

COMPARISON OF DEEP LEARNING TECHNIQUES FOR DETECTION OF DOORS IN INDOOR ENVIRONMENTS

Burak KALECİ^{1*}, Kaya TURGUT²

¹Eskişehir Osmangazi Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü

ORCID No : <https://orcid.org/0000-0002-2001-3381>

²Eskişehir Osmangazi Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü

ORCID No : <https://orcid.org/0000-0003-3345-9339>

Keywords

Door Detection
Point Cloud Data
Deep Learning
Door Dataset

Abstract

In indoor environments, the detection of doors (open, semi-opened, and closed) is a crucial task for a variety of fields such as robotics, computer vision, and architecture. The studies that are addressed the door detection problem can be divided into three major categories: 1) closed doors via visual data, 2) open doors via range data, and 3) open, semi-opened, and closed doors via point cloud data. Although some successful studies have been proposed being detected doors via visual and range data under specific circumstances, in this study, we exploited point cloud data due to its ability to describe the 3D characteristic of scenes. The main contribution of this study is two-fold. Firstly, we mainly intended to discover the potential of point-based deep learning architectures such as PointNet, PointNet++, Dynamic Graph Convolutional Neural Network (DGCNN), PointCNN, and Point2Sequence, in contrast to previous studies that generally defined a set of rules depending on the type of door and characteristics of the data. Secondly, the OGUROB DOORS dataset is constructed, which contains point cloud data captured in the GAZEBO simulation environment in different robot positions and orientations. We used precision, recall, and F1-score metrics to analyze the merit and demerit aspects of these architectures. Also, some visual results were given to describe the characteristics of these architectures. The test results showed that all architectures are capable of classifying open, semi-opened, and closed doors over 98% accuracy.

İÇ ORTAMLARDA KAPILARIN TESPİTİ İÇİN DERİN ÖĞRENME TEKNİKLERİNİN KARŞILAŞTIRILMASI

Anahtar Kelimeler

Kapı Bulma
Nokta Bulutu Verisi
Derin Öğrenme
Kapı Veri Kümesi

Öz

İç ortamlarda kapıların (açık, yarı açık ve kapalı) tespit edilmesi robotik, bilgisayarlı görü ve mimari gibi çok çeşitli uygulama alanlarında kritik bir görevdir. Kapı tespiti problemine çözüm bulmaya çalışan çalışmalar üç temel kategoriye ayrılabilir: 1) görsel veri ile kapalı kapılar, 2) mesafe verisi ile açık kapılar ve 3) nokta bulutu verisi ile açık, yarı açık ve kapalı kapılar. Kapıları görsel ve mesafe verisi ile bazı belirli şartlar altında başarılı bir şekilde bulan yöntemler önerilmiş olsa da bu çalışmada sahnelerin 3B karakteristiğini anlatma kabiliyeti sebebiyle nokta bulutu verisi kullanılmıştır. Bu çalışmanın iki temel katkısı bulunmaktadır. Birincisi, kapının tipi ve verinin karakteristiğine bağlı olarak genellikle bir kurallar kümesi tanımlayan önceki çalışmalardan farklı olarak PointNet, PointNet++, Dinamik Çizge Erişimsel Sinir Ağları (DGCNN), PointCNN ve Point2Sequence gibi nokta tabanlı derin öğrenme mimarilerinin potansiyelinin keşfedilmesini amaçlanmıştır. İkincisi, GAZEBO benzetim ortamında farklı

* Sorumlu yazar; e-posta : burakaleci@gmail.com



Bu eser, Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) hükümlerine göre açık erişimli bir makaledir.

This is an open access article under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>).

robot konum ve yönelimlerinden elde edilen nokta bulutlarından oluşan OGUROB DOORS veri kümesi oluşturulmuştur. Bu mimarilerin olumlu ve olumsuz yönlerini analiz etmek için kesinlik, duyarlılık ve F1 skor ölçütlerini kullandık. Buna ek olarak, mimarilerin karakteristiklerini ortaya koymak amacıyla bazı görsel sonuçlar verilmiştir. Test sonuçları bütün mimarilerin açık, yarı açık ve kapalı kapıları %98 üzerinde bir başarı ile sınıflandırabildiğini göstermiştir.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 01.03.2021	Submission Date	: 01.03.2021
Kabul Tarihi	: 27.09.2021	Accepted Date	: 27.09.2021

1. Introduction

The door detection problem has been studied since the beginning of the 2000s and is still one of the hot topics in many research areas. The main reason for that is the doorway locations separate indoor environments into different sections. Hence, identifying doors is critical for a variety of applications. For example, being aware of a doorway location could promote the autonomy of robots. In localization problem, doorways can be used as landmarks to decrease localization errors (Ehlers, Stuede, Nuelle, and Ortmaier, 2020). Determining doorway locations and traversing narrow passages may help automated wheelchair platforms and robots' fully autonomous navigation tasks (Kakillioglu, Ozcan, and Velipasalar, 2016; Derry and Argall, 2013). Also, classifying the status of the doors as open, semi-opened, and closed can improve the performance of humanoid robots for service tasks (Meeussen, Wise, Glaser, and Chitta, 2010). Besides, door detection is also crucial for Building Information Model (BIM) applications such as generating up to date models of large buildings (Jung, Stachniss, Ju, and Heo, 2018), for emergency cases, automatic extraction of IndoorGML models and segmentation of walkable spaces (Staats, Diakit , Vo te, and Zlatanova, 2019; Flikweert, Peters, D az-Vilari o, Vo te, and Staats, 2019), semantic mapping, and space subdivision (Nikoohemat, Peter, Elberink, and Vosselman, 2017; Zheng, Peter, Zhong, Oude Elberink, and Zhou, 2018).

The previous studies related to the door detection problem generally examined closed and open door situations. These studies separated into three categories depending on the type of door (closed or open) and sensor employed to detect doorway locations. In the first category, the studies aimed to identify closed doors with visual data. These studies suffer from two main drawbacks: 1) The distance and angle between robot and door frame seriously affect the performances of these studies because their solution depends on the pixels of visual data. 2) The success of these methods thoroughly relies on lightening conditions of the environments since they try to detect doorway locations using the neighboring pixels' similarity (Yang and Tian, 2010; Murillo, Ko eck , Guerrero and Sag es, 2008). In the second category, the studies utilized range data acquired with ultrasonic sensors or 2D laser range

finders to detect open doors. They generally attempt to find a void area corresponding to the dimensions of a door when the robot is located within or in the proximity of a door frame (ElKaissi, Elgamel, Bayoumi, and Zavidovique, 2006). The main disadvantage of these studies is having a high false-positive ratio. Besides, some approaches in the previous studies used both visual and range data to detect doorway locations (Hensler, Blaich, and Bittel, 2010; Meeussen, Wise, Glaser, and Chitta, 2010). In the last category, the studies used point cloud data and/or depth images, which have been frequently preferred in recent years for robotic and BIM applications, to detect open, semi-opened, and closed doors. These studies are not affected by varying light conditions of the environment (Rusu, Meeussen, Chitta, Beetz, 2009). Besides, the robot does not need to be located within or proximity of the door frame to recognize an open door (Panzarella, Schwesinger, and Spletzer, 2016). Similar to studies that used visual data, the angle between robot and door frame could affect the success of these methods. As the angle narrows, the door detection rate generally decreases for studies in the last category (Souto, Castro, Goncalves, and Nascimento, 2017; Derry and Argall, 2013).

In this study, we opted to use point cloud data to detect open, semi-opened, and closed doors. The primary reasons for that are the ability of point cloud data to describe 3D characteristics of scenes, and making decisions about some critical tasks such as path planning, navigation, and opening the closed or semi-opened doors when the robot placed further from the doors. As explained in the related work section, existing studies generally have designated a set of rules that rely on the type of door and data. Although some of these studies were successful under specific circumstances, the generalization ability of these studies was inadequate. To deal with this problem and to classify doors as closed, semi-opened, and open doors independent from the distance and angle between robot and door frames, we discovered the potential of point-based deep learning architectures such as PointNet, PointNet++, DGCNN, PointCNN, and Point2Sequence. Besides, we noticed that the datasets in the literature are insufficient due to the lack of the number of samples and type of doors. For that reason, we built the OGUROB DOORS dataset, which contains 1353 point cloud data

for open, semi-opened, and closed door types, captured in the GAZEBO simulation environment in different robot positions and orientations. The test results indicated that all architectures were capable of identifying open, semi-opened, and closed doors.

The rest of the study is organized as follows: In section 2, related works are reviewed. The point-based deep learning techniques are briefly explained in section 3. The OGUROB DOORS dataset is described in section 4. We present quantitative and visual results in section 5 and concluded with section 6.

2. Related Works

In this section, we divided studies related to the door detection problem into four categories. We first briefly discussed some milestone approaches that utilized visual and range data. Then, a comprehensive review for point cloud-based and learning-based studies were presented.

