PointNet, that directly feeds point clouds into a DL architecture. It has gained remarkable success for objects classification, part segmentation and semantic segmentation of indoor point clouds. The authors (Qi et al., 2017a) showed on par or better performance than state-of the-art multi-view and volumetric approaches (Su et al., 2015; Qi et al., 2016). PointNet is simple, computationally efficient, and has strong 3D representation ability. Many other successful networks relying on PointNet's basic architecture have since been proposed (Qi et al., 2017b; Achlioptas et al., 2018; Sun et al., 2019). However, people frequently use this algorithm for classification and semantic segmentation because of its simplicity and fast computational capability. Inappropriately, many researchers use PointNet with values for the hyper-parameters unchanged from the original paper without investigating the sensitivity to the hyper-parameters.

This paper evaluates prospects and limitations of PointNet for semantic segmentation in large-scale ALS point clouds in urban environment (i.e., outdoor). We estimate the sensitivity of the hyper-parameters such as: (i) block size, (ii) the number of points in a block, and (iii) the batch-size used in the PointNet algorithm. Additionally, we study the effects of the different combinations of the input vectors, we performed PointNet only with 3D (x, y, and z coordinates), 3D adding with their normalized values (x_n, y_n, z_n), and the other input vectors like intensity (I), return number (RN), scan angle (SA), and color (R, G, B) information.

The remainder of the paper includes a brief discussion on PointNet algorithm together with relevant point-based methods in Section 2. Experiments in Section 3 demonstrate the sensitivity of the PointNet algorithm on required hyper-parameters, and underlying data nature through three large-scale ALS datasets in urban environments followed by a short general discussion and concluding remarks in Section 4.

2. REVISIT POINTNET AND RELATED DL NETS

A point cloud is an un-ordered set of vectors, whose basic structure can be represented as a set of 3D points $\{p_i = (x_i, y_i, z_i) | i = 1,2, \dots, n)\}$, where x, y, z are the three coordinates of the points. Additional characteristics such as color, intensity (I), and RN may be available.

PointNet architecture (Fig. 1) processes each point independently, learns per-point features using shared multilayer perceptrons (MLPs) layers followed by a global max-pooling layer. This architecture combines three basic modules: (i) a symmetry function, (ii) local and global information aggregation, and (iii) a joint alignment network. The max-pooling is a symmetric function that is used to make a model invariant to input permutation. The max-pooling layer aggregates information from each point, and extracts global shape features. Zaheer et al. (2017) demonstrated that summing up all representations and applying nonlinear transformations help to achieve permutation invariance. The second module is crucial for semantic segmentation. After computing the global point cloud feature, this module feeds it back to the per point feature by concatenating the global feature with each of the point features. This new point feature holds both the local and global information, hence the network implies both local geometry and global semantics. The function of the third module is to make the semantic labelling invariant to certain geometric (e.g., rigid) transformation. A solution of semantic labelling invariant to geometry is to make an alignment of all inputs to a canonical

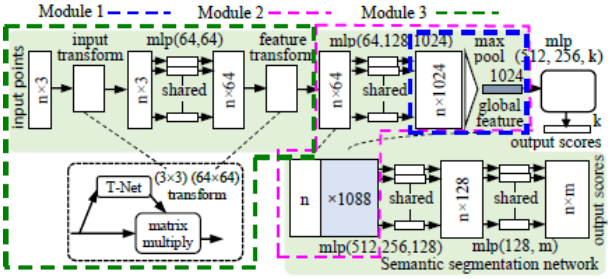

Figure 1. PointNet architecture, mlp stands for multilayer perceptrons, numbers in brackets are layer sizes; m and k are the number of classes. Figure curtesy (Qi et al., 2017a).

space before feature extraction. Jaderberg et al. (2015) developed a spatial transformer network to align 2D images through sampling interpretation. As each point in PointNet is transformed independently, the input format is easy to apply for the affine transformation. A data-dependent spatial transformer network, which is a mini network called T-Net; is included in the basic net-structure to canonicalize (predict affine transformation matrix) the data before processing. This T-Net is directly applied to the points' coordinates, and regresses to a 3×3 matrix. Another T-Net on point features is included to align features from different input point clouds. This T-Net is the same as the first one except that its function is to regress to a 64×64 output matrix. Adding T-Nets in the network can improve performance of the network (Qi et al., 2017a). The final fully connected layers combine these learnt optimal values into the global descriptor before being used to predict the points' labels.

PointNet ignores local spatial relationships in the data that makes it computationally efficient, but limits its performance when it comes to understand fine-grained patterns and when generalizing complex scenes (Qi et al., 2017b). Qi et al. (2017b) developed PointNet++, which extracts local features from local neighborhoods. Local features are grouped to make larger units, and then processed to get higher level features. PointNet is chosen in PointNet++ as a local feature learner. Point-based networks that have been developed to learn local structure, can be categorized into four groups: neighboring feature pooling (Qi et al., 2017b; Zhao et al., 2019), kernel-based convolution (Su et al., 2018; Thomas et al., 2019), graph-message passing (Wang et al., 2019; Kang and li, 2019), and attention-based aggregation (Yang et al., 2020; Zhang et al., 2019). These methods produce good semantic segmentation results, but most of them are limited to small datasets of around 4k points or 1m× 1m blocks with 4,096 points, and cannot be generalized directly to larger datasets (Hu et al., 2020). A handful of methods have been developed to process large-scale point clouds (Landrieu and Simonovsky, 2018; Rethage et al., 2018; Chen et al., 2019; Boulch, 2020; Han et al., 2021), but most of them are computationally expensive due mainly to their data pre-processing requirements.

