International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

A PointNet Application for Semantic Classification of Ramps in Search and Rescue Arenas

Kaya TURGUT*[1], Burak KALECİ[2]

Submitted: 10/05/2019 Accepted : 26/09/2019

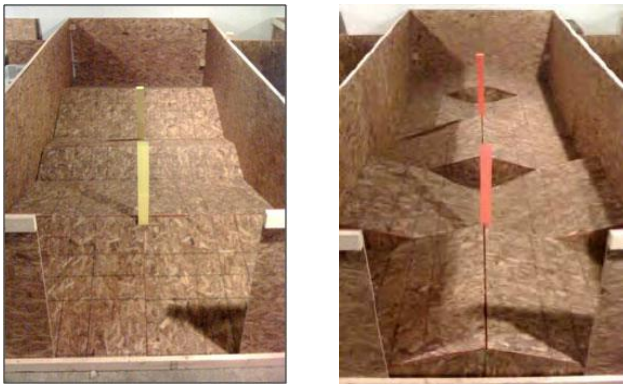
Abstract: Search and rescue environments could be dangerous for humans due to risks for potential structural breaking down and leakage of hazardous materials. Using robots in these environments would be an appropriate solution to reduce these risks. The National Institute of Standards and Technology (NIST) proposed reference test areas for measuring autonomous navigation capabilities of mobile robots. In this paper, we present a PointNet application for semantic classification of ramps through point cloud data of reference test arenas. Since the walls and terrain carry important semantic information for robot navigation, they are also considered. The previous studies that address the semantic classification problem mostly used image and/or 2D laser range data. However, the image data may not be suitable for dusty and poorly lightened search and rescue environments and 2D laser range data may not represent 3D geometry of the objects. Since point cloud data have the ability to describe 3D geometry and it is not affected by the negative aspects of these environments, it could be appropriate to classify ramps, walls, and terrain. Eskisehir Osmangazi University (ESOGU) laboratory building is modelled in GAZEBO simulation environment. Then, the ESOGU RAMPS dataset is generated through navigating a Pioneer P3-AT mobile robot with Asus Xtion Pro sensor in this environment. The robot is controlled via Robot Operation System (ROS). The dataset contains two types of ramp (inclined and flat), terrain and wall classes. The PointNet is applied to train and test the dataset. The metric and visual results are presented to analyze the classification performance of the PointNet.

Keywords: *PointNet, deep learning, NIST ramps, mobile robot, 3D Point Cloud*

1. Introduction

As robotic and computer vision improves, robots have been used in many application areas. One of these application areas is search and rescue missions. These missions could be challenging because robots must operate in dirty, dull, and dangerous environments. In addition, robots must have adequate abilities such as situation awareness and autonomous navigation for performing tasks. It is necessary to constantly measure the efficiency of the developed algorithms for improving these capabilities of robots. On the other hand, search and rescue environments are not frequently encountered and construction of these environments in the laboratory is both costly and difficult. In order to bring a solution to this situation, organizations such as RoboCup and DARPA have organized competitions related to search and rescue tasks. In these competitions, robots have been evaluated with certain standards [1]. The National Institute of Standards and Technology (NIST) proposed reference test areas for measuring autonomous navigation capabilities of mobile robots [2]. Figure 1 depicts examples of NIST reference test areas.

Over the years, robots have increased autonomous navigation capabilities thanks to competitions. These improvements have encouraged the researchers for using robots in more difficult arenas. In recent years, the competition arenas have involved more crossing and continuous ramps instead of flat terrain. Additionally, 3D terrain classification task, which requires advanced sensing and reasoning skills, has been introduced [3]. In order to cope with these difficulties while navigating autonomously, robots must have adequate information about ramps, walls, and terrain. This information can improve robot navigation in various ways: 1) Robots can adjust their velocity by using ramp slope to navigate more reliably. 2) Robots can choose appropriate waypoints to avoid losing their balances while passing through the ramps. 3) Robots can improve the maps by including ramps, walls, and terrain. Then, these maps may be utilized to generate path plans.