2.1. Vision-Based Approaches

Andreopoulos and Tsotsos (2008) proposed a computer-controlled wheelchair system that automatically detects doors and determines the position of a door frame with the aid of visual data. They assumed that a door frame is a set of corners combined by line segments. They then extracted corners and line segments and defined a set of rules to model the door frame. In a similar work, Yang and Tian presented a method that detects the geometric model of a door (corners and line segments) with boundary features. For that reason, their method was robust against varying colors and textures. However, the method could not distinguish between open and closed doors since it only considers boundaries. Kim, Cheong, Kim, and Park (2011) first recognized doorknobs considering the viewpoint of a robot and the geometric position of doorknobs. They determined a region of interest for doorknobs so that they decreased computational complexity and improved the performance of the object recognition scheme. Then, door locations were specified through doorknobs. Unfortunately, their methods were available for closed doors. Sekkal, Pasteau, Babel, Brun, and Leplumey (2013) aimed to detect doors situated in a corridor via a monocular camera. They first determined boundaries between wall and floor. Then, they obtained a 3D model of walls by using these boundaries and searched the door location in wall planes.

2.2. Range-Based Approaches

ElKaissi et al. attempted to detect door edges using two successive laser readings. They defined a set of rules over these laser readings. They did not apply Hough transform since it has high computational complexity

and inappropriate for real-time applications. Nieuwenhuisen, Stückler, and Behnke (2010) emphasized that doors are movable objects. They first employed a traditional occupancy grid generation approach to obtain map of the environment. Then, they defined doors as line segments that can move around the door frame's vertical axis. Lastly, they searched mismatch between the map and the current laser reading to detect movable doors. Gillham, Howells, Spurgeon, Kelly, and Pepper (2013) used an infrared sensor to detect doors. They extracted two types of features from high and lower-resolution raw data to feed various classifiers. They performed experiments with data acquired from varying angles (from -45 to 45 degrees) between the sensor and door frame to represent their method's efficiency. Kaleci, Şenler, Dutagaci, and Parlaktuna (2015) proposed a method for open doors. They noticed that a bottleneck characteristic occurs when the robot is situated within a door frame. Then, they defined a set of rules to detect that bottleneck characteristic from a laser scan. In this way, they aimed to decrease the false-positive ratio, which is the main drawback of range-based door detection methods. The extensive experiments showed that their method was able to significantly reduce the false-positive ratio while the door detection rate over 90%.

2.3. Point Cloud Based Approaches

In recent years, the studies that address Building Information Model (BIM) applications have been gained popularity. These studies generally use Mobile Laser Scanners (MLS) to obtain point cloud data together with trajectories of MLS. The data is segmented to dominant planar surfaces such as walls, terrain, and ceiling. Lastly, open doors are detected by searching in wall planes to separate different parts of indoor environments, such as rooms and corridors. For example, Budroni and Böhm (2010) applied the volume sweep reconstruction method to determine dominant planar surfaces. Then, they projected wall planes into a 2D data structure and searched a gap along the wall plane with the aid of a sweep line parallel to the terrain surface at 1m height. Nikoohemat et al. proposed a graph-based method to obtain planar surfaces. Then, they represented wall planes with an octree data structure and examined empty voxels to detect open doors. Wang, Xie, and Chen (2017) utilized Delaunay triangulation, alpha-shape algorithm, and MLS trajectory for void detection in wall planes that correspond to an open door. Díaz-Vilariño, Verbree, Zlatanova, and Diakité (2017) proposed a different approach without segmenting wall planes to detecting doors. They first calculated a vertical profile during the trajectory. Then, they examined the number of points and average z-coordinates of the vertical profile to determine open door locations. Jung et al. created a histogram according to z coordinates of points

to segment terrain and ceiling planes. Then, they applied the Douglas-Peucker method to wall planes for determining void regions in a rectangle shape. Lastly, they tested the void region for dimensions of a door frame. Cui, Li, Yang, Xiao, Chen, and Dong (2019) specified predetermined horizontal slices at different heights from the ground plane to identify door candidates. Then, point clouds corresponding to these candidate locations were converted into binary images and applied Hough transform to locate the door frame. Staats et al. transferred the point cloud data into voxel space. Then, they used the trajectory of the MLS to determine open doors. They controlled both vertical and horizontal voxels in the direction of trajectory to decrease the false-positive rate.

The studies that aim to develop autonomous wheelchair systems also interest in detecting open doors. For example, Derry and Argall first extracted wall planes with RANSAC method. Then, they performed a horizontal search along the wall plane to find a void region corresponding to door locations. They conducted extensive experiments with point cloud data acquired different positions and orientations. The angles and distance between the sensor and the door frame varying from 20 to 75 degrees and from 1.8 to 4.2 meters, respectively. The experimental results indicated that the door detection rate decreases below 70% as the angle increases. Burhanpurkar, Labbe, Guan, Michaud, and Kelly (2017) separated walls in the point cloud data and filtered points according to the z-axis to obtain the midsection of the wall. In this way, they aimed to develop a method robust against occlusions. They considered both the protruding and non-protruding hinged doors. Experiments are performed with different approaching angles and distances to the door frame. Panzarella et al. used two RGB-D cameras to gather point cloud data. The data was transferred into depth images, and traditional 2D image processing approaches were applied to depth images to detect discontinuities that occur at door locations. Then, convolution was employed the pixels corresponding discontinuities for extracting door edges. Besides, the method utilized the Harris operator to distinguish corners. Lastly, the edges and corners were connected with the line intersection technique to calculate door frame locations.

Kakillioglu et al. transferred wall planes into depth images and applied a histogram-based method to the images for searching empty regions resembling open doors. Then, they exploited Aggregate Channel Features to train visual data so that they decreased the false positive rate. Michailidis and Pajarola (2017) first decided on wall candidates with an occlusion-aware method. Then, they utilized the alpha-shapes approach to calculate the general shape of the wall candidates. The boundary lines of walls were calculated with RANSAC, and they construct a graph (in other words, 2D cell complex), which is the input of the graph-cut

optimization problem. The doors were classified as solving the problem that determines cells belong to door regions. Kaleci and Turgut (2020) proposed a rule-based approach for locating open doors. In this approach, they aimed to identify four corners of doors while searching neighbors of unmeasured (NaN) readings. Hence, they determined boundary edges of open doors. Lastly, they controlled the dimensions of these edges to locate doors.

Borgsen, Schöpfer, Ziegler, and Wachsmuth (2014) aimed to detect closed doors using normals of points. They first decided on planes through the region growing method, and for each plane, they extracted features such as the number of points, width, and height of the planes. They used normals to identify the edges of door frames. Lastly, a confidence value was estimated while considering ideal and calculated door dimensions in terms of difference and standard deviation. A similar work detected closed doors presented by Nagahama, Takeshita, Yaguchi, Yamazaki, Yamamoto, and Inaba (2018). They first passed the point cloud data from the bilateral filter, and they calculated the normals of points. The points were clustered according to the normals, and all planes perpendicular to the ground plane were extracted. The edges of doors were determined by applying the Canny operator. Lastly, the edges were controlled whether they form a rectangular region or not. Quijano and Prieto (2016) used contextual and geometric relationships of planes to detect open and closed doors recessed into walls. They applied RANSAC to wall planes for line segmentation. They removed lines that are not parallel to the ground and wall plane. The remaining lines were utilized to determine door locations. Bayram, Kolaylı, Solak, Tatar, Turgut, and Kaleci (2019) aimed to detect semi-opened and closed doors. In this method, the authors considered the narrow angles problem of visual and point cloud-based approaches. They proposed a parameter tuning approach that depends on the angle and distance between the door frame and sensor. Lastly, they used the output of the parameter tuning approach for dimensions of doors.

Rusu et al. proposed a method to locate open, semi-opened, and closed doors. They first applied downsampling to point cloud data for obtaining an appropriate method for real-time applications. Then, they calculated point normals and they clustered point cloud data depending on normals. They employed RMSAC (a variation of RANSAC) method for each cluster to fit a plane model. Lastly, they extracted features such as width, height, maximum, and minimum points of planes to determine the best door candidates. Souto et al. aimed to detect open, semi-opened, and closed doors for the localization of robots. Similar to Rusu et al., the authors first filtered the point cloud data and determined wall planes. Then, they extracted points that belong to the boundary of doors. Lastly, they considered

the position of these points to detect doors. Quintana, Prieto, Adán, and Bosché (2018) transferred point cloud data into an octree data structure to detect open doors. Then, they extracted dominant planar surfaces such as walls, ground, and ceiling with the aid of the octree. Simultaneously, they classified voxels as empty and non-empty. They examined empty voxels lie on wall planes to locate open doors. If the method does not find empty voxels for open doors, it tries to identify closed doors with visual and point cloud data. Besides, they performed a preprocessing step to avoid decreasing the true positive rate due to varying lightened conditions.