3. EXPERIMENTS AND EVALUATION

We conducted several experiments, and investigated various issues that influence PointNet for semantic segmentation in large-scale point clouds. More specifically, we evaluated the effects of point density variation, block size, batch size, the number of points in a block, and input vectors through three sets of outdoor ALS data including two ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark datasets. We perform the PointNet algorithm on a computer with an NVIDIA GeForce RTX 2080 Super with Max-Q graphics card,

64 GB RAM, Intel® Core™ i7-10875H CPU @ 2.30GHz. To this end, we evaluate the following performance metrics: Intersection over Union (IoU), mean IoU (mIoU), F_1-score (F_1), mean F_1 (mF_1) and the Overall Accuracy (OA). These evaluation metrics are defined as follows:

$$\mathrm{IoU}_i = \frac{C_{ii}}{C_{ii}+\sum_{j\neq i} C_{ij}+\sum_{k\neq i} C_{ki}}, \quad (1)$$

$$\mathrm{F}_1 - \mathrm{score} = 2 \times \frac{P \times R}{P+R} = \frac{C_{ii}}{C_{ii}+\frac{1}{2}(\sum_{j\neq i} C_{ij}+\sum_{k\neq i} C_{ki})}, \quad (2)$$

$$\mathrm{OA} = \frac{\sum_{i=1}^{N} C_{ii}}{\sum_{j=1}^{N}\sum_{k=1}^{N} C_{jk}}, \quad (3)$$

where C_{ii} (true positive, TP) is the number of points from ground-truth class *i* identified as *i*th class, C_{ij} (false negative, FN) is the number of points from ground-truth class *i* but wrongly identified as *j*th class, C_{ki} (false positive, FP) is the number of points wrongly identified as ground-truth class *i*, but are from *k*th class, Precision, *P* = [TP/(TP+FP)] and Recall, *R* = [TP/(TP+FN)]. F_1 and OA do not balance different class frequencies giving higher impact to larger classes (Hackel et al., 2017), hence to compensate the influence of different class frequencies, additionally, we report IoU and mIoU.

3.1 Experiment 1 (Vaihingen data)

In the first experiment, we consider the ISPRS benchmark open access dataset (Niemeyer et al., 2014) used for 3D semantic labelling context. This was collected over the city of Vaihingen, Germany using a Leica ALS50 system. This dataset is known as the Vaihingen dataset. Scanning height for the dataset was 500m with field of view 45°. The dataset is split into a training set (Fig. 3) and a test set (Fig. 4a) consisting of 753,876 and 411,722 points, respectively. The training set is mostly residential, with isolated house and high-rise buildings, and covers 399m×421m area. The test set covers an area of 389m×419m within the city center, and contains dense and complex buildings.

Vaihingen data have an average point density of 4 points/m^2, each point has coordinates (*x*, *y*, *z*), I, RN, and the number of returns. The points are labelled as power line (PL), low vegetation (LV), impervious surface (IS), car, fence, roof, facade, shrub and tree. Necessary parameters such as number of hidden layers, and optimizers (*Adam*; Kingma and Ba et al., 2017) with learning rate of 0.001, and momentum 0.9 are fixed as they are defined in the original paper (Qi et al., 2017a). In PointNet, block size for the indoor datasets was fixed with 1m×1m of 4,096 points. Bar diagrams of Fig. 2 show significant disparity among the number of points within the groups. To balance among the nine classes, using stratified splits, we sampled 20% points of the training set as the validation set that is needed to assess the performance of the model during training. To see the effects of different point densities, we prepared training and validation data of three different block sizes: 5m×5m, 10m×10m, and 15m×15m with an overlap of 1m. Overlapping can increase the quantity of data and robustness. We then sampled 2,048 points from each block. Each point within the blocks is characterized by a 9D vector of *x*, *y*, *z*, I, RN, the height-above-ground (z_h) and normalized *x*, *y*, *z* values (x_n, y_n, z_n). The z_h values are computed using the LAStools (Isenburg, 2014) software. Batch normalization (Ioffe and Szegedy, 2015) is used for all layers with the ReLU (Nair and Hinton, 2010) activation function, and dropout layers are only used for the last MLP. To see the effects of batch size, we evaluated PointNet with batch sizes of 24, 32 and 36 at the time of training, and trained the model with 50 epochs. At the end of each epoch the model computes the OA for the evaluation (validation) set, and finally the best model with the highest OA is used to label the test data. We calculated the performance metrics: F_1, mF_1, IoU, mIoU, and OA for the test set. The results are shown in Table 1, and Fig. 4b.

3.1.1 Sensitivity with block size: The highest OA (72.8%) with mF_1=46.0% and mIoU=34.7% is achieved when the block size is 10m×10m, sampled 2,048 points from each block, and batch size 32. Although, the results (OA, mF_1 and mIoU) of block size 10m×10m with batch size 32 seem better than the results of the block sizes of 5m×5m and 15m×15m with the same batch size, the results of all the individual classes do not follow the same pattern. For example, for the class car, F_1 score of block size 5m×5m having 2,048 points in a block, and batch size of 32 is 52.8%, which is higher than the F_1 score 39.7% for block size of 10m×10m having 2,048 points in each block and batch size 32.