(a) Continuous Ramp Example (b) Crossing Ramp Example

Fig. 1. Examples of NIST reference test areas [4]

The main motivation of this study is to classify ramps in environments like reference test areas via a deep learning technique. Since the walls and terrain carry important semantic information for robot navigation, they are also considered. To the best knowledge of the authors, there is no dataset for NIST standard ramps. In this study, we introduce such a dataset, namely ESOGU RAMPS. Firstly, ESOGU laboratory building that includes continuous, crossing, and flat ramps, walls, and terrain is

[1] *Electrical and Electronics Eng., Eskişehir Osmangazi University, Eskişehir – 26480, TURKEY, ORCID ID : 0000-0003-3345-9339*
[2] *Electrical and Electronics Eng., Eskişehir Osmangazi University, Eskişehir – 26480, TURKEY, ORCID ID : 0000-0002-2001-3381*
** Corresponding Author Email: kayaturgut@hotmail.com*

modelled in GAZEBO [5] simulation environment. A Pioneer P3-AT mobile robot with Asus Xtion Pro sensor is used in collecting 3D point cloud data. The robot is controlled via ROS [6]. Then, the dataset was labelled. The dataset was trained and tested using PointNet [7]. The metric and visual results demonstrate that the PointNet classified ramps, walls, and terrain with an overall 98% classification rate.

The rest of the paper is organized as follows: In Section 2, we summarize previous works about 3D point cloud segmentation and deep learning techniques to classify 3D objects. In Section 3, PointNet architecture is explained. The ESOGU RAMPS dataset is introduced in Section 4. We present experimental results in Section 5 and conclude with Section 6.

2. Related Work

Recently, fast and easy 3D model construction of indoor and outdoor environments becomes possible because of sensors such as Kinect and LIDAR. Thus, extracting meaningful information from the 3D models has gained importance in the computer vision and robotics communities. To achieve this, segmentation and classification problems have been studied as an active subject. Segmentation is defined as grouping points according to their characteristics, whereas the classification is aiming to assign the points to the classes. Many different methods have been proposed for these problems and popular algorithms are reviewed by Grilli et al. [8]. This study divides them into 5 different categories as edge-based [9], region growing [10], model fitting [11, 12], hybrid method [13], and machine learning. These methods, except machine learning, do not require a training phase. In addition, they can easily be implemented via the open source point cloud libraries such as PCL [14]. For these reasons, these methods are widely used in robotic applications. Previous attempts on 3D point cloud segmentation have been quite successful only under certain constraints. For example, region-based methods such as region growing [10] suffer from the time complexity and they are sensitive to points that are selected to start the growing process. RANSAC [11] is fast, accurate, and robust against noise, but the closeness of the points in the whole of the scene should be almost the same. Since RANSAC provides easy implementation, it is preferred in robotic applications. In our previous work [15], RANSAC was applied to point cloud data to segment ramps, walls, and terrain. Then, the segments were classified semantically according to the plane equation. In classical machine learning, firstly, descriptors that are appropriate to the characteristics of the 3D model are determined and then the attributes are obtained. According to these attributes, the point cloud is segmented into meaningful parts. The limitation of this approach is largely dependent on the descriptors and not suitable for complex data [16]. In addition, it tends to over-fitting because 3D descriptors are very high dimensional [17].