2.4. Learning-Based Approaches

Learning-based approaches are divided into two groups: machine learning and deep learning. When the studies are examined, it is seen that visual or point cloud data is used to detect and classify doors. Murillo et al. proposed a door detection method that takes into account both geometric and appearance properties of a door. They defined the geometric model of a door with a few parameters, and they used a learning approach for appearance properties. Then, a probabilistic approach calculated geometric and appearance likelihood to determine doors. Hensler et al. utilized visual and range data to detect doors. They solved the decision problem using the AdaBoost algorithm after extracting the door's weak features such as width, gap, color, etc. Ye and Qian (2018) proposed a method to recognize structural objects (e.g., stairways, doorways, walls, and so on) in the indoor environment for helping the navigation of visually impaired people. After removing the planar regions in the point cloud model, inter-plane relationships were extracted for every two planes revealed. Based on these relationships, planes were classified with the Gaussian Mixture Model with six high level features.

Recently, 2D or 3D deep learning approaches have also started to be used for door detection as well as in many areas. Chen, Qu, Zhou, Weng, Wang, and Fu (2014) applied Convolutional Neural Network (CNN) to visual data to detect closed doors. They significantly decreased the false positive rate since they used a dataset that contains numerous negative samples. The main drawbacks of this work were high false negatives and being only available for closed doors. Othman and Rad (2020) found the doorways using images taken from a monocular camera on the humanoid robot. Their method decided whether there is an open door or no-door in the image with the CNN approach. For the door transition points, the pixels are searched on the depth map obtained from the image with deep learning approaches. Bersan, Martins, Campos, and Nascimento (2018) used 2D CNN-based YOLO (Redmon, Divvala, Girshick, and Farhadi, 2016) object identifier and 3D model-based segmentation algorithm for door object

during semantic metric mapping of the environment. The point cloud was obtained using the depth information of the pixels in the bounding box with YOLO, and only the points belonging to the door were extracted using RANSAC. Similar algorithms for detecting the doors and door handles have been applied to humanoid robots (Arduengo, Torras, and Sentis, 2019; Llopart, Ravn, and Andersen, 2017). In another study (Beraldo, Termine, and Menegatti, 2019) that uses YOLO, the door positions and center of apertures for navigation of mobile robots were decided, and their statues were determined as open or closed. A depth-based filter was used to reduce false predictions.

3D deep learning approaches have also been used for door detection problems. Since the points are permutation invariant and there is no defined structural relationship between points in the point cloud, it is not directly applied classical convolutional approaches. With the proposal of PointNet (Qi, Su, Mo and Guibas, 2016), which is the first point-based deep learning architecture, studies on point-based deep learning architectures and application areas have increased. Koo, Jung, and Yu (2021) demonstrated the applicability of the 3D geometric deep learning approach MVCNN (Su, Maji, Kalogerakis and Learned-Miller, 2015) and PointNet independently for the door and wall detection in the BIM model. In the MVCNN architecture, 2D images were rendered from different angles around the 3D object, and after the features were extracted with classical CNN layers, the dominant ones were taken into account with the image pooling method. The success rate of single door, double door, sliding door, and revolving door categories created from CAD models is over 90% for MVCNN and PointNet. Ramôa, Alexandre, and Mogo (2020) proposed a method to classify closed, open and semi-open doors by combining 2D semantic segmentation and 3D classification methods. For 2D semantic classification, the bounding box of the door is extracted using FastFCN (Wu, Zhang, Huang, Liang and Yizhou, 2019) and FC-HardNet (Chao, Kao, Ruan, Huang, and Lin, 2019) methods. PointNet classifies point cloud data obtained from the depth information of the pixels in the bounding box. While an average accuracy of 49.4% was obtained using the two methods together, an average accuracy of 43.3% was obtained performing only the PointNet architecture.

As mentioned above, the vision-based approaches generally extracted corners and line segments that connect corners through traditional 2D computer vision methods. Then, they defined a set of rules depending on the positions of corners and line segments to detect doors. Similarly, range-based approaches attempted to recognize door characteristics relying on a set of rules over laser readings. The studies that addressed open door detection via point cloud data generally first segment wall planes and then searched a void region in the wall planes. Semi-opened and closed-door detection

methods preferred to using normal and color attributes of point cloud data. These studies suffered a lack of generalization ability since they used a set of rules depending on the type of door and data characteristic. On the other hand, learning-based approaches frequently applied CNN approaches to visual data or depth images, which ability to describe 3D characteristics of a scene is limited. In this study, apart from the previous studies, we preferred to discover the potential of point-based deep learning architectures for door detection. For that reason, we examined the comprehensive review of 3D deep learning approaches presented by Guo, Wang, Hu, Liu, and Bennamoun (2019). Then, we decided to employ five approaches, which belong to different categories in the review: 1) Pioneer architecture PointNet (Qi, 2016), 2) PointNet++ (Qi, Yi, Su and Guibas, 2017), which takes into account local regions hierarchically, 3) graph-based architecture DGCNN (Wang, Sun, Liu, Sarma, Bronstein and Solomon, 2019), 4) Convolution-based PointCNN (Li, Bu, Sun, Wu, Di and Chen, 2018), and 5) RNN-based Point2Sequences (Liu, Han, Liu, and Zwicker, 2019).

3. Methods

In this section, we briefly explain the significant properties of PointNet, PointNet++, DGCNN, PointCNN, and Point2Sequence point-based deep learning architectures.

In this study, research and publication ethics were followed.

3.1. PointNet

PointNet is a pioneering architecture for point-based deep learning approaches. It evaluates points independently and individually, which means that PointNet does not consider neighbors of points in the feature extraction process. Multi-Layer Perceptrons (MLPs) with shared weights are applied to x, y, and z coordinates of points and other attributes such as color, normals, and curvature can be used if they are available to obtain features of points. In order to generate the global feature maximum pooling method which is called symmetric function is introduced. Thanks to the method, the global feature is independent in the order of points. With this function, the global feature is extracted by taking the maximum of the feature channels of the points. The global feature is mapped for categories with classifier layers.

3.2. PointNet++

PointNet++ architecture is an extended version of PointNet. Similar to PointNet, the architecture extracts features for each point. However, PointNet++ takes

account into neighbors during the feature extraction process. To do that, local regions are created for center points which are sampled uniformly depending on a predefined radius or K-nearest neighbors. Then, PointNet is applied to x, y, and z coordinates of points in each local region defined around the selected center points. Each center point feature is obtained by summarizing the features of the neighbor points using the symmetric function. In the successive layer, the local regions are grown in a hierarchical manner so that the architecture generates local features that are able to describe the characteristic of the scene. In all layers, local areas are constructed according to the spatial coordinates. The global features are extracted just like PointNet with the aid of symmetric function and are projected onto the category space through the classifier layer.

3.3. Dynamic Graph CNN (DGCNN)

DGCNN architecture belongs to the graph-based category. Similar to PointNet++ architecture, DGCNN exploits neighbors of points during the local feature extraction process. Local regions around each center point in the point cloud consist of its K nearest neighbors. The distances between points to form local regions are calculated in spatial and feature space for the first and successive layers, respectively. In contrast to PointNet++, the local regions are not grown hierarchically. After local regions are settled, for each point in the point cloud, a graph is constructed. The points in each local region are called nodes, and the distances between the center node and each neighbor node are called edges. In the first layer, edge weights are relative x, y, and z coordinates of neighbor nodes to the center node. Apart from the PointNet++, for successive layers edge weights are determined from features. The features are summarized with the symmetric function after extracting the edge features with weights-shared multi-layer perceptrons over the edges. In each layer, the graph structure is dynamically updated according to the extracted feature. This process explained above is introduced as EdgeConv operator. DGCNN architecture is a combination of EdgeConv and PointNet. After EdgeConv extracts the features, the features are fed as input to architecture similar to PointNet.

3.4. PointCNN

PointCNN architecture aims to evaluate points together with their neighbors in the feature extraction process, unlike the architectures explained before that are considering points individually. To do that, the CNN approach is adapted to the point cloud since the point cloud data is permutation invariant, and a direct convolution approach cannot be applied. X-Conv convolution operator is introduced. X-Conv operator

learns a transformation matrix for each local region around selected points. Before the convolution is applied to the local region, the point features are weighted and ordered in a canonical form with the aid of the transformation matrix. The proposed X-Conv operator is added hierarchically to extract features and local areas are enlarged in successive layers like PointNet++. The classifying layer provides the relationship between the global features and categories.