3.1.2 Sensitivity with a number of points belong to a class: We see that the classes with more training points dominate the classification accuracy. Fig. 2 shows that for both the training and test datasets the classes LV, IS, roof and tree contain expressively more points than other classes like car and fence. The classes with more training points are labelled significantly more accurate than the classes having a smaller number of points. For example, with the block size 10m×10m and batch size 32, IS points are labelled more accurately with 87.5% and 77.7% of F_1 score and IoU, respectively, whereas fence points are identified with F_1 score of 17.4% and IoU of 9.4%.

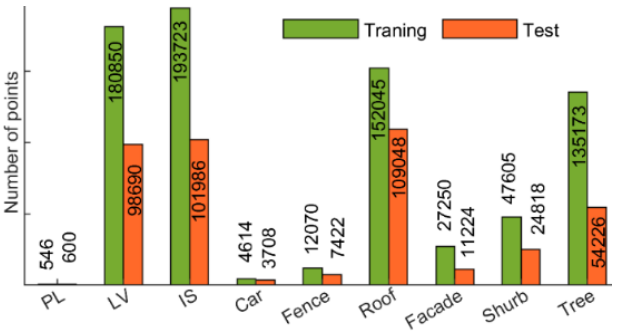

Figure 2. Bar diagrams of the point distribution of the Vaihingen training and test datasets.

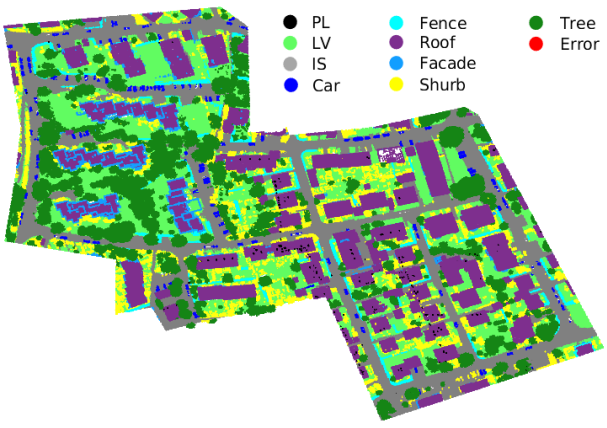

Figure 3. Vaihingen training dataset with ground-truth labels in different colors.

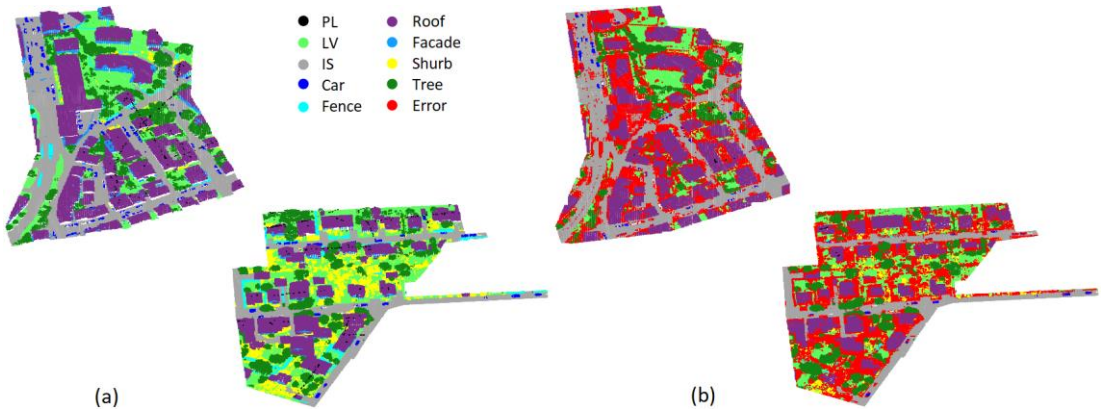

Figure 4. (a) Vaihingen test dataset with ground-truth labels, and (b) semantic segmentation results with error (false negative, red).