With deep learning being popular, it is possible to obtain task-specific attributes from the models. Inspired by the effective results of CNN architectures in 2D, deep learning techniques are adapted for 3D data. When the recent studies are examined, CNN architectures that are applied to 3D models are separated into three different structures: 1) Volumetric CNN (3D CNN), 2) Multi-view CNN (MVCNN) and 3) Geometric (Spectral) CNN. ShapeNet [18] was the first 3D CNN implementation to apply deep learning on 3D models and demonstrated that the attributes learned by deep learning are more effective than hand-crafted attributes. This work represents 3D geometric shapes as a probabilistic distribution using a convolution deep-belief network over the voxel grid. VoxNet [19] is a simple 3D CNN structure with fewer parameters and it can classify point clouds faster and more effectively. 3D CNN should deal with the trade-off between spatial resolution and computational cost. Since the convolution of 3D voxels increases in cubic proportions with respect to spatial resolution, the computational cost is greatly increased. Therefore, the resolution has to be kept in small proportions. Moreover, as the resolution increases, the increased sparsity of the grid structure prevents effective learning of filters [20]. The MVCNN architectures employed the standard CNN and their input is 2D images of different views obtained from 3D models. The features obtained from each image are combined for better recognition rate [17]. MVCNN provides greater accuracy rates than 3D CNN models. However, MVCNN architectures lead to the loss of important structural information in the 3D structure [21]. Geometric (Spectral) CNN architectures generalize the CNN architecture to non-Euclidean areas such as manifold or graph [22]. In [23], Bruma transformed the convolution concept to the spectral field. In this architecture, a spectral convolution layer similar to the classical Euclidean convolution layer was introduced. In this way, the grid was replaced by weighted graphs while CNN was generalized for the graphs. Since spectral CNN structures are Fourier-based, they are domain-dependent. Therefore, a model learned with this architecture cannot be easily transferred to another [24]. In addition to the implementation of the deep learning architecture to the graphs, it is generalized to the 3D models represented by the manifold such as meshes. Masci [25] developed the first CNN on meshed structures. The mixed model MoNet [22], which can be applied for graphs or manifold structures, uses the parametrical structure rather than manual weighted functions.

In contrast to the aforementioned studies, there are multi-layer perceptron (MLP) architectures that accept the raw point cloud [7, 26, 27]. PointNet [7], an effective and simple architecture for unstructured point cloud, was the first study in this field and pioneered for other studies. The architecture yields successful outcomes for object classification, object segmentation, and semantic segmentation problems. PointNet does not take into account the neighborhood of points for the local characteristics. Based on this, the PointNet++ (adaptive PointNet) [26] architecture that uses the PointNet model has been proposed. In this architecture, local features that capture geometric structures from small neighborhoods are extracted. However, determining this neighborhood parameter is not a trivial problem. Recently, studies have been conducted to examine different neighborhood relations based on PointNet architecture. Instead of taking a single neighborhood to increase the semantic content of the model, mechanisms have been developed to increase the content, taking into account multiple neighborhoods of the same center [27].

In this study, we mainly focus on the classification of simple structures such as wall, terrain, inclined ramp, and flat ramp. Since many 3D sensors give a raw point cloud for the model of the scene, we aim to use raw data. In this way, we could avoid performing preprocessing steps such as conversion to graph or manifold and voxelization. When the previous studies are analyzed, 3D CNN architectures are only used for object classification problems and they require voxelization pre- processing.

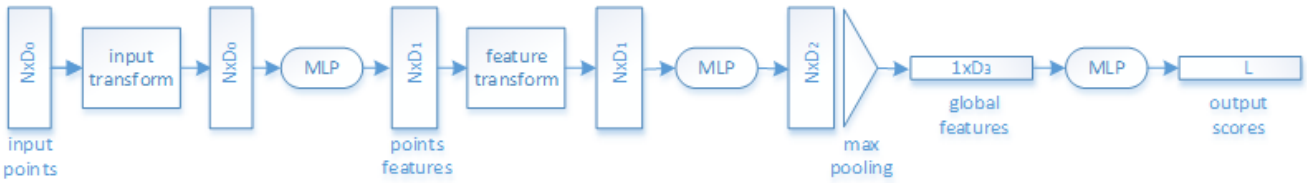


Fig. 2. The simplified model of the classification problem

Moreover, the voxel resolution is an important parameter because there is a trade-off between computational cost and model sensitivity. In MVCNN architectures, the surface model must represent point cloud. The features are extracted through images taken from the surface model and thus it does not fully represent the 3D geometry. Geometric CNN architectures that use graphs or manifolds also require pre-process, such as neighborhood determination or surface modeling. It is also a challenging problem to generalize since it is eigen-based. On the other hand, the PointNet architecture is applied directly to the raw point cloud without requiring any pre-processing. In addition, it does not require the determination of difficult parameters such as the radius and the number of neighborhoods, unlike PointNet++. For these reasons, PointNet is simple and easy to use end-to-end deep learning architecture and it is appropriate to classify simple structures such as wall, terrain, inclined ramp, and flat ramp.

