TRAINING DEEP NETWORKS WITH BIM MODELS FOR INDOOR POINT CLOUD CLASSIFICATION

MUHAMMAD NUZUL MAHFIRUDDIN SYAH DEMA
August 2023

SUPERVISORS:
Dr. Ville Valtteri Lehtola (First Supervisor)
Dr. Ir. Sander Oude Elberink (Second Supervisor)

TRAINING DEEP NETWORKS WITH BIM MODELS FOR INDOOR POINT CLOUD CLASSIFICATION

MUHAMMAD NUZUL MAHFIRUDDIN SYAH DEMA
Enschede, The Netherlands, August 2023

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. Ville Valtteri Lehtola (First Supervisor)
Dr. Ir. Sander Oude Elberink (Second Supervisor)

THESIS ASSESSMENT BOARD:
Prof.Dr.Ir. M.G. Vosselman
Dr. Maarten Bassier

ABSTRACT

Deep learning methods has been used in the point cloud classification applications. Particularly, it is used to provide as-built conditions of the buildings for construction progress monitoring. However, there is limited availability of labeled indoor point cloud datasets publicly available to train the deep learning network. Consequently, it can brings incorrect information and lead to cost overrun. Nevertheless, Building Information Models or BIM are available as it is used as the design model for the buildings. Therefore, this research leverages the BIM models to generate synthetic point clouds that can overcome this problem.

The main results of this research is that this approach can successfully generate the synthetic point clouds to be used as additional dataset for point clouds classification. The networks trained on the synthetic point clouds has 14.22% mean – Intersection over Union (m-IoU) differences compared to the benchmark point clouds dataset, the S3DIS. Additionally, by augmenting the synthetic point clouds and the S3DIS dataset, it has 17.69% m-IoU differences compared to only using the S3DIS dataset. However, this approach failed completely classify stair and window elements due to class-imbalance and inter-class similarity problems.

Keywords: Building information model, deep Learning, point cloud classification, construction progress monitoring

ACKNOWLEDGEMENTS

I would like to convey my profound gratitude to all those who have played a pivotal role in the realization of this research. Most notably, I profoundly grateful for the trust the LPDP RI in granting me this prestigious scholarship. I would also like to express my sincere thanks to my supervisors, namely Prof. Francesco, Dr. JR, and Prof. Vosselman. Their invaluable guidance and expertise have been influential in immersing me into the realm of 3D deep learning and ensuring the completion of this research.

My appreciation also extends deeply to my treasured family and friends for their support throughout my academic journey. To my father and mother, your boundless affection is truly cherished. To Ali, Archita, Arivia, Aulia, Hafidz, Wildan, Mas Yan, Mas Aufar, Mas Andhika, and Mas Deva, I am immensely grateful to have you as companions during my time in the Netherlands.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Point cloud classification is challenging in processing Terrestrial Laser Scanning (TLS) data. Despite the use of deep learning methods, there is limited availability of labeled indoor point cloud datasets publicly available to train the network (Gao et al., 2020; Marcus, 2018; Sun et al., 2017). For example, even though the Stanford 3D Indoor Scene Dataset or S3DIS (Armeni et al., 2016) provides a large-scale indoor point cloud, it only covers office architectural layout types. Utilizing the S3DIS dataset for non-office buildings (e.g., universities, hospitals, malls, or schools) can lead to network performance degradation (Gao et al., 2020). As point cloud classification is one of the important procedures in construction progress monitoring, it can provide incorrect as-built conditions of the buildings and lead to cost overrun (Son & Kim, 2010). The reason is that the deep learning methods require labeled datasets corresponding to the target application (Gao et al., 2020). Therefore, this research leverages BIM models to generate labeled point clouds that can overcome this problem.

The uses of BIM models in construction progress monitoring are introduced in Section 1.1. Then, Section 1.2 and Section 1.3 present the deep learning methods for point cloud classification and its challenges, respectively. Last, Section 1.4, Section 1.5, and Section 1.7 define the hypothesis and objectives of this research. Some references in this research defined point cloud classification as point cloud semantic segmentation, as it is often used in computer vision (Xie et al., 2019).