Block size (No. of points)	5m×5m (2,048)						10m×10m (2,048)						15m×15m (2,048)						10m×10m (4,096)	
Batch size	24		32		36		24		32		36		24		32		36		32	
Class\Metrics	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
PL	12.4	6.6	4.6	2.4	21.2	11.9	3.0	1.5	15.1	8.2	5.7	2.9	4.4	2.4	0.8	0.4	1.1	0.5	1.6	0.8
LV	70.3	54.3	74.2	58.9	76.7	62.1	75.5	60.7	74.8	59.7	71.9	56.2	73.4	57.9	73.3	57.9	73.2	57.7	73.9	58.6
IS	83.9	72.3	82.9	70.7	86.7	76.5	88.5	79.3	87.5	77.7	87.7	78.1	87.9	78.4	86.7	76.5	85.9	75.3	86.3	75.9
Car	24.8	14.2	**52.8**	35.9	51.1	34.3	38.1	23.5	39.7	24.8	32.5	19.4	30.0	17.7	29.6	17.3	30.3	17.8	39.3	24.6
Fence	16.6	9.1	14.4	7.8	19.9	11.1	11.5	6.1	17.4	9.4	11.1	5.9	12.0	6.4	10.3	5.4	15.8	8.6	23.1	13.1
Roof	69.8	53.6	71.0	55.1	75.0	59.9	73.0	57.4	79.9	66.5	59.6	42.4	65.6	48.8	71.1	55.2	65.3	48.4	72.2	56.5
Facade	15.1	8.2	14.8	8.0	12.4	6.6	14.0	7.5	14.7	7.9	10.4	5.1	7.5	3.9	5.7	2.9	10.8	5.9	13.3	7.1
Shurb	26.5	15.3	29.6	17.3	33.6	20.2	22.5	12.7	24.3	13.8	28.2	16.4	28.3	16.7	28.7	16.7	20.7	11.6	17.6	9.7
Tree	61.3	44.2	63.2	46.2	66.5	49.8	59.2	42.1	61.0	43.9	52.2	35.3	50.9	34.2	53.9	36.9	49.9	33.2	58.9	41.7
mF_1, mIoU	42.3	30.9	45.3	**33.6**	49.2	36.9	43.3	32.3	46.0	34.7	39.9	29.1	40.0	29.6	39.6	**29.9**	39.2	28.8	**42.9**	**32.0**
OA	64.4		**67.7**		70.4		70.0		**72.8**		64.4		66.6		**68.2**		65.2		68.1	

Table 1. PointNet performance metrics for the Vaihingen test dataset (values are in %).

3.1.3 Sensitivity with batch size: For the batch size of 32, OA (and mIoU) for block sizes 5m×5m, 10m×10m and 15m×15m are 67.7% (33.6%), 72.8% (34.7%), and 68.2% (29.9%), respectively. These results show that specific (larger/smaller) block size does not guarantee better results. We also perform the algorithm and classified points with 10m×10m blocks having 4,096 points instead of 2,048 points in each block. Results in the last two columns in Table 1 show that more points (4,096) in a 10m×10m block and batch size of 32 do not produce better results than less sample points (2,048) in the same size block of 10m×10m. Rather, the mF1 (46.0%) and mIoU (34.7%) for 4,096 samples are decreased to 42.9% and 32.01%, respectively.

To achieve the highest OA of 72.8% the algorithm took time of 543s and 24s for training and tests, respectively.

3.2 Experiment 2 (DALES data)

We used another recently introduced ISPRS benchmark ALS dataset for the second experiment (Varney et al., 2020). This LiDAR dataset was collected using a Riegl Q1560 dual channel system with the flying altitude of 1,300 m, over the City of Surrey in British Columbia, Canada. This dataset is named DALES (Dayton Annotated LiDAR Earth Scan). This dataset with labels covers an area of 10km^2 that consists of 40 tiles. Each tile of 500m^2 contains on average 12 million points with a resolution around 50 points/m^2. The mean error for the vertical accuracy is ±8.5 cm. The dataset was denoised by a robust statistical method (Nurunnabi et al., 2015). It consists of a variety of landscape including office, park, high rise buildings, residential buildings, and natural objects. The data points are labelled as ground, vegetation (Veg.), car, truck, power line (PL), fences, poles, buildings and unclassified (uC). The dataset randomly separated into 29/11 tiles of 70/30 % of points for training/test sets. We selected eight tiles of 98,181,415 points having different objects for training set, one tile of 11,345,691 points for validation set, and one tile of 13,669,819 points for the test set (Fig. 5a). We performed PointNet in a similar way using the same hyper-parameters used in Experiment 1. This time, we chose a block size of 10m×10m, and a batch size of 24.

3.2.1 Sensitivity with number of points in a block: To evaluate the effects of sampling a different number of points from a block, we sampled 1024, 2048 and 4096 points for a block of 10m×10m. Table 2 reports that the results for 2,048 (OA=91.1%) and 4,096 (OA=91.5%) points from a block are almost similar, but both are significantly better than the result (OA=87.0%) for the blocks with 1,024 sample points. Results of sampling 4,096 points in a block of 10m×10m are in Fig. 5b that show many points are misclassified as FN (red) for all the classes. The Fig. 5b indicates a roof within a yellow circle is fully misclassified as ground.

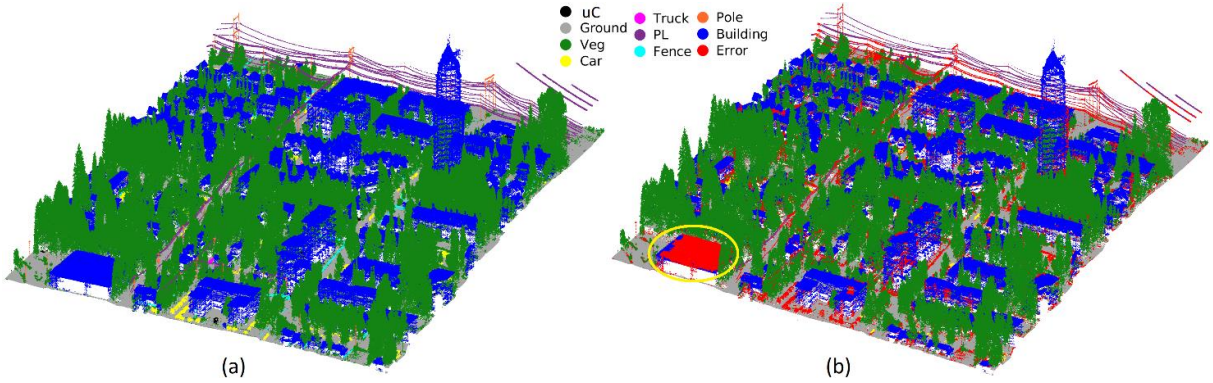

Figure 5. (a) DALES test dataset with ground-truth point labels, and (b) semantic segmentation results for the test dataset, including error (false negative, red).