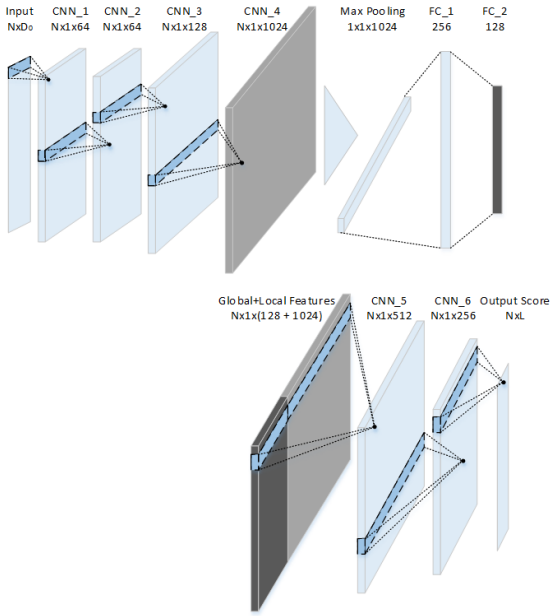
3. The PointNet Architecture

Point cloud data could be problematic due to its unstructured format to feed deep learning architectures. In order to overcome this problem, some researchers transform the data into regular formats such as voxels or images before applying to the architectures. The PointNet is the first simple end-to-end deep learning architecture, which accepts point cloud without the necessity of preprocessing. Besides the PointNet is simple and fast, it produces better or closer result compared to CNN architectures.

Point cloud is composed of n points $\{p_i | i = 1, \dots, n\}$ in which each point lies in the 3D Euclidean space corresponds to a vector consisting of xyz coordinates. These vectors can be extended by adding global or local features such as color, normal, and curvature. Although it has a simple data structure, some difficulties arise when attempting to process: 1) The points are in an unstructured form. Therefore, the architecture should be invariant from all possible permutations for point cloud set. 2) Since the neighborhood points can be related to each other; the relationship between the neighborhood points must be considered. 3) Transformations could be applied to points, thus the architecture should be robust against these transformations. PointNet architecture designed considering these challenges provides capabilities such as independence from permutations, invariance under transformation, and capturing local features.

The simplified representation of the PointNet model for the classification problem is given in Figure 2. It accepts NxD_0 $\{N$: number of points, D: feature dimension$\}$ point cloud set as input. The input transformation has a mini PointNet network (T-net) that standardizes the point cloud according to rotation before the feature extraction process. Then, multi-layer perceptron (MLP) is applied to extract the features. Similar to the input transformation in spatial space, the feature transformation architecture ensures that the features learned from the point cloud are independent from transformation. The standardized features are mapped to a higher dimension through the MLP weight matrices shared in each consecutive layer. The transition from point-based features to model-based features is achieved thanks to the maximum pooling method on the local features extracted for each point in the point cloud model. In addition, maximum pooling method allows the architecture to be independent of the point order. Summarizing process is executed to acquire global features for the classification problem.

Fig. 3. The PointNet architecture of the semantic segmentation problem



The PointNet architecture can be used for object classification, part segmentation, and semantic segmentation problems. In this study, we focused on the architecture of the semantic segmentation problem. Unlike the other problems, the input of the architecture for semantic segmentation consists of larger scenes. Therefore, instead of using the whole scene it divides the scene into certain blocks and feeds the architecture with the points inside the blocks. This process prevents the loss of data substantially because the architecture requires a certain number of points. Since the scene is divided into blocks, input and feature transformations are not required. Figure 3 shows a detailed representation of the model for semantic segmentation problem of PointNet architecture. It accepts NxD_0 $\{N$: number of points, D_0: input feature dimension$\}$ point cloud block set as input. In the first layer, $1x\ D_0$ stride and $1x\ D_0x1x64$ weight matrices are used to merge input features. Consecutive CNN (MLP with shared weight) layers include 1x1 stride and $1x1xD_{n-1}xD_n$ weight matrices to learn D_n dimensional feature instance, in Figure 3, the third CNN layer's weight matrix is $1x1x64x128$ $(1x1xD_2xD_3)$. Learned local features for each point are summarized with the maximum pooling method and a single feature is obtained for the point cloud. Global features of the point cloud are acquired after fully connected layers. While global features are appropriate for the classification problem, local characteristics are also needed to summarize the relationships between points for segmentation. After point-wise local features and global features are concatenated for each

point, the feature extraction step is applied again with CNN layers. Thus, the score for L class (L: class number) of N points is obtained. PointNet uses rectified linear unit (ReLU) as activation function, batch normalization, drop out before the last layer and cross-entropy loss function.from D_{n-1} dimension where n indicates the layer number. For