1.1. Construction Progress Monitoring

Delays or deviations from the planned schedule often occur in building construction (Baldwin et al., 1971). It can be due to multiple factors, including bad weather, equipment failure, material shortage, jurisdictional disputes, etc. Consequently, it can increase the risk of exceeding the allocated labor, equipment, and materials expenses. Therefore, construction progress monitoring is required.

Construction progress monitoring ensures the building constructions follow the planned schedule (Arditi & Gunaydin, 1997). It involves regularly obtaining information on the as-built condition of the building's indoor scenes. In particular, the information is expressed in terms of the ID, category attribute classes, and location of the indoor building elements (e.g., beam, ceiling, column, door, floor, railing, stair, wall, and window) that has been built. The information is compiled into documentation and shared with the other stakeholders. Then, the information is analyzed and compared against the planned schedule and the design model to determine the construction progress. After that, mitigation measures are taken if delays are identified.

BIM, or Building Information Models, has been widely used as the design model in the construction domain (Kim et al., 2013; Son & Kim, 2010; Turkan et al., 2012; Xiong et al., 2013). BIM models refer to a three-dimensional digital representation of buildings planned to be constructed. It contains information about the building's elements (e.g., ID, category attribute classes, and location).

1.2. Scan-to-BIM Methods

The traditional methods of construction progress monitoring are done through manual data collection with documentation in written descriptions, photographs, videos, or sketches (Arditi & Gunaydin, 1997). While it gives a basic identification of the construction progress, it demands significant time, increasing the inefficiency. It requires 30 to 49 % of site managers' time (Son & Kim, 2010). Then, it also lacks comprehensive documentation, which can cause improper decision-making from the stakeholders. For example, if 60% of project completion is judged as 50%, stakeholders will assign more resources than needed, leading to construction cost overruns (Son & Kim, 2010). As a result, the Scan-to-BIM method has been utilized to facilitate construction progress monitoring with faster data collection and comprehensive documentation (Hajian & Becerik-Gerber, 2010).

The Scan-to-BIM methods can provide a 3D model of the as-built condition of the indoor building elements (Kim et al., 2013; Son & Kim, 2010; Turkan et al., 2012; Xiong et al., 2013). The 3D model enhances understanding regarding the construction progress, allowing efficient stakeholder communication and appropriate decision-making. In addition, the 3D model can be overlayed with the design model for a direct and fast comparison. The workflow of the methods is data collection, data annotation, and 3D model reconstruction per building element.

Terrestrial Laser Scanning (TLS) technology can be used for the Scan-to-BIM process as data collection (Hajian & Becerik-Gerber, 2010). It has faster data collection than the traditional methods of construction progress monitoring, with 0.347 hours per $100\mathrm{m}^2$ compared to 0.875 hours per $100\mathrm{m}^2$ (Griffiths & Boehm, 2019a). It measures the laser pulse travel distances sent by the sensor system, reflected by the visible building elements' surfaces, and received back at the sensor system (Vosselman & Maas, 2010). Then, millions of 3D points, known as point clouds, are generated. The point clouds represent the visible building elements' surfaces with three-dimensional spatial coordinates and color features. The measurements are performed in several locations inside the buildings to acquire necessary coverage of the buildings. After that, the point clouds acquired from each measurement are merged in a single coordinate system.

Point clouds, acquired from Terrestrial Laser Scanning (TLS), lack semantic information regarding the building element it represents (Vosselman & Maas, 2010). Therefore, classification is necessary to turn them into meaningful scenes for 3D model reconstruction (Garcia-Garcia et al., 2017). In particular, point cloud classification labels each point cloud based on what building element it represents. Deep learning methods have been applied for point cloud classification, especially for indoor scenes (Garcia-Garcia et al., 2017; Griffiths & Boehm, 2019a; Guo et al., 2019). Compared to the traditional methods (e.g., Support Vector Machine or Random Forest), the deep learning methods has the key advantage of not needing a huge fine-tuning to extract meaningful features, generating higher results (Garcia-Garcia et al., 2017). In particular, the Kernel Point Fully Convolutional Network, or KP-FCNN (Thomas et al., 2019) achieved a good result in the indoor point cloud dataset of Stanford Large-Scale 3D Indoor Spaces or S3DIS (Armeni et al., 2016). The KP-FCNN has a mean Intersection-over-Union or m-IoU of 67.1% for the network's overall performance.