3.5. Point2Sequence

Point2Sequence architecture is based on PointNet++, however, it is aimed to encode the contextual information of local regions. This architecture establishes multi-scale concentric local regions around each center point and then extracts each region features like PointNet++. The correlation between these different scale features for each center point is learned with recursive neural networks. LSTM (Long Short-Term Memory) structure is used to expose relation of features. Then, with the attention approach, hidden states of the LSTM are highlighted and each local region feature is obtained by aggregating different scale features with learned context vector. The global feature is extracted from the local region features by the MLPs and symmetric function. Unlike PointNet++, there is no successive layer since they merge multi-scale feature in one layer.

4. The OGUROB DOORS Dataset

In previous works, the studies that have been addressed indoor modeling generally used publicly available datasets such as ISPRS Benchmark on Indoor modeling (Khoshelham, Vilariño, Peter, Kang, and Acharya, 2017) for door detection. Since these datasets generally contain a rare number of samples corresponding to door locations, they are not appropriate for examining the performance of door detection methods. On the other hand, Derry and Argall constructed a dataset that includes approximately 100 samples for each open-door type. They captured these samples from different positions and orientations. Similarly, Burhanpurkar et al. built a dataset for their wheelchair application. The wheelchair was located at different positions within the range of 1 and 3 meters. The orientation of the wheelchair was generally toward the door frame. The dataset consists of 96 open door samples. Panzarella et al. created an open door dataset with 100 samples similar to Burhanpurkar's work. Kaleci and Turgut introduced a simple open-door dataset gathered via the Gazebo simulation environment. Apart from the previous open-door datasets, the robot was placed not only toward the door frame but also at narrow angles. Borgsen et al. constructed a colored closed-doors dataset that includes doors with different colors and a

variety of materials. The samples were captured from 7 different positions, at which the sensor is located 0.9, 1.4, and 2.2 meters distant from the door frame. Also, the angle between the sensor and the door frame was -20, 0, and 20 degrees. Quintana et al. introduced two point cloud datasets (simulated and real) for semi-opened and closed doors. They used Blensor to obtain a synthetic model that includes five doors with different opening angles. There are 35 door samples in the real dataset. They separated the dataset into five categories, from simple to complex. Rusu et al. constructed a door dataset for open, semi-opened, and closed doors. They handled more than 50 situations from different angles.

It is obvious that existing door datasets are inappropriate, especially deep learning architectures that require a large number of samples in the training phase. Additionally, the samples that are placed in these datasets were generally positioned toward the door frame. However, we intend to discover deep learning architectures' potential when the robot is located in all possible positions and orientations toward a door frame. For this reason, we built a new dataset, namely, OGUROB DOORS, through the Gazebo simulation environment (Gazebo, 2021) and Robot Operating System (ROS, 2021) robot control interface. To construct the dataset, a Pioneer 3-AT mobile robot with an Asus Xtion Pro RGB-D camera was launched to the simulation environment. The parameters of the camera were kept as default. In other words, we did not add noise to the measurements. We selected the minimum and maximum distance between robot and door frame as 1.5 and 3.5 meters, respectively. Then, we positioned the robot in uniformly distributed locations with a 10 cm interval in each dimension. The orientation of the robot was determined randomly. For each robot location, we captured point cloud data for open, semi-opened, and closed doors. As a result, we obtained 1353 samples for each type of door. Then, we separated the samples depending on the distance between the robot and the door frame into four equal parts with a step size of 0.5 meters. Besides, we categorized the samples into five regions according to the angle between the robot and the door frame. Figure 1 shows the position of the door frame and robot locations with a large black circle and small filled circles, respectively. In the figure, cyan semi-circles depict boundaries of the parts according to distance. Also, pink line segments separate regions depending on the angle. In orange and blue regions, the angle between robot and door frame varies from 0° to 30° and from 150° to 180°, respectively. The angle alters from 30° to 60° and from 120° to 150°, respectively, in red and yellow regions. Lastly, in the green region, the angle changes between 60° and 120°.

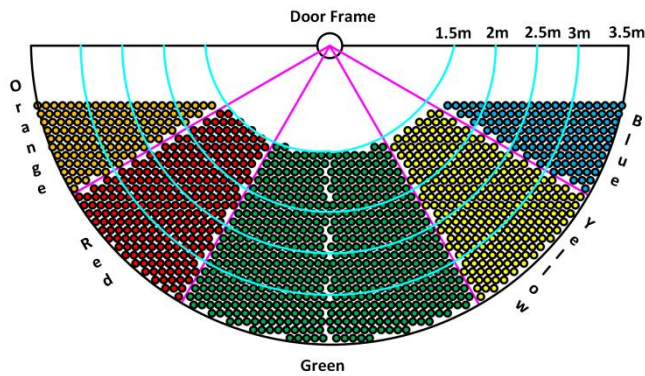


Figure 1. Robot locations. Orange (0° - 30°), Red (30° - 60°), Green (60° - 120°), Yellow (120° - 150°), and Blue (150° - 180°)

Table 1 shows the number of samples depending on angle and distance between robot and door frame.

Table 1

Number of Samples Depending On Angle And Distance					
Angle	Dist				Total
	1.5-2m	2-2.5m	2.5-3m	3-3.5m	
Orange	9	21	41	59	130
Red	50	55	70	91	266
Green	117	115	140	189	561
Yellow	50	55	70	91	266
Blue	9	21	41	59	130
Total	235	267	362	489	1353

Figure 2 indicates samples for open, semi-opened, and closed doors belong to orange, red, green, yellow, and blue regions with different distances. In the figure, the first two columns indicate examples of open doors. RGB image and point cloud data of the corresponding scene are given in the first and second columns, respectively. Similarly, the third and fourth columns depict examples of semi-opened doors, and the last two columns show closed-doors examples. OGUROB DOORS dataset is publicly available and can be downloaded from (OGUROB DOORS, 2021).

5. Experimental Works

5.1. Experimental Setup

We employed PointNet, PointNet++, DGCNN, PointCNN, and Point2Sequence point-based architectures to

classify open, semi-opened, and closed doors. The architectures were implemented with Python programming language using TensorFlow library (2021). Point-based deep learning architectures are needed to employ a preprocessing step to point cloud data for classification problems. The point cloud data includes NaN values for the points are out of measurement range. Before fed these architectures, NaN points are excluded. Besides, these architectures receive a fixed number of points. Therefore, the number of points in a point cloud is chosen as 4096, and the farthest point sampling algorithm is applied to obtain uniformly distributed 4096 points. The resultant point clouds were used as an input of these architectures. OGUROB DOORS dataset was employed in the experiments. In the dataset, each point cloud data contains points represented with only x, y, and z coordinates. For each class, approximately 70%, 10%, and %20 samples were chosen randomly for training, validation, and test stages considering angles and distances between door frame and the robot. We used default parameters for all architectures in the training stage, and 5-fold cross validation in the test stage.

We assessed the performance of the architectures through precision, recall, and F1-score metrics. If a sample that belongs to a positive class is classified correctly, it is called a true positive (TP), or incorrectly classified is called a false positive (FP). If a sample that belongs to a negative class is incorrectly classified as a positive class, it is called a false negative (FN). The ratio of true positive of a class to the total number of samples of that class is called recall (Equation (1)). Precision defines the ratio of true positive of a class to the total number of classified samples for that class (Equation (2)). F1 is the harmonic mean of precision and recall (Equation (3)). The results are the mean of 5-fold cross validation.