4. The ESOGU RAMPS Dataset

To construct the ESOGU RAMPS dataset, Gazebo simulation environment and ROS robot interface are used. In the first stage, we modelled ESOGU Electrical & Electronics Engineering Laboratory Building in Gazebo simulation environment. Then, we exploited hector_nist_arenas_gazebo ROS package [28] to include ramps into the simulation environment (Figure 4). A Pioneer P3-AT mobile robot with Asus Xtion Pro RGB-D sensor is used to create the dataset. We employed Robot Operating System (ROS) to control the robot [6].

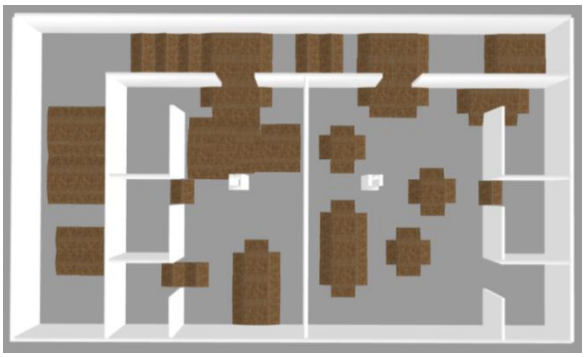


Fig. 4. ESOGU Electrical & Electronics Engineering Laboratory Building Gazebo Model

In the second stage, we located the robot in different positions and orientations and captured scenes via PCL [14]. In these scenes, points must belong to one of the four classes: wall, terrain, inclined ramp, and flat ramp. Figure 5 shows examples for wall, terrain, inclined ramp, and flat ramp classes. In ESOGU RAMPS dataset, there are 681 scenes. The dataset is partitioned into two parts for training and test. The training set contains 581 scenes. In the test set, there are 100 scenes. In both sets, the percentage of points belonging to the wall class, terrain class, inclined ramp class, and flat ramp is 50%, 18%, 24%, and 8%, respectively.

In the last step, we labelled the dataset. Figure 6 shows examples from the labelled dataset. The right column of the figure depicts examples of ground truth of the scenes. In these figures, blue, red, magenta, and yellow represent inclined ramps, walls, flat ramps, and terrain classes, respectively. In the left column, the corresponding RGB images are indicated.

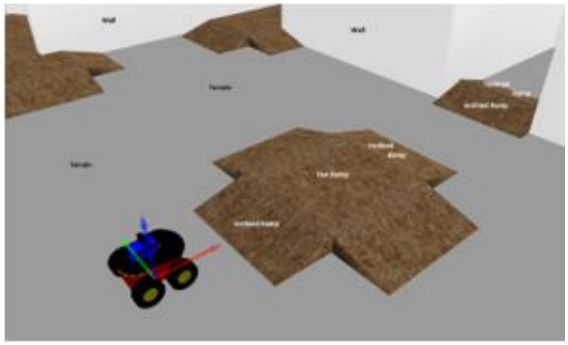


Fig. 5. Examples for wall plane, terrain plane, inclined plane, and straight plane classes