As mentioned at the beginning of the research, the key requirements for using deep learning methods are to provide labeled datasets (Gao et al., 2020; Marcus, 2018; Sun et al., 2017). Specifically, the datasets must be accurate regarding geometric and semantic information. Then, datasets also must be relevant to the target application regarding the shapes, spatial distributions, orientations, range, and noise. The deep learning methods use the datasets to train the network, where the network learns discriminative features of each class in the dataset. Then, the learned features are used to predict the unlabeled dataset of the target application. If the key requirement is unmet, the network can encounter inconsistent features in the predicted datasets, resulting in biased parameter estimation and poor performance.

1.3. Problem Statement

As mentioned at the beginning of the research, there is a limited availability of labeled indoor point cloud datasets publicly available. Several indoor point cloud datasets, such as NYU Depth Dataset V2 or NYDV2 (Couprie et al., 2013), SceneNN (Hua et al., 2016), Matterport3D (Chang et al., 2017), and ScanNet (Dai et al., 2017) was captured using an RGB-D camera with lower accuracy than laser scanning technology. Then, despite covering indoor scenes, Paris-Lille-3D (Roynard et al., 2018) and HPS (Guzov et al., 2021) do not have any annotation regarding category attributes, which requires manual annotation. Last, the Stanford 3D Indoor Scene Dataset or S3DIS (Armeni et al., 2016) only consists of indoor office scenes. Therefore, there is a massive need for a more diverse point cloud dataset, particularly one that covers indoor scenes with other architectural layout types.

Gao et al. (2020) have a performance degradation when the networks that trained on rural scenes, from SemanticKITTI (Zhou et al., 2020) and SemanticPOSS (Pan et al., 2020), are used to predict on urban scenes, and contrariwise. The reason is that the rural scenes dataset have distinct shapes from the urban scenes dataset (Gao et al., 2020; Marcus, 2018; Sun et al., 2017). One method to address this problem is to obtain additional labeled datasets involving manual data collection and classification (Garcia-Garcia et al., 2017). However, similar to the traditional methods of construction progress monitoring mentioned in Section 1.2, the process is laborious and time-consuming, which can increase the construction cost. Additionally, manual classification is a subjective process that can make the results inconsistent from multiple operators and decrease the network performance.

1.4. The First Hypothesis Solution

Given the problems in Section 1.3, this research proposed a method to leverage existing BIM models. The geometric and semantic information from the BIM models is converted into synthetic point clouds (Ma et al., 2020; Noichl et al., 2021; Zhai et al., 2022). Then, the synthetic point clouds have a label corresponding to the building elements (e.g., beam, ceiling, column, door, floor, railing, stair, wall, and window). This approach does not utilize manual data collection and classification, which makes it an inexpensive and non-subjective process. Furthermore, since the buildings are constructed based on the BIM models, the synthetic point clouds will share similar architectural layouts with the constructed building. It makes the synthetic point clouds relevant to the point cloud classification for this building. Therefore, the first hypothesis of this research is that utilizing the BIM models has the potential to help point cloud classification at indoor scenes. In particular, it generates labeled synthetic point clouds to train the deep learning network.

1.5. The Second Hypothesis Solution

Multiple research recommends generating synthetic datasets must have real point cloud characteristics to increase the classification performance (Ma et al., 2020; Noichl et al., 2021; Zhai et al., 2022). These characteristics include the local point cloud distribution, occlusion effect, sensor system noise, and glass reflectivity, which will be described more in Section 2.1.4, Section 2.1.5, and Section 2.1.6. Thus, the second hypothesis of this research is that including real point cloud characteristics in the synthetic point clouds can increase the point cloud classification performance.