No. of points in a block	1,024		2,048		4,096		4,096		4,096 (x, y, z)	
Class\Metric	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
uC	0.8	0.4	3.2	1.6	2.5	1.2	-	-	0.7	0.4
Ground	94.6	89.7	95.1	90.7	95.3	91.1	94.6	89.8	**92.9**	**86.8**
Vegetation	83.5	71.6	89.4	80.8	89.5	81.0	89.1	80.4	84.3	72.8
Car	6.3	3.3	30.1	17.7	**46.1**	**30.0**	**29.6**	**17.4**	12.1	6.5
Truck	0.0	0.0	0.0	0.0	0.0	0.0	-	-	0.0	0.0
PL	55.6	38.5	73.1	57.6	69.0	52.7	56.9	39.8	51.1	34.4
Fence	0.0	0.0	9.1	4.8	7.2	3.7	-	-	0.0	0.0
Poles	0.0	0.0	16.9	9.2	19.1	10.6	-	-	1.2	6.1
Building	81.8	69.2	90.6	82.8	91.1	83.7	88.6	79.6	85.6	74.8
mF_1, mIoU	35.8	30.3	45.3	38.4	**46.6**	**39.3**	71.8	61.4	36.4	**30.7**
OA	**87.0**		**91.1**		**91.5**		90.8		**87.5**	

Table 2. PointNet performance metrics for the DALES test dataset (values are in %), '-' indicates no available results for the class, because the class was not considered for the analysis.

Class	Training points	Test points
uC	725,115	26,385
Ground	49,670,953	5,071,173
Veg	30,820,218	5,223,011
Car	784,488	122,904
Truck	95,266	1,883
PL	170,811	70,356
Fence	465,389	46,431
Pole	96,333	10,734
Building	15,352,842	3,096,942
Total	98,181,415	13,669,819

Table 3. Points distribution for the training and test datasets.

3.2.2 Sensitivity with a number of points belong to a class: The point distributions for the training and test sets are given in Table 3. Table 3 shows huge gaps between the number of points for some of the classes, e.g., in the training sets, truck and ground classes consist of 95,266 and 49,670,953 points, respectively. Likewise, the 1st experiment, the results reveal that the accuracy of the point classification is subjugated by the classes having more training points. For example, for sample size 4,096, Table 2 (Columns 6, 7) shows the highest accuracy of F_1=95.3%, IoU=91.1% for the ground class, and missed to classify any point accurately from Trucks. We also computed results without the groups having a smaller number of points: uC, truck, fence and pole. The results are shown in Table 2 (Columns 8, 9), which explore that the values of both F_1 and IoU for all the classes are smaller than earlier when all the classes (points) were considered. For example, after removing the classes (uC, truck, fence and pole) the values of F_1 and IoU for car are 29.6% and 17.4% that were 46.1% and 30.0%, respectively, where all the classes were considered.

3.2.3 Sensitivity with the input vectors: To see the effects of the input vectors, we perform PointNet with 9 input vectors (x, y, z, RN, z_h, scan angle, x_n, y_n, and z_n). Unlike the previous experiment, we did not consider intensity, because it was not available. Moreover, we perform the algorithm only with 3 inputs vectors (x, y, z). Notable changes occur in the results, accuracies drop down significantly for all the classes with OA=87.5%, mF_1=36.4% and mIoU=30.7%. For example, for the ground class, considering only x, y, and z as the input vectors, F_1 and IoU are 92.9% and 86.8%, which are 95.3% and 91.1%, when we consider all 9 input vectors, and with all the classes.

3.3 Experiment 3 (Dudelange data)

As the third experiment, we use the LiDAR survey data of the Luxembourg territory that provided by Administration of Cadastre and Topography (ACT), covering the city of Dudelange. We name it Dudelange data, used in Nurunnabi et al. (2021). These data have open access at https://data.public.lu/en/datasets/lidar-2019-releve-3d-du-territoire-luxembourgeois/. They were collected by using a LiDAR based ALS system. Scanning height for the dataset was 1100 m and side overlap was minimum 60%. They have an average resolution of around 15 points/m^2 with a horizontal and vertical precision of ±3 cm and ±6 cm, respectively. The data are designed into 500m×500m tiles, the tiles are grouped

together up to nine tiles covering a 1500m×1500m area. Each tile contains on average around 5-7 million points.

We selected four tiles of 500m×500m in urban areas consisting of 24,481,206 points. Two tiles are selected randomly and coupled for the training dataset, and the validation and test datasets are selected from the other two tiles. The training, validation and test sets consist of 12,866,206, 5,731,462 and 5,883,538 points, respectively. The Dudelange data are available with labels: ground, low vegetation, medium vegetation, high vegetation, buildings, water, bridges, high voltage power lines and unclassified (uC) points. To get more accurately labelled data, we relabelled the datasets into four groups: ground, vegetation, buildings and unclassified (all the other different objects) again by using the Trimble Business Center (TBC) and LAStools software. We perform PointNet as the previous experiments using the same hyper-parameters. We chose a block size of 10m×10m, and a batch size of 32 in this experiment.