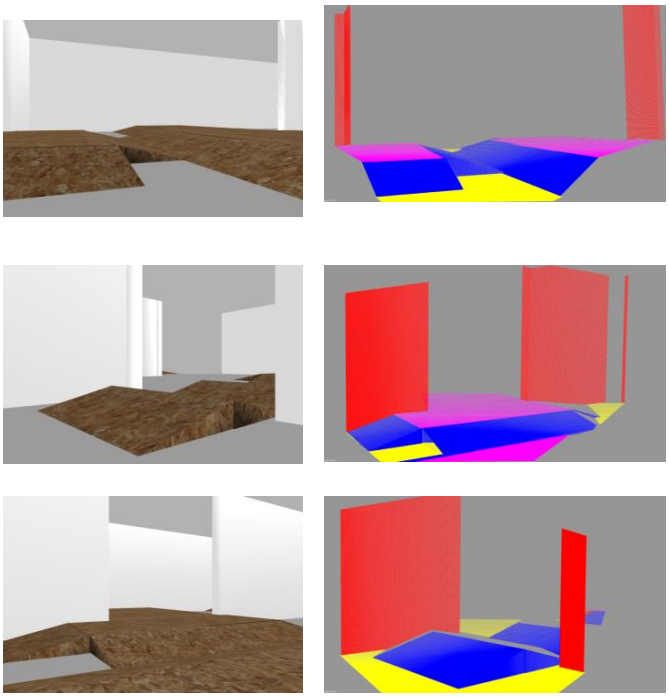


Fig. 6. Example scenes from ESOGU RAMPS dataset

5. Experimental Work

We used PointNet architecture to classify ramps (inclined and flat), terrain and wall classes in the ESOGU RAMPS dataset. To prepare the dataset for the PointNet architecture, each point cloud is divided into $1m^2$ blocks in the xy plane and independent of z dimension. In a point cloud, each point represented with a 6-dimensional (x, y, z, x', y', and z') vector. In the vector, x, y, and z indicate point coordinates while $x', y',$ and z' depict their normalized coordinates respect to maximum point. If the number of points in these blocks is less than 500, the set is discarded. In addition, upsampling or downsampling is applied by using random selection because PointNet accepts a fixed number of points. Thus, the number of points in each block is fixed at 4096. In the training stage, the following parameters are used: the number of points is 4096, batch number is 12, epoch is 50, the learning rate is 0.001, momentum is 0.9, optimizer type is Adam, decay step is 300000, and decay rate is 0.5.

We evaluate the experimental results using the following measures: 1) Intersection over Union (IoU) refers to the ratio of correctly classified samples of a class to summation of the total number of samples and incorrectly classified samples of that class. 2) The recall values of inclined ramp, wall, flat ramp, and terrain classes. Recall refers to the ratio of correctly classified samples of a class to the total number of classified samples for that class. 3) Precision indicates the ratio of correctly classified samples of a class to the total number of samples of that class. 4) Mean Intersection of Union (MIoU) defines the mean of class-based Intersection over Union (IoU). 5) The overall classification accuracy; i.e. the ratio of the number of correctly classified samples to the total number of samples (ACC). The metric and visual results are given in Table 1 and Figure 7, respectively. In the figure, the left column shows ground truth and the right column depicts the corresponding classification result.