1.6. The Third Hypothesis Solution

Compared to synthetic point clouds, the S3DIS dataset can possess certain unknown real point cloud characteristics beyond the local point cloud distribution, occlusion effect, sensor system noise, and glass reflectivity. To address this difference and enhance classification performance, a domain adaptation method can include these characteristics in the synthetic point clouds. Domain adaptation is a method to adapt networks when the training and test datasets are derived from different source domains (Torralba & Efros, 2011). This method can be done by training the network with the augmentation of the S3DIS datasets and the synthetic point clouds. In this method, the network is trained with the augmentation of the S3DIS datasets and the synthetic point clouds.

Ma et al. (2020) have previously generated synthetic point clouds in indoor scenes to tackle the limited datasets availability problem. However, they had lower classification results of - 11.81% mean Intersection over Union (m-IoU) when the networks trained on the synthetic point clouds compared with the real point clouds. Nevertheless, they achieved a 7.1% m-IoU increase in indoor classification tasks by combining the real point clouds with the synthetic point clouds derived from the BIM model. Similarly, Yue et al. (2018) and Wang et al. (2019) also demonstrated that augmenting synthetic point clouds with real point clouds boost the classification results by 9.0% m-IoU compared to solely using the real point clouds to train the network. Therefore, the third hypothesis of this research is that utilizing the combination of the synthetic point clouds and the S3DIS dataset to train the network can include unknown real point cloud characteristics into the synthetic point clouds and can improve the classification performance.

1.7. Research Objectives and Questions

The research's main objective is to confirm the BIM models' effectiveness for point cloud classification applications to overcome the problem of limited availability of labeled indoor point cloud datasets. Specifically, the BIM models are converted into labeled synthetic point clouds to train the deep learning network. In this regard, a framework is designed for construction progress monitoring by implementing the BIM models, a virtual laser scanner tool, and a KP-FCNN in the synthetic point cloud generation and classification. The main objective of the research is divided into sub-objectives with the following research questions.

1. **Mitigate the problem of limited availability of labeled indoor point cloud datasets.**
    - How will the results changed when the same synthetic point clouds are used in different construction stages of the buildings?
    - How can the augmentation of the synthetic point clouds and the S3DIS dataset can improve the classification results?

2. **Generate the synthetic point cloud as realistic as possible to improve the classification results.**
    - What is the right way to simulate the local point cloud distribution and occlusion effect to help the point cloud classification?
    - How can including sensor system noise in the synthetic point clouds help the point cloud classification?
    - How can the synthetic point clouds that consider the glass as transparent object help the point cloud classification?

3. **Configure the deep learning network to classify point clouds in indoor construction scenes.**
    - How robust is the KP-FCNN deep learning network in point cloud classification in indoor scenes?

2. LITERATURE REVIEW

This Chapter presents the constraints of various methods in addressing the limited availability of labeled indoor point cloud datasets mentioned in Section 1.3. The objective is to establish a motivation for using the approach described in Section 1.4. This chapter also describes the limitations of some deep learning networks to motivate the adoption of the Kernel Point-Fully Convolutional Neural Network or KP-FCNN (Thomas et al., 2019).

2.1. Battling The Limited Availability of Labeled Indoor Point Clouds Datasets

Multiple research has been conducted to overcome the problems due to the limited availability of labeled indoor point cloud datasets publicly available, described at the beginning of the paper and in Section 1.3,

2.1.1. Methods Not Requiring Huge Labeled Point Clouds

Various research developed deep learning methods for point cloud classification that do not need huge labeled point clouds. It includes incorporating weak supervision methods and self-supervised methods into the network. Xu & Lee (2020) performed weak supervision methods, where the network is trained using partially labeled point cloud datasets. It estimates the learning gradient and utilizes additional spatial and color smoothness constraints. However, the methods involve multiple iterations, making the process computationally more expensive than the supervised methods.

Motivated by the prediction pretext task for image classification, Sauder & Sievers (2019) propose self-supervised methods. The network learns the spatial distribution of point clouds where some parts are randomly rearranged. The limitation of this approach is the lack of evaluation regarding how to fine-tune the approach to a specific domain.

2.1.2. The Data Augmentation Methods

Multiple research also proposed data augmentation methods to generate additional labeled point clouds and increases the point clouds' diversity. Chen et al. (2020) introduce PointMixup. It interpolates and finds the linear shortest path between two point clouds to generate a new scene. Compared to the supervised methods that use only the available point clouds, PointMixup in a semi-supervised setting increases the network accuracy from 73.5% to 82.0%.