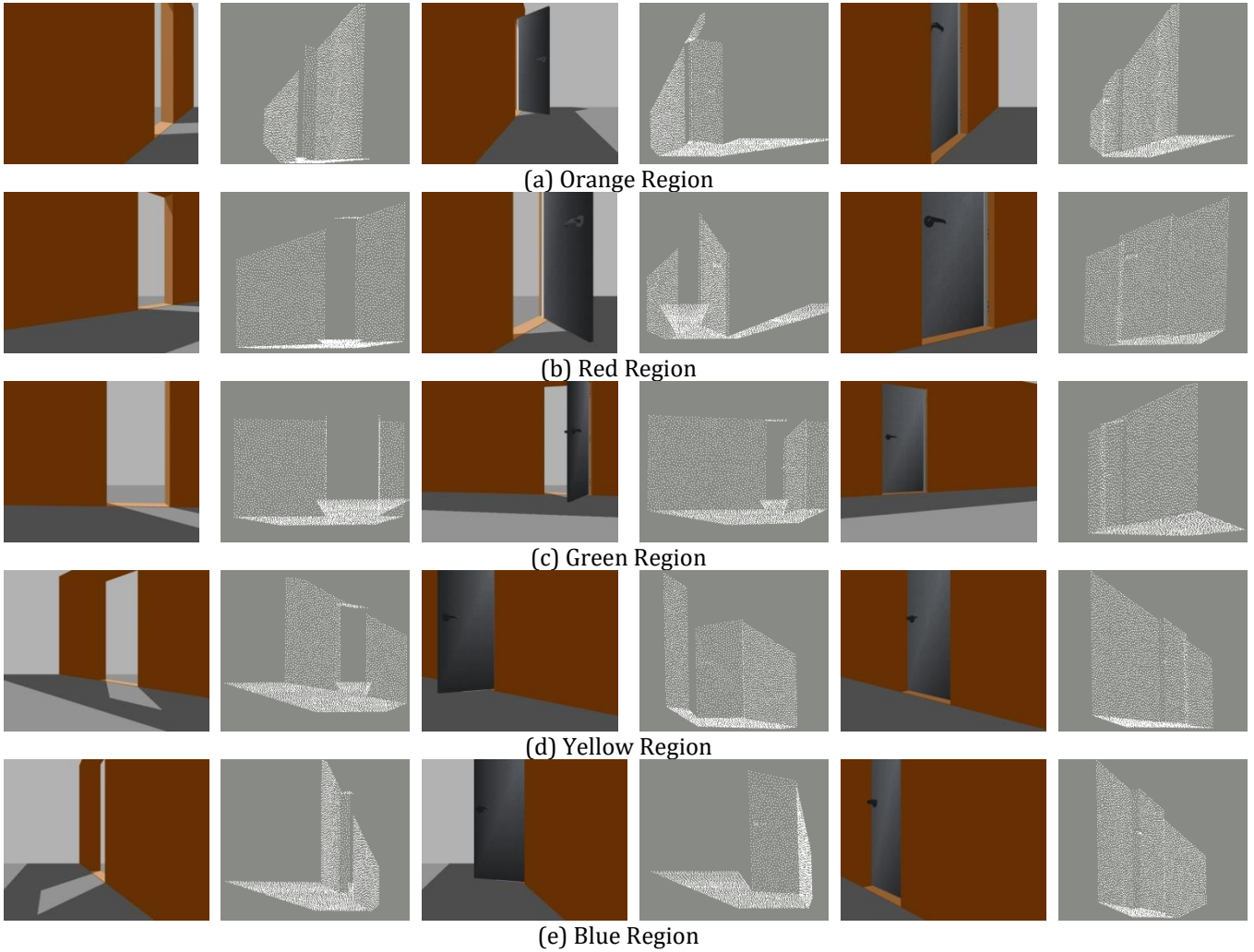


Figure 2. Samples for open, semi-opened, and closed doors belong to orange, red, green, yellow, and blue regions with different distances. The first column indicates RGB image of open doors. The second column is point cloud data of the corresponding scene. The third column indicates RGB image of semi-opened doors. The fourth column is of the point cloud data corresponding scene. The fifth column indicates RGB image of closed doors. The sixth column is point cloud data of the corresponding scene.

$$recall = \frac{TP}{TP + FP} \tag{1}$$

$$precision = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 \frac{recall * precision}{recall + precision} \tag{3}$$

5.2. Experimental Results

The 5-fold cumulative confusion matrices of models trained with five different architectures on the test dataset are given in Table 2. As seen from the table, point-based deep learning architectures are appropriate for door detection problems. Besides, the architectures identified each type of door except a few samples via the point cloud data captured when the robot is located in all possible positions and orientations toward a door frame, which indicates the generalization ability of these architectures. The recall, precision, and F1-score values of these architectures for each type of door are demonstrated in Table 3.

Table 2
5-fold cumulative confusion matrices for architectures

Method	Confusion Matrix																							
PointNet	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>open</th> <th>semi-open</th> <th>close</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Predicted label</th> <th>open</th> <td>1,340</td> <td>19</td> <td>8</td> </tr> <tr> <th>semi-open</th> <td>6</td> <td>1,331</td> <td>3</td> </tr> <tr> <th>close</th> <td>9</td> <td>5</td> <td>1,344</td> </tr> <tr> <th colspan="2"></th> <th colspan="3">Actual label</th> </tr> </tbody> </table>			open	semi-open	close	Predicted label	open	1,340	19	8	semi-open	6	1,331	3	close	9	5	1,344			Actual label		
		open	semi-open	close																				
Predicted label	open	1,340	19	8																				
	semi-open	6	1,331	3																				
	close	9	5	1,344																				
		Actual label																						
PointNet++	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>open</th> <th>semi-open</th> <th>close</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Predicted label</th> <th>open</th> <td>1,355</td> <td>5</td> <td>0</td> </tr> <tr> <th>semi-open</th> <td>0</td> <td>1,348</td> <td>2</td> </tr> <tr> <th>close</th> <td>0</td> <td>2</td> <td>1,353</td> </tr> <tr> <th colspan="2"></th> <th colspan="3">Actual label</th> </tr> </tbody> </table>			open	semi-open	close	Predicted label	open	1,355	5	0	semi-open	0	1,348	2	close	0	2	1,353			Actual label		
		open	semi-open	close																				
Predicted label	open	1,355	5	0																				
	semi-open	0	1,348	2																				
	close	0	2	1,353																				
		Actual label																						
DGCNN	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>open</th> <th>semi-open</th> <th>close</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Predicted label</th> <th>open</th> <td>1,349</td> <td>6</td> <td>0</td> </tr> <tr> <th>semi-open</th> <td>4</td> <td>1,348</td> <td>1</td> </tr> <tr> <th>close</th> <td>2</td> <td>1</td> <td>1,354</td> </tr> <tr> <th colspan="2"></th> <th colspan="3">Actual label</th> </tr> </tbody> </table>			open	semi-open	close	Predicted label	open	1,349	6	0	semi-open	4	1,348	1	close	2	1	1,354			Actual label		
		open	semi-open	close																				
Predicted label	open	1,349	6	0																				
	semi-open	4	1,348	1																				
	close	2	1	1,354																				
		Actual label																						
PointCNN	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>open</th> <th>semi-open</th> <th>close</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Predicted label</th> <th>open</th> <td>1,352</td> <td>3</td> <td>1</td> </tr> <tr> <th>semi-open</th> <td>3</td> <td>1,351</td> <td>0</td> </tr> <tr> <th>close</th> <td>0</td> <td>1</td> <td>1,354</td> </tr> <tr> <th colspan="2"></th> <th colspan="3">Actual label</th> </tr> </tbody> </table>			open	semi-open	close	Predicted label	open	1,352	3	1	semi-open	3	1,351	0	close	0	1	1,354			Actual label		
		open	semi-open	close																				
Predicted label	open	1,352	3	1																				
	semi-open	3	1,351	0																				
	close	0	1	1,354																				
		Actual label																						
Point2Sequence	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>open</th> <th>semi-open</th> <th>close</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Predicted label</th> <th>open</th> <td>1,352</td> <td>8</td> <td>0</td> </tr> <tr> <th>semi-open</th> <td>3</td> <td>1,347</td> <td>3</td> </tr> <tr> <th>close</th> <td>0</td> <td>0</td> <td>1,352</td> </tr> <tr> <th colspan="2"></th> <th colspan="3">Actual label</th> </tr> </tbody> </table>			open	semi-open	close	Predicted label	open	1,352	8	0	semi-open	3	1,347	3	close	0	0	1,352			Actual label		
		open	semi-open	close																				
Predicted label	open	1,352	8	0																				
	semi-open	3	1,347	3																				
	close	0	0	1,352																				
		Actual label																						

As seen from the table, all architectures are able to classify open, semi-opened, and closed doors over 98% recall values. Also, precision and F1-score values are over 98% for all architectures. We will give recall, precision, and F1-score values in the following subsections according to the angle and distance for each door type. Also, we will discuss why the architectures do not classify some examples with visual results.

5.2.1 Open Door Results

The quantitative and visual results for open doors are presented in Table 4 and Figure 3, respectively. As seen from the table, PointNet has a slightly lower recall value when it is compared with other architectures. The reason for that PointNet architecture extracts point's features individually and independently. In other words, it does not take into account local information. Therefore, in some scenes, PointNet does not distinguish the door frame and wall planes, and misclassified scenes occur. An example for these situations is given in Figure 3(a). DGCNN also incorrectly classified the same scene with PointNet. Although DGCNN constructs local regions and considers neighbor points in the local region during the feature extraction process, local regions are not expanded hierarchically. Besides, the local regions become consisting of the points that have the same features since the graph structure is dynamically updated according to the extracted feature in each layer. Therefore, DGCNN may not understand the characteristic of the scenes in some cases. PointCNN, unlike other architectures, evaluates points together with their neighbors in the feature extraction process. The local regions are hierarchically enlarged, and PointCNN can learn planar structures. If the scene has these planar structures, PointCNN correctly classified these scenes. However, a tiny part of a planar structure (i.e., wall) can exist in a scene; this may lead to incorrect classification of the scene. An example is given in Figure 3(b). The ellipse shows a small number of points that belong to the wall plane. Point2Sequence misclassified a few samples, and we will explain why it produces incorrect results with visual examples in the next subsection. Lastly, PointNet++ successfully classifies all samples.