3.3.1 Sensitivity with number of points in a block: We sampled 1024, 2,048 and 4,096 points for a block similar to the 2nd experiment. 1st part (Columns 4 to 9) of the Table 4 shows that unlike the result of 2nd experiment, with a block size of 10m×10m the results for 1,024 (OA=95.9%), 2,048 (OA=95.5%) and 4,096 (OA=95.8%) points from a block are almost similar. Results of sampling 1,024 points in a block are in Fig. 7a that show FN (misclassified) points for all the classes as red. To see the results in more detail, in Fig. 7b, we plot a specific part in the white rectangle in Fig. 7a, where a high-rise building and some roofs are highlighted, Fig. 7b shows that many facade and roof points are misclassified as non-building points (yellow rectangles and cyan ellipses). It shows that the false negative points are mostly from the building-facades.

3.3.2 Sensitivity with input vectors: In this experiment, we performed PointNet based on two sets of 9 input vectors: A = {x, y, z, I, RN, z_h, x_n, y_n, z_n} and B = {x, y, z, R, G, B, x_n, y_n, z_n}. We see remarkable gap between the results from the two sets (A and B) of input vectors. When 1,024 points are sampled from each block of size 10m×10m, PointNet achieves 95.9%, 84.6% and 76.8% of OA, mF_1, and mIoU, respectively for the input set A. Whereas, for the input set B, when we replace R, G, B as the alternative of I, RN and z_h, results of OA, F_1, and IoU are equal to 89.1%, 62.8% and 54.4%, respectively. Note that, Qi et al. (2017a) used the set B as the input vectors in their experiments for semantic segmentation.

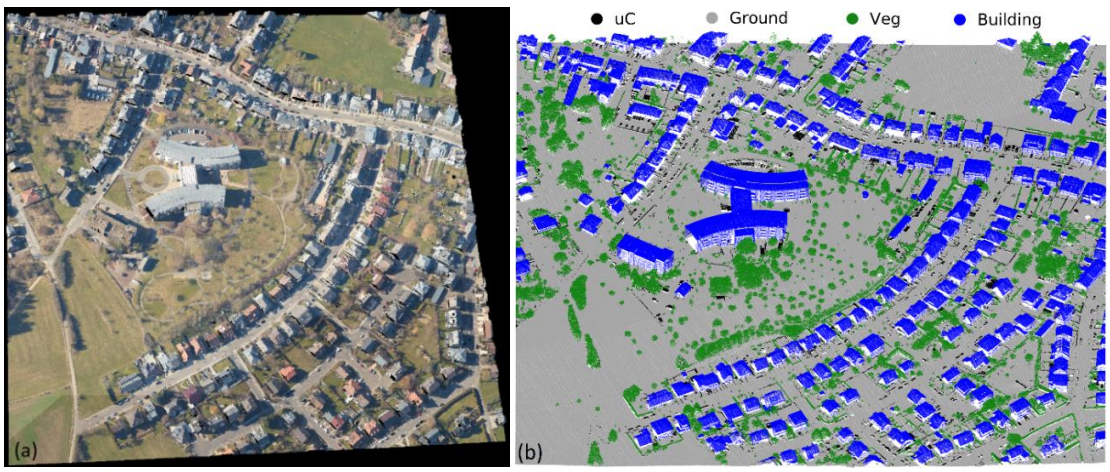

Figure 6. (a) Dudelange test dataset in an urban area contains large-medium-small trees, large-small commercial and residential buildings, car, etc., (b) test dataset with ground-truth point labels.

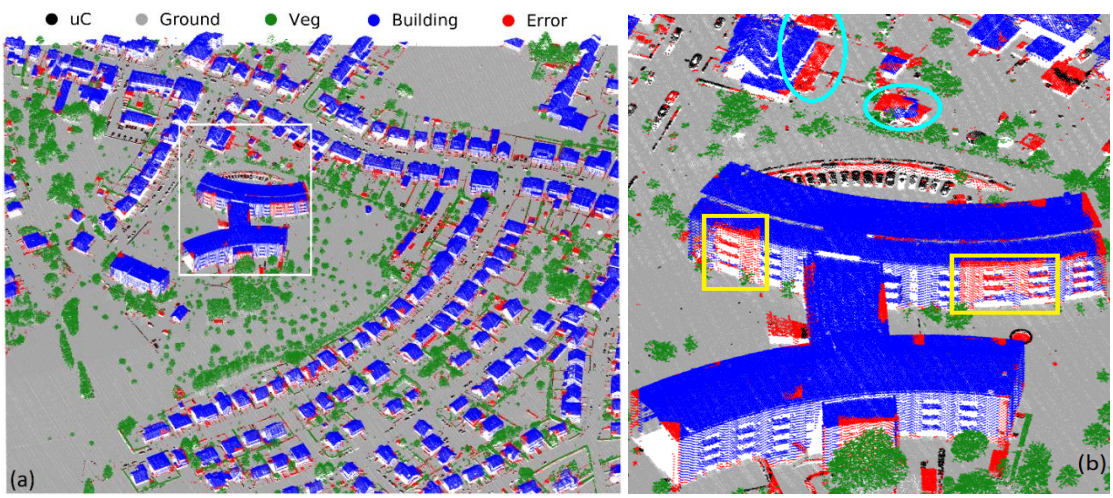

Figure 7. (a) Semantic segmentation results with errors (false negative, red) for the Dudelange test dataset, and (b) selected area in white rectangle in Fig. 7(a) to magnify the detail, many facade points in the yellow rectangles and roof points in the cyan ellipses are falsely identified (FN, red) as non-building points.