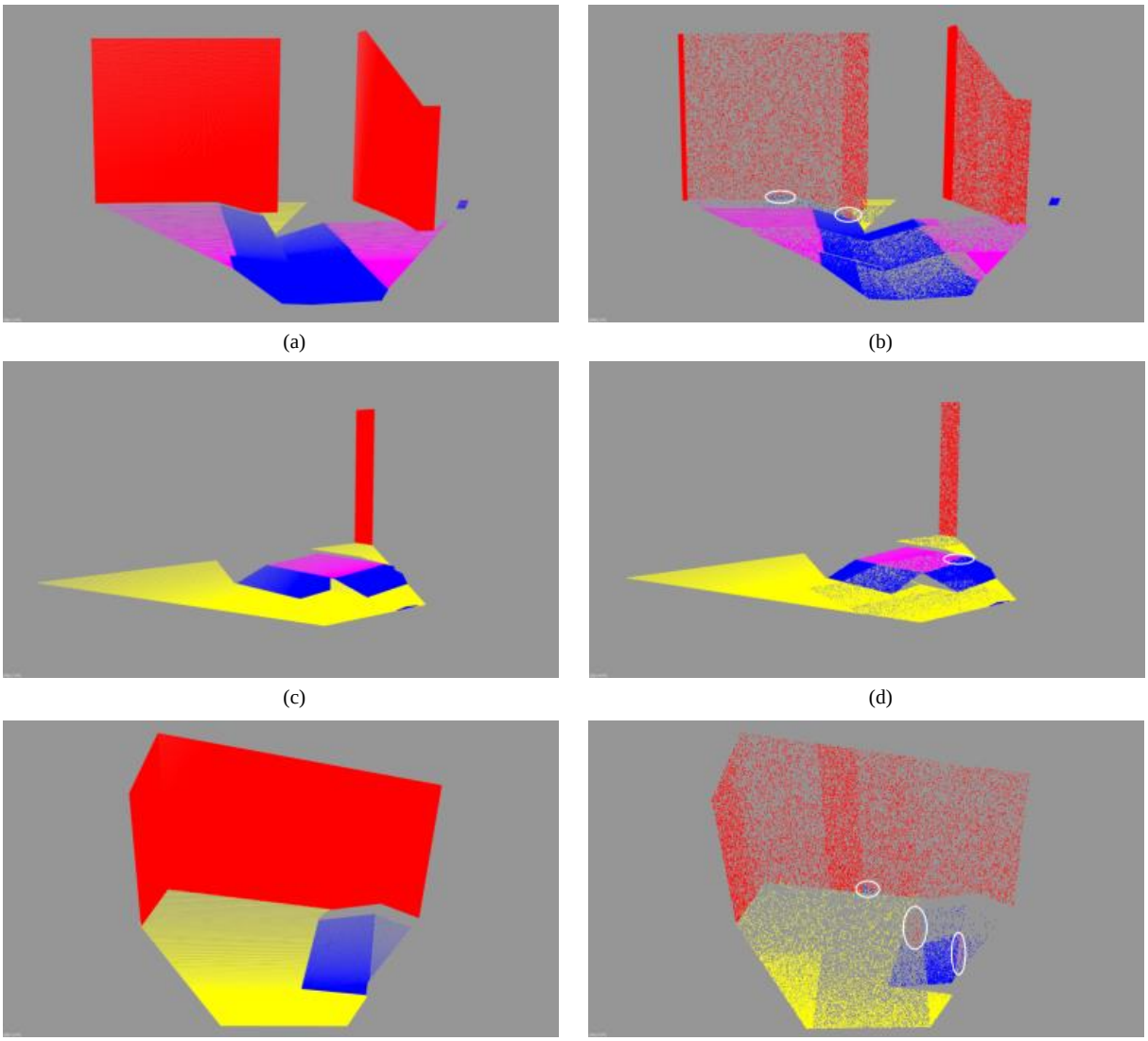


Fig. 7. Visual results of PointNet semantic classification. The left column shows ground truth and the right column depicts the corresponding classification result.

Table. 1. The metric results of the PointNet

	IoU	Recall	Precision	MIoU	ACC
Inclined Ramp	0.946	0.983	0.961	0.971	0.989
Wall	0.990	0.991	0.999		
Flat Ramp	0.950	0.968	0.981		
Terrain	0.998	0.999	0.998		

The IoU results indicate the ratio of overlap between ground truth and classified data. When the IoU values in the table are examined, it is observed that the highest value belongs to terrain class. The IoU value for the terrain is supported by the recall and precision values of this class. The recall value of the terrain class depicts that the PointNet is able to classify this class with high accuracy. Besides, it is clear that the PointNet does not confuse the terrain class with other classes when we interpret the precision value of this class. The wall class has the second-highest IoU value in the table. When we analyze the precision value of the wall class, PointNet produces almost the same result with terrain. The recall value of the wall class is a little lower than the terrain class. The reason for this is shown in Figure 7 (b). In some cases, which are emphasized with white circle in the figures, the wall class points are incorrectly classified as inclined ramp class. A similar result to the wall class also comes out for the flat ramp class. However, the recall and precision values for flat ramp class are lower when it is compared to the wall class. Figure 7 (d) illustrates an example to clarify the results. The flat ramp class and the inclined ramp class are usually placed consecutively and in some cases, as shown in the figure, the PointNet does not accurately classify flat ramp class. The low precision value of the inclined ramp class indicates that this class tends to confuse with other classes (Figure 7 (f)). As a result, some incorrectly classified points were encountered between the inclined ramp, flat ramp and, wall classes. However, as seen from the numerical and the visual results, the number of these incorrectly classified points is very low compared to the total number of points.