Table 3

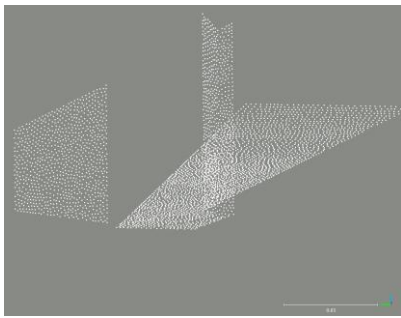
Recall, precision, and F1-score values for architectures

Door Type	Open			Semi-opened			Closed		
	Recall	Precision	F1-score	Recall	Precision	F1-score	Recall	Precision	F1-score
Method									
PointNet	98.89	98.04	98.46	98.23	99.34	98.78	99.19	98.99	99.08
PointNet++	100.00	99.63	99.82	99.48	99.85	99.67	99.85	99.85	99.85
DGCNN	99.56	99.56	99.56	99.48	99.63	99.56	99.92	99.78	99.85
PointCNN	99.78	99.71	99.74	99.70	99.78	99.74	99.93	99.93	99.93
Point2Sequence	99.78	99.41	99.59	99.41	99.56	99.48	99.78	100	99.89

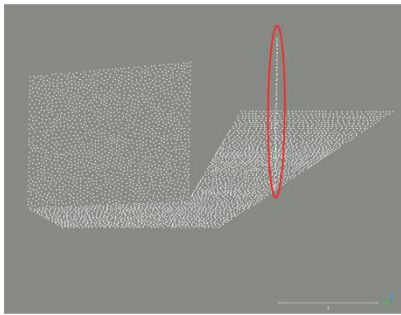
Table 4

Quantative results for open door class

A n g l e (o)	Dist (m)	N O S	Methods														
			PointNet			PointNet++			DGCNN			PointCNN			Point2Sequence		
			Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
0-30	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	4	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	92.30	95.00	93.63	100.00	100.00	100.00	98.46	98.33	99.40	100.00	100.00	100.00	100.00	100.00	100.00
30-60	1.5-2	10	96.36	98.00	97.18	100.00	100.00	100.00	100.00	98.00	98.99	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	86.49	98.18	91.96	98.33	100.00	99.16	95.26	98.18	96.70	98.33	100.00	99.16	98.33	100.00	99.16
	2.5-3	14	97.33	98.57	97.95	97.33	100.00	98.65	98.67	98.57	98.62	100.00	98.57	99.28	97.33	98.57	97.95
	3-3.5	18	96.78	98.89	97.82	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.95	100.00	99.47
60-120	1.5-2	24	99.20	99.17	99.18	98.40	100.00	99.19	99.20	100.00	99.60	98.40	100.00	99.19	98.40	100.00	99.19
	2-2.5	23	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.26	99.12	99.17	100.00	99.58
	2.5-3	28	100.00	99.29	99.64	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	38	99.00	99.47	99.24	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.49	98.95	99.22
120-150	1.5-2	10	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	14	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	18	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
150-180	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	93.33	100.00	96.55	100.00	100.00	100.00
	2-2.5	4	96.00	100.00	97.96	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	98.46	91.67	94.94	100.00	100.00	100.00	100.00	96.67	98.31	100.00	100.00	100.00	100.00	100.00	100.00

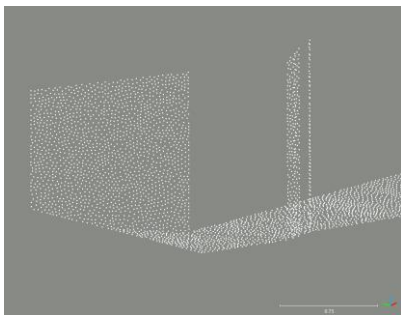


(a)

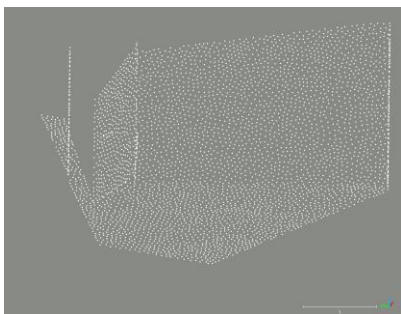


(b)

Figure 3. Examples for misclassified samples for open doors. (a) PointNet and DGCNN misclassified example, and (b) PointCNN misclassified example.



(a)



(b)

Figure 4. Examples for misclassified samples for semi-opened doors. (a) PointNet, DGCNN, and Point2Sequence misclassified example, and (b) PointNet++ misclassified example.

5.2.2 Semi-opened Door Results

The quantitative and visual results for semi-opened doors are shown in Table 5 and Figure 4, respectively. As seen from the table, similar to open door results, PointNet has an insignificantly lower recall value when it is compared with other architectures. Figure 4(a) indicates a misclassified example for PointNet, DGCNN, and Point2Sequence architectures. The reasons described for open door results can explain why PointNet and DGCNN produce an incorrect result. Point2Sequence architecture builds multi-scale concentric local regions to extract features. Then, the correlation between these different scale features is learned. Besides, some features are highlighted while some are suppressed. For that reason, in some cases like Figure 4(a), some distinguishable features could be suppressed, leading to incorrect classification results. PointNet++ generally produces successful results since it considers neighbors while extracting features and the local regions expand hierarchically in each layer. Therefore, it can understand the characteristics of scenes and correctly classify doors. However, in some cases given Figure 4(b), PointNet++ may yield incorrect results due to missing parts of structural planes.

5.2.3 Closed Door Results

The quantitative results for closed doors are given in Table 6. As seen from the table, architectures are able to classify closed doors, except for a few examples. The misclassified examples are similar to those given for open and semi-opened results.

Conclusion and Future Works

In this study, we intended to discover the potential of point-based deep learning architectures for door detection problems because the existing studies suffer from the generalization ability. Their successes rely on the distance and angle between the door frame and the sensor. Therefore, we implemented five point-based deep learning architectures. Besides, we constructed OGURUB DOORS dataset because existing door datasets were inappropriate, especially deep learning architectures that require a large number of samples in the training phase. The test results indicate that all architectures can classify doors over 98% recall, precision, and F1-score values via the point cloud data captured when the robot is located in all possible positions and orientations toward a door frame. The reason for that, the point-based deep learning architectures attempt to learn general characteristics of the scenes regardless of the position and orientation of the robot.

Table 5
Quantative results for semi-opened door class

A n g l e (o)	Dist (m)	NOS	Methods														
			PointNet			PointNet++			DGCNN			PointCNN			Point2Sequence		
			Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
0-30	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	4	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	100.00	98.33	99.16	100.00	100.00	100.00	98.46	98.33	98.40	100.00	100.00	100.00	100.00	100.00	100.00
30-60	1.5-2	10	98.18	96.00	97.08	100.00	100.00	100.00	98.18	100.00	99.08	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	98.33	83.64	90.39	100.00	98.18	99.08	98.33	94.55	96.40	100.00	98.18	99.08	100.00	98.18	99.08
	2.5-3	14	98.67	97.14	97.90	100.00	97.14	98.55	98.67	98.57	98.62	98.67	98.57	98.62	98.67	97.14	97.90
	3-3.5	18	100.00	97.78	98.88	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.89	99.44
60-120	1.5-2	24	99.20	98.33	98.76	99.17	97.50	98.33	100.00	99.17	98.58	100.00	98.33	99.16	100.00	98.33	99.16
	2-2.5	23	100.00	100.00	100.00	99.17	100.00	99.58	99.17	100.00	98.58	98.40	100.00	99.19	98.33	99.13	98.73
	2.5-3	28	98.67	99.29	98.98	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	38	99.49	98.42	98.95	100.00	99.47	99.74	100.00	100.00	100.00	100.00	100.00	100.00	98.49	99.47	98.98
120-150	1.5-2	10	98.18	98.00	98.09	100.00	100.00	100.00	100.00	98.00	98.99	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	14	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	18	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
150-180	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	4	96.00	100.00	97.96	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	100.00	98.33	99.16	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 6
Quantative results for closed door class

A n g l e (o)	Dist (m)	NOS	Methods														
			PointNet			PointNet++			DGCNN			PointCNN			Point2Sequence		
			Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
0-30	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	4	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	95.60	91.67	93.59	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
30-60	1.5-2	10	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	14	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.67	100.00	99.33	100.00	100.00	100.00
	3-3.5	18	98.95	98.89	98.92	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
60-120	1.5-2	24	99.20	100.00	99.60	99.20	99.17	99.18	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	23	100.00	100.00	100.00	100.00	99.13	99.56	100.00	99.13	99.56	100.00	100.00	100.00	100.00	98.26	99.12
	2.5-3	28	99.31	99.29	99.30	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	38	99.49	100.00	99.74	99.49	100.00	99.74	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.47	99.74
120-150	1.5-2	10	98.18	98.00	98.09	100.00	100.00	100.00	98.18	100.00	99.08	100.00	100.00	100.00	100.00	100.00	100.00
	2-2.5	11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	14	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	18	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
150-180	1.5-2	2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	90.00	94.74	100.00	100.00	100.00
	2-2.5	4	100.00	90.00	94.74	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	2.5-3	8	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	3-3.5	12	91.92	98.33	95.02	100.00	100.00	100.00	96.92	100.00	98.44	100.00	100.00	100.00	100.00	100.00	100.00

For future works, we plan to consider more complex scenes that include occlusion and clutter around the proximity of door frames. Besides, we intend to apply augmentation during the training process. Although the OGUROB DOORS dataset consists of samples as possible as all robot positions and orientations depending on the range of the RGB-D camera, augmentation can promote the results. Lastly, the existence of sensor noises heavily affects the performance of the tasks in the field of robotics, especially in real-world applications. For that reason, we plan to extend our OGUROB DOORS dataset with the scenes captured from the real world. In this way, we can observe the influences of the noise on the implemented architectures' performance. However, it is important to note that we expect to obtain close results with and without sensor noises. The reason for that, these architectures generally aim to learn the characteristic of the scenes through points' position. Although the noise of a sensor can slightly alter these positions, the scene characteristics will remain the same.