Input vectors			A= {x, y, z, I, RN, z_h, x_n, y_n, z_n}						B= {x, y, z, R, G, B, x_n, y_n, z_n}					
Training points count	Test points count	No. of points in a block	1,024		2,048		4,096		1,024		2,048		4,096	
		Class\Metric	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
709,901	192,214	uC	54.6	37.6	42.9	27.3	51.5	34.7	2.3	1.2	1.7	0.9	3.0	1.5
10,765,583	4,050,146	Ground	98.9	97.7	98.6	97.2	98.8	97.6	94.7	90.0	87.9	77.9	94.3	89.2
4,077,626	721,862	Vegetation	90.4	82.5	89.6	81.1	89.9	81.6	69.8	53.6	70.4	54.3	74.7	59.6
3,044,558	919,316	Building	94.4	89.4	93.9	88.6	94.1	88.8	84.4	73.0	63.2	46.2	84.9	73.7
18,597,668	5,883,538	mF_1, mIoU	**84.6**	**76.8**	81.3	73.6	**83.6**	**75.7**	**62.8**	**54.4**	55.8	44.8	64.2	56.0
		OA	**95.9**		**95.5**		**95.8**		**89.1**		78.4		89.2	

Table 4. PointNet performance for the Dudelange test dataset (values are in %).

4. GENERAL DISCUSSION AND CONCLUSIONS

This paper investigated PointNet, the first developed end-to-end DL algorithm to directly processes raw point clouds for large-scale outdoor environment. Experiments show that although the architecture is computationally efficient and achieved notable success in indoor point clouds, but when it is for large-scale outdoor point clouds it was not up to the desired level. From the results of three ALS datasets, it is revealed that PointNet is vulnerable to (i) point density, (ii) block size, (iii) number of points within a block, (iv) batch size, and (v) input point vectors. This algorithm performs better for the data with sufficient density, e. g., for 10m×10m block with 2,048 points; OA (91.1%) for the DALES data with density of 50 points/m^2 were significantly larger than the OA (72.8%) for the Vaihingen data with density of 4 points/m^2. It performs well for semantic segmentation for the classes with a sufficient number of points, but the results are very poor for the classes with a small number of points, even sometimes unable to label the points in a class of small number of points. Although, Dudelange data have lesser point density (15 points/m^2) than the DALES data (50 points/m^2), for the input vector set A and 4,096 points in a block the values of the performance metrics for the Dudelange data (OA=95.8%, mF_1=83.6% and mIoU=75.7%) are significantly higher than for the DALES data (OA=91.5%, mF_1=46.6% and mIoU=39.3%). The results explored the cause is that point distributions for different classes are almost homogenous for the Dudelange data. These results indicate that PointNet provides a promising opportunity for semantic segmentation without any input data transformation, if enough training data are available having a sufficient point density and enough points in a class. Users should be careful about fixing hyper-parameters to get the best results. Hyper-parameters behave different on diverse datasets based on their underlying pattern. Moreover, input vectors have significant impact on the results for point labelling. We see for the Dudelange data of sample size 1,024 for a 10m×10m block, OA was 6.8% (95.9%-89.1%) higher when input vectors I, RN and z_h were used in places of R, G and B.

However, the algorithm needs improvement with a deeper knowledge about the underlying characteristics of ALS data to get the required level of accuracy. Since, there are big gaps between the spatial extent of different sizes of objects in large-scale ALS point clouds, inclusion of adaptive and/or multi-scale neighborhood information into the network as input vectors is expected to be beneficial. Additionally, objects geometry should be considered for the better results. Further research is envisaged to improve the network to make the method more robust to the standard hyper-parameters used in PointNet. Another issue will be considered to improve the classification accuracy for the classes (e.g., car, poles, and building facades) usually that have a smaller number of points, and/or consist of vertical surfaces.

ACKNOWLEDGEMENTS

This study is with the Project 2019-05-030-24, SOLSTICE - Programme Fonds Européen de Developpment Régional (FEDER)/Ministère de l'Economie of the G. D. of Luxembourg.

REFERENCES

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2018. Learning representations and generative models for 3D point clouds. Intl. Conf. Machine Learning, PMLR, 40 –49.

Agoub, A., Schmidt, V., Kada, M., 2019. Generating 3D city models based on the semantic segmentation of lidar data using convolutional neural networks. ISPRS Ann. of Photogramm. Remote Sens. Spat. Info. Sci., 4.

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. Computers & Graphics, 88: 24–34.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests., ISPRS Archive, XXXVIII-3(W8).

Chen, L., Wang, Q., Lu, X., Cao, D., Wang, F-Y., 2019. Learning driving models from parallel end-to-end driving data set. Proc. of the IEEE, 2019, 108(2): 262–273.

Gong, Z., et al., 2020. A frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data. ISPRS J. Photogramm. Remote Sens., 159: 90–100.