In training and test processes, the PointNet divides the data to the blocks and each block must have exactly 4096 points. When a block includes more than 4096 points, the PointNet applies the downsampling to that block. Therefore, in the figures, point-based classification result has fewer points than the ground truth.

6. Conclusion and Future Work

In this study, we aimed to classify ramps in environments like reference test areas via a deep learning technique. Since the walls and terrain carry important semantic information for robot navigation, they were also considered. We applied deep learning

architecture PointNet that accepts point cloud as input for semantic classification of terrain, walls and especially ramps in search and rescue environments. To the best knowledge of the authors, there is no dataset for NIST standard ramps. Thus, ESOGU RAMPS dataset is constructed. The experimental results are analyzed both metric and visual. The result shows that the PointNet is capable to classify terrain and walls accurately. The inclined and flat ramps are rarely confused with walls. For future work, we plan to investigate the PointNet++ architecture to reduce the number of erroneous points. Also, we have a test environment which is similar to NIST test arenas in ESOGU Laboratory Building and we plan to construct a real dataset from this test environment. Then, the dataset will be applied to the PointNet and the PointNet++.

References

[1] Kitano, Hiroaki & Tadokoro, Satoshi, "RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems," AI Magazine, 22. 39-52, 2001.

[2] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro and Y. Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2003, pp. 3396-3403, 2003.

[3] S. Raymond, S. Schwertfeger and V. Arnoud, "16 Years of RoboCup Rescue," Künstliche Intelligenz, 30(3), pp. 267-277, 2016.

[4] https://www.nist.gov/sites/default/files/documents/el/isd/ks/DHS_NIST_ASTM_Robot_Test_Methods-2.pdf

[5] http://gazebosim.org/

[6] https://www.ros.org/

[7] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 77-85,2017.

[8] E., Grilli, F. Menna, & F. Remondino, "A review of point clouds segmentation and classification algorithms," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42(2W3), 339–344. issn: 16821750, 2017.

[9] T. Rabbani, F. Van Den Heuvel, & G. Vosselmann, "Segmentation of point clouds using smoothness constraint," International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36(5), pp. 248-253, 2006.

[10] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 2, pp. 167-192, March 1988.

[11] M. A. Fischler, & R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, Vol. 24(6), pp. 381-395, 1981.

[12] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, Vol. 13(2), pp. 183-194, 1991.

[13] G. Lavoue', F. Dupont, and A. Baskurt, "A new CAD mesh segmentation method, based on curvature tensor analysis," Computer-Aided Design, Vol. 37(10), pp. 975-987, 2005.

[14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.

[15] B. Kaleci, "İç ortamlarda anlamsal tabanlı keşif algoritmalarının geliştirilmesi," Ph. D Thesis, Dept. Elec. Electronic Eng., ESOGU Univ., Turkey, 2016.

[16] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), Manila, 2013, pp. 225-230. doi: 10.1109/RAM.2013.6758588

[17] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," In Proc. ICCV, to appear, 2015.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. "3D ShapeNets: A deep representation for volumetric shapes," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1912-1920.

[19] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 922-928.

[20] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas. "Fpnn: Field probing neural networks for 3D data," arXiv preprint arXiv:1605.06240, 2016.

[21] J. Huang and S. You, "Point cloud labeling using 3D Convolutional Neural Network," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 2670-2675.

[22] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda and M. M. Bronstein, "Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5425-5434.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[24] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," in IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18-42, July 2017.

[25] J. Masci, D. Boscaini, M. M. Bronstein and P. Vandergheynst, "Geodesic Convolutional Neural Networks on Riemannian Manifolds," 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, 2015, pp. 832-840.

[26] C. R. QI, L. YI, H. SU, L. J. GUIBAS, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," arXiv preprint arXiv:1706.02413, 2017.

[27] F. Engelmann, T. Kontogianni, A. Hermans and B. Leibe, "Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds," 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017, pp. 716-724.

[28] http://wiki.ros.org/hector_nist_arenas_gazebo