Author Contributions

Burak Kaleci contributed to literature search, construction of dataset, preparation of figures and manuscript, Kaya Turgut, contributed to literature search, implementation of techniques, experimental work, and preparation of figures and manuscript.

Conflict of Interest

There is no conflict of interest.

References

- Andreopoulos, A. & Tsotsos, J. K. (2008). Active vision for door localization and door opening using playbot: A computer controlled wheelchair for people with mobility impairments. *In 2008 Canadian Conference on Computer and Robot Vision*, 3-10, Windsor, Canada. doi: <https://doi.org/10.1109/CRV.2008.23>
- Arduengo, M., Torras, C. & Sentis, L. (2019). Robust and Adaptive Door Operation with a Mobile Robot, arXiv, 1902.09051. doi: <https://doi.org/10.1007/s11370-021-00366-7>
- Bayram, K., Kolaylı, B., Solak, A., Tatar, B., Turgut, K. ve Kaleci, B. (2019). 3B Nokta Bulutu Verisi ile Bölge Büyütme Tabanlı Kapı Bulma Uygulaması. *Türkiye Robotbilim Konferansı*, 139-145, İstanbul, Türkiye. doi: <https://doi.org/10.31796/ogummf.723781>
- Beraldo G., Termine E., Menegatti E. (2019). Shared-Autonomy Navigation for Mobile Robots Driven by a Door Detection Module. *In: Alviano M., Greco G., Scarcello F. (eds) AI*IA 2019 – Advances in Artificial Intelligence. AI*IA 2019. Lecture Notes in Computer Science*, vol 11946. Springer, Cham. doi: https://doi.org/10.1007/978-3-030-35166-3_36
- Bersan, D., Martins, R., Campos M. & Nascimento, E. R. (2018). Semantic Map Augmentation for Robot Navigation: A Learning Approach Based on Visual and Depth Data, 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), João Pessoa, Brazil, 45-50. doi: <https://doi.org/10.1109/LARS/SBR/WRE.2018.00018>
- Borgesen, S. M. Z., Schöpfer, M., Ziegler, L. & Wachsmuth, S. (2014). Automated Door Detection with a 3D-Sensor. *Canadian Conference on Computer and Robot Vision*, 276-282, Montreal, QC, Canada. doi: <https://doi.org/10.1109/CRV.2014.44>
- Budroni, A. & Böhm, J. (2010). Automatic 3D modelling of indoor manhattan-world scenes from laser data. *Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5), 115-120. Retrieved from: <https://www.isprs.org/proceedings/XXXVIII/part5/papers/137.pdf>
- Burhanpurkar, M., Labbe, M., Guan, C., Michaud, F. & Kelly, J. (2017). Cheap or Robust? The practical realization of self-driving wheelchair technology. *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 1079-1086, London, UK. Retrieved from: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8009393>
- Chao, P., Kao, C. Y., Ruan, Y. S., Huang, C. H. & Lin, Y. L. (2019). Hardnet: A low memory traffic network, *ArXiv*, vol. abs/1909.00948. Retrieved from: https://openaccess.thecvf.com/content_ICCV_2019/papers/Chao_HardNet_A_Low_Memory_Traffic_Network_ICCV_2019_paper.pdf
- Chen, W., Qu, T., Zhou, Y., Weng, K., Wang, G. & Fu, G. (2014). Door recognition and deep learning algorithm for visual based robot navigation. *In 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 1793-1798, Bali, Indonesia. doi: <https://doi.org/10.1109/ROBIO.2014.7090595>
- Cui, Y., Li, Q., Yang, B., Xiao, W., Chen, C. & Dong, Z. (2019). Automatic 3-D reconstruction of indoor environment with mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8), 3117-3130, doi: <https://doi.org/10.1109/ISTARS.2019.2918937>

- Derry, M. & Argall, B. (2013). Automated doorway detection for assistive shared-control wheelchairs. *IEEE International Conference on Robotics and Automation*, 1254-1259, Karlsruhe, Germany. doi: <https://doi.org/10.1109/ICRA.2013.6630732>
- Díaz-Vilariño, L., Verbree, E., Zlatanova, S. & Diakité, A. (2017). Indoor modelling from SLAM-based laser scanner: Door detection to envelope reconstruction. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 42, 345-352. doi: <https://doi.org/10.5194/isprs-archives-XLII-2-W7-345-2017>
- Ehlers, S. F. G., Stuede, M., Nuelle, K. & Ortmaier, T. (2020). Map Management Approach for SLAM in Large-Scale Indoor and Outdoor Areas. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 9652-9658, Paris, France. doi: <https://doi.org/10.1109/ICRA40945.2020.9196997>
- ElKaissi, M., Elgamel, M., Bayoumi, M. & Zavidovique, B. (2006). SEDLRF: A new door detection system for topological maps. In *2006 international workshop on computer architecture for machine perception and sensing*, 75-80, Montreal, QC, Canada. doi: <https://doi.org/10.1109/CAMP.2007.4350358>
- Flikweert, P., Peters, R., Díaz-Vilariño, L., Voûte, R. & Staats, B. (2019). Automatic extraction of a navigation graph intended for IndoorGML from an indoor point cloud. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2/W5), 271-278. doi: <https://doi.org/10.5194/isprs-annalsIV-2-W5-271-2019>
- Gazebo (2021). Robot Simulation Open source robotics foundation (OSRF). Retrieved from: <http://gazebo.org/>.
- Gillham, M., Howells, G., Spurgeon, S., Kelly, S. & Pepper, M. (2013). Real-time doorway detection and alignment determination for improved trajectory generation in assistive mobile robotic wheelchairs. In *2013 fourth international conference on emerging security technologies*, 62-65, Cambridge, UK. doi: <https://doi.org/10.1109/EST.2013.18>
- Guo, Y., Wang, H., Hu, Q., Liu, H., and Bennamoun M. (2019), Deep Learning for 3D Point Clouds: A Survey, *arXiv*, Retrieved from: <http://arxiv.org/abs/1912.12033>.
- Jung, J., Stachniss, C., Ju, S. & Heo, J. (2018). Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM. *Advanced Engineering Informatics*, 38, 811-825. doi: <https://doi.org/10.1016/j.aei.2018.10.007>
- Hensler, J., Blaich, M. & Bittel, O. (2010). Real-Time Door Detection Based on AdaBoost Learning Algorithm. In: *Gottscheber A., Obdržálek D., Schmidt C. (eds). Research and Education in Robotics. Communications in Computer and Information Science*, 82, 61-73. doi: https://doi.org/10.1007/978-3-642-16370-8_6
- Kakillioglu, B., Ozcan, K. & Velipasalar, S. (2016). Doorway detection for autonomous indoor navigation of unmanned vehicles. *2016 IEEE International Conference on Image Processing (ICIP)*, 3837-3841, Phoenix, AZ, USA. doi: <https://doi.org/10.1109/ICIP.2016.7533078>
- Kaleci, B., Şenler, Ç. M., Dutagaci, H. & Parlaktuna, O. (2015). Rule-Based Door Detection Using Laser Range Data in Indoor Environments. *IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 510-517, Vietri sul Mare, Italy. doi: <https://doi.org/10.1109/ICTAI.2015.81>
- Kaleci, B. & Turgut, K. (2020). NOKTA BULUTU VERİSİ İLE KURAL TABANLI AÇIK KAPI BULMA YÖNTEMİ. *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 28(2), 164-173. doi: <https://doi.org/10.31796/ogummf.723781>
- Khoshelham, K., Vilariño, L. D., Peter, M., Kang, Z. & Acharya, D. (2017). The ISPRS benchmark on indoor modelling. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W7, 367-372. doi: <https://doi.org/10.5194/isprs-archives-XLII-2-W7-367-2017>.
- Kim, S., Cheong, H., Kim, D. H. & Park, S. K. (2011). Context-based object recognition for door detection. In *2011 15th International Conference on Advanced Robotics (ICAR)*, 155-160, Tallinn, Estonia. doi: <https://doi.org/10.1109/ICAR.2011.6088578>
- Koo, B., Jung, R., Yu, Y. (2021). Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks, *Advanced Engineering Informatics*, 47, ISSN 1474-0346. doi: <https://doi.org/10.1016/j.aei.2020.101200>.
- Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). PointCNN: Convolution on x-transformed points, in *Advances in Neural Information Processing Systems*, 31. Retrieved from: <https://proceedings.neurips.cc/paper/2018/file/f5f8590cd58a54e94377e6ae2eded4d9-Paper.pdf>
- Liu, X., Han, Z., Liu, Y., & Zwicker, M. (2019). Point2Sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network, *Thirty-Third AAAI Conference on Artificial Intelligence*. Retrieved from: <https://ojs.aaai.org/index.php/AAAI/article/download/4903/4776>
- Llopart, A., Ravn, O. & Andersen, N. A. (2017). Door and cabinet recognition using Convolutional Neural Nets and real-time method fusion for handle detection and grasping, *2017 3rd International Conference on*