Guo, Z., et al., 2018. Semantic segmentation for urban planning maps based on U-Net. IGARSS, 6187–6190.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, 2020. Deep learning for 3D point clouds A survey.IEEE TPAMI,1–27.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., Pollefeys, M., 2017. Semantic3D.Net: A new large-scale point cloud classification benchmark. arXiv:1704.03847.

Han, X., Dong, Z., Yang, B., 2021. A point-based deep learning network for semantic segmentation of MLS point clouds. ISPRS J. Photogramm. Remote Sens., 175:199 –214.

Hu, Q., et al., 2020. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. CVPR, 11108–11117.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML, 448–456.

Isenburg, M., 2014. LAStools-efficient LiDAR processing software, http://rapidlasso.com/LAStools

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial transformer networks. ArXiv:1506.02025.

Jing, Z,, Guan, H., Zhao, P., Li, D., Yu, Y., Zang, Y., Wang, H., Li, J., 2021. Multispectral LiDAR point cloud classification using SE-PointNet++, Remote Sensing, 13, 2516.

Kang, Z., Li, N., 2019. PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. Australian Journal of Intelligent Information Processing Systems, 16 (2): 35–43.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ko, Y., Lee, S-H., 2020. Novel method of semantic segmentation applicable to augmented reality. Sensors 20(6): 1737.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process Syst, 25: 1097–1105.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. IEEE CVPR,4558–4567.

LeCun, Y., et al., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4): 541–551.

Li, G., Yang, Y., Qu, X., 2019. Deep learning approaches on pedestrian detection in hazy weather. IEEE Trans. Ind. Electron., 67(10): 8889–8899.

Li, W., Luo, Z., Xiao, Z., Chen, Y., Wang, C., Li, J., 2021. A GCN-based method for extracting power lines and pylons from airborne LiDAR data, IEEE Trans. Geosci. Remote Sens.

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M., 2020. Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale urban environments, IEEE Trans. Intell. Transp. Syst.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. IEEE TPAMI, doi: 10.1109/TPAMI.2021.3059968.

Nair, V., Hinton, G., 2010. Rectified linear units improve restricted Boltzmann machines. ICML.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogramm. Remote Sens., 87:152–165.

Nurunnabi, A., Belton, D., West, G., 2014. Robust statistical approaches for local planar surface fitting in 3D laser scanning data. ISPRS J. Photogramm. Remote Sens., 96: 106 –122.

Nurunnabi, A., West, G., Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. Pattern Recognit., 48(4):1404–1419.

Nurunnabi, A., Belton, D.,West, G., 2016. Robust segmentation for large volumes of laser scanning three-dimensional point cloud data.IEEE Trans. Geosci. Remote Sens.,54(8):4790-4805.

Nurunnabi, A.,Teferle, F. N., Li, J., Lindenbergh, R., Hunegnaw, A., 2021. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci., 24:1–8, XXIV-ISPRS Congress, 5-9 July.

Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L., 2016. Volumetric and multi-view CNNs for object classification on 3d data. IEEE CVPR.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep learning on point sets for 3d classification and segmentation. IEEE CVPR, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413.

Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F., 2018. Fully-convolutional point networks for large-scale point clouds. ECCV, 596 –611.

Romero-Jarén, R., Arranz, J. J.,2021. Automatic segmentation and classification of BIM elements from point clouds. Autom. Constr., 124:103576, 1–17.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E. G., 2015. Multi-view convolutional neural networks for 3d shape recognition. ICCV.

Su, H., et al., 2018. SPLATNet: Sparse lattice networks for point cloud processing. IEEE CVPR, 2530–2539.

Sun, X., Lian, Z., Xiao, J., 2019. SRINet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. ACM Int. Conf. Multimedia, 980–988.

Thomas, H., Qi, C. R., Deschaud, J-E., Marcotegui, B., Goulette, F., Guibas. L. J., 2019. KPConv: Flexible and deformable convolution for point clouds. IEEE ICCV, 6411–6420.

Varney, N., Asari, V. K., Graehling, Q., 2020. DALES: a large-scale aerial LiDAR data set for semantic segmentation. IEEE CVPR Workshops, 186 –187.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M., 2019. Dynamic graph CNN for learning on point clouds. ACM Transactions On Graphics, 38(5):1–12.

Yang, B., Wang, S., Markham, A., Trigoni, N., 2020. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction. IJCV, 128 (1): 53–73.

Yu, D., Ji, S., Liu, Wei, S., 2021. Automatic 3D building reconstruction from multi-view aerial images with deep learning. ISPRS J. Photogramm. Remote Sens., 171: 155 –170.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., Smola, A., 2017. Deep sets, arXiv preprint arXiv:1703.06114.

Zhang, J., Lin, X., Ning, X., 2013. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. Remote Sensing, 5(8): 3749–3775.

Zhang, L., Li, Z., Li, A., Liu, F., 2018. Large-scale urban point cloud labeling and reconstruction. ISPRS J. Photogramm. Remote Sens., 138: 86 –100.

Zhao, H., Jiang, L., Fu, C-W., Jia, J., 2019. PointWeb: Enhancing local neighborhood features for point cloud processing. IEEE CVPR, 5565 – 5573.

Zhang,W., Xiao, C., 2019. PCAN: 3D attention map learning using contextual information for point cloud-based retrieval. IEEE CVPR, 12436–12445.