- Control, Automation and Robotics (ICCAR)*, Nagoya, 144-149. doi: <https://doi.org/10.1109/ICCAR.2017.7942676>
- Meeussen, W., Wise, M., Glaser, S. & Chitta, S. (2010). Autonomous door opening and plugging in with a personal robot. *IEEE International Conference on Robotics and Automation*, 729-736, Anchorage, AK, USA. doi: <https://doi.org/10.1109/ROBOT.2010.5509556>
- Michailidis, G. T. & Pajarola, R. (2017). Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments. *The Visual Computer*, 33(10), 1347-1355. doi: <https://doi.org/10.1007/s00371-016-1230-3>
- Murillo, A. C., Košecká, J., Guerrero, J. J. & Sagüés, C. (2008). Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6), 512-521. doi: <https://doi.org/10.1016/j.robot.2008.03.003>
- Nagahama, K., Takeshita, K., Yaguchi, H., Yamazaki, K., Yamamoto, T. & Inaba, M. (2018). Estimating door shape and manipulation model for daily assistive robots based on the integration of visual and touch information. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7660-7666, Madrid, Spain. doi: <https://doi.org/10.1109/IROS.2018.8593391>
- Nieuwenhuisen, M., Stückler, J. & Behnke, S. (2010). Improving indoor navigation of autonomous robots by an explicit representation of doors. In *2010 IEEE International Conference on Robotics and Automation*, 4895-4901, Anchorage, AK, USA. doi: <https://doi.org/10.1109/ROBOT.2010.5509689>
- Nikooheemat, S., Peter, M., Elberink, S. O. & Vosselman, G. (2017). Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 355-362, Wuhan, China. doi: <https://doi.org/10.5194/isprs-annals-IV-2-W4-355-2017>
- OGUROB DOORS Dataset (2021). Web address: <https://ai-robotlab.ogu.edu.tr/Sayfa/Index/30>
- Othman, K. M. & Rad, A. B. (2020). A Doorway Detection and Direction (3Ds) System for Social Robots via a Monocular Camera, *Sensors*, 20(9), 2477. doi: <https://doi.org/10.3390/s20092477>
- Panzarella, T., Schwesinger, D. & Spletzer, J. (2016) CoPilot: Autonomous Doorway Detection and Traversal for Electric Powered Wheelchairs. In: *Wettergreen D., Barfoot T. (eds) Field and Service Robotics. Springer Tracts in Advanced Robotics*, 113, 233-248. doi: https://doi.org/10.1007/978-3-319-27702-8_16
- Pioneer P3-AT (2021). Retrieved from: <http://www.ist.tugraz.at/attach/Publish/Kmr06/pioneer-robot.pdf>
- Ramôa, J. G., Alexandre, L. A. & Mogo, S. (2020). Real-Time 3D Door Detection and Classification on a Low-Power Device, *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Ponta Delgada, Portugal, 2020, pp. 96-101. doi: <https://doi.org/10.1109/ICARSC49921.2020.9096155>
- Qi, C. R., Su, H., Mo, K. & Guibas, L. J. (2016). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *arXiv preprint arXiv:1612.00593*. Retrieved from: <https://arxiv.org/pdf/1612.00593.pdf>
- Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017), PointNet++: Deep hierarchical feature learning on point sets in a metric space, in *NeurIPS*, *arXiv preprint arXiv:1706.02413*. Retrieved from: <https://arxiv.org/pdf/1706.02413.pdf>
- Quijano, A. & Prieto, F. (2016). 3d semantic modeling of indoor environments based on point clouds and contextual relationships. *Ingeniería*, 21(3), 305-323. doi:<http://dx.doi.org/10.14483/udistrital.jour.reviving.2016.3.a04>
- Quintana, B., Prieto, S. A., Adán, A. & Bosché, F. (2018). Door detection in 3D coloured point clouds of indoor environments. *Automation in Construction*, 85, 146-166. doi: <https://doi.org/10.1016/j.autcon.2017.10.016>
- Redmon J., Divvala, S., Girshick, R. & Farhadi, A. (2016) You only look once: Unified, real-time object detection, *IEEE CVPR*. Retrieved from: <https://arxiv.org/pdf/1506.02640.pdf>
- Robot Operating System (ROS) (2021). Open source robotics foundation (OSRF). Web address: <http://ros.org/>.
- Rusu, R. B., Meeussen, W., Chitta, S. & Beetz, M. (2009). Laser-based perception for door and handle identification. *International Conference on Advanced Robotics*, 1-8, Munich, Germany. Retrieved from: <https://ieeexplore.ieee.org/iel5/5166725/5174665/05174717.pdf>
- Rusu, R. B. & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation*, 1-4, Shanghai, China. doi: <https://doi.org/10.1109/ICRA.2011.5980567>
- Sekkal, R., Pasteau, F., Babel, M., Brun, B. & Leplumey, I. (2013). Simple monocular door detection and tracking. In *2013 IEEE International Conference on Image Processing*, 3929-3933, Melbourne, VIC, Australia. doi: <https://doi.org/10.1109/ICIP.2013.6738809>

- Staats, B. R., Diakité, A. A., Voûte, R. L. & Zlatanova, S. (2019). Detection of doors in a voxel model, derived from a point cloud and its scanner trajectory, to improve the segmentation of the walkable space. *International Journal of Urban Sciences*, 23(3), 369-390. doi: <https://doi.org/10.1080/12265934.2018.1553685>
- Souto, L. A. V., Castro, A., Gonçalves, L. M. G. & Nascimento, T. P. (2017). Stairs and Doors Recognition as Natural Landmarks Based on Clouds of 3D Edge-Points from RGB-D Sensors for Mobile Robot Localization. *Sensors*, 17(8). doi: <https://doi.org/10.3390/s17081824>
- Su, H., Maji, S., Kalogerakis, E. & Learned-Miller, E. (2015). Multi-view Convolutional Neural Networks for 3D Shape Recognition, *Proceedings of ICCV*. Retrieved from: [https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Su Multi-View Convolutional Neural ICCV 2015_paper.pdf](https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Su_Multi-View_Convolutional_Neural_ICCV_2015_paper.pdf)
- Tensorflow (2021), Retrieved from: <https://www.tensorflow.org/>.
- Wang, R., Xie, L. & Chen, D. (2017). Modeling indoor spaces using decomposition and reconstruction of structural elements. *Photogrammetric Engineering & Remote Sensing*, 83(12), 827-841. doi: <https://doi.org/10.14358/PERS.83.12.827>
- 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 (TOG)*. doi: <https://doi.org/10.1145/3326362>
- Wu,H., Zhang, J., Huang, K., Liang, K. & Yizhou,Y. (2019). Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation, *arXiv preprint arXiv:1903.11816*. Retrieved from: <https://arxiv.org/pdf/1903.11816.pdf>
- Yang, X. & Tian, Y. (2010). Robust door detection in unfamiliar environments by combining edge and corner features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 57-64, San Francisco, CA, USA. doi: <https://doi.org/10.1109/CVPRW.2010.5543830>
- Ye, C. & Qian, X. (2018). 3-D Object Recognition of a Robotic Navigation Aid for the Visually Impaired. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2), 441-450, doi: 10.1109/TNSRE.2017.2748419
- Zheng, Y., Peter, M., Zhong, R., Oude Elberink, S. & Zhou, Q. (2018). Space subdivision in indoor mobile laser scanning point clouds based on scanline analysis. *Sensors*, 18(6), 1838. doi: <https://doi.org/10.3390/s18061838>