Emunds et al. (2021) present IFCNet as data augmentation. It comprises geometric and semantic information of single-entity IFC classes to generate additional point clouds. However, these methods limit their focus to object-level point clouds without scene-level point clouds.

2.1.3. Generating Synthetic Point Clouds

Numerous research converts the virtual environment into synthetic point clouds to extend the limited point cloud datasets availability. These methods can perform point cloud classification at the scene-level, overcoming the previous methods. Yue et al. (2018) and Wu et al. (2018) produced synthetic point clouds in urban scenes derived from video games' three-dimensional models, named GTA-V. Despite that, manual efforts were involved for semantic labeling since video game models have insufficient semantic information.

Ma et al. (2020) and Zhai et al. (2022) generated synthetic point clouds in indoor scenes. Using Autodesk Revit and Trimble SketchUp, they constructed the BIM models from the S3DIS dataset (Armeni et al., 2016). Then, they randomly put the point clouds on the BIM model surfaces using Feature Manipulation Engine (FME) Workbench.

Both approaches stated that the synthetic point clouds do not have some characteristics of the real point clouds or those acquired from Terrestrial Laser Scanners (TLS). It includes local point cloud distribution, occlusion effect, sensor system noise, and glass reflectivity. Consequently, the performance of the network trained with the synthetic point clouds is not comparable with one trained with the S3DIS dataset.

2.1.4. Generating Synthetic Point Clouds that Consider Local Point Cloud Distribution and Occlusion Effect

During the scanning process, the occlusion effect can happen when an area is occluded from the sensor system's view by an object between them (Vosselman & Maas, 2010). It can result in incomplete data in the acquired point cloud. Then, local point cloud distribution from the point clouds acquired by the sensor is uneven. It happens because of the sensor system's perspective effects and radial motions. It is also due to the varying distance between the sensor system and the object, as the point cloud density increases when the object is closer to the sensor system.

Dosovitskiy et al. (2017), Griffiths & Boehm (2019b), and F. Wang et al. (2019) modeled the occlusion effect and uneven local point cloud distribution in the synthetic point clouds at urban scenes. Using the autonomous driving simulator CARLA (Dosovitskiy et al., 2017) and the Blensor (Gschwandtner et al., 2011), they simulated a laser emitted from a Terrestrial Laser Scanner by placing multiple virtual sensor systems in the virtual environments. However, they do not examine the importance of the occlusion effect and the uneven local point cloud distribution.

2.1.5. Generating Synthetic Point Clouds that Consider Sensor System Noise

Sensor system measurements always include noise, which refers to unwanted variations in the output (Vosselman & Maas, 2010). Based on the sensor system quality, it can be caused by the mistake in measuring the distance between the sensor system and the objects. The sensor system noise can interfere with the position of the point cloud and represent a flat surface as a rough surface.

Wu et al. (2018) transferred the noise distribution of SemanticKITTI (Behley et al., 2019) into synthetic point clouds using a domain adaptation method of geodesic correlation alignment. The motivation behind the method is to address the domain shift problem, where there is a discrepancy in the sensor system noise level between the synthetic point clouds and SemanticKITTI point clouds. The network accuracy doubled from 29.0% to 57.4% compared to the network without geodesic correlation alignment. Hence, there is a possibility that point cloud classification results can be improved by including sensor system noise.

2.1.6. Generating Synthetic Point Clouds that Consider Glass Reflectivity

Window elements in the buildings consist of transparent glass and a non-transparent frame (Vosselman & Maas, 2010). When a laser pulse from a laser scanner encounters the transparent glass, most of the laser pulse can pass through without returning the laser pulse. The reason is that the transparent object's refractive index closely matches the surrounding medium. As a result, a point cloud for a glass object is not generated. Nevertheless, as illustrated in Figure 2.1, there are also cases where some fraction of the laser energy is absorbed by the transparent object, generating a point cloud. Unfortunately, there has not been any research that verifies whether the glass object should be considered transparent or non-transparent in synthetic point clouds to have good classification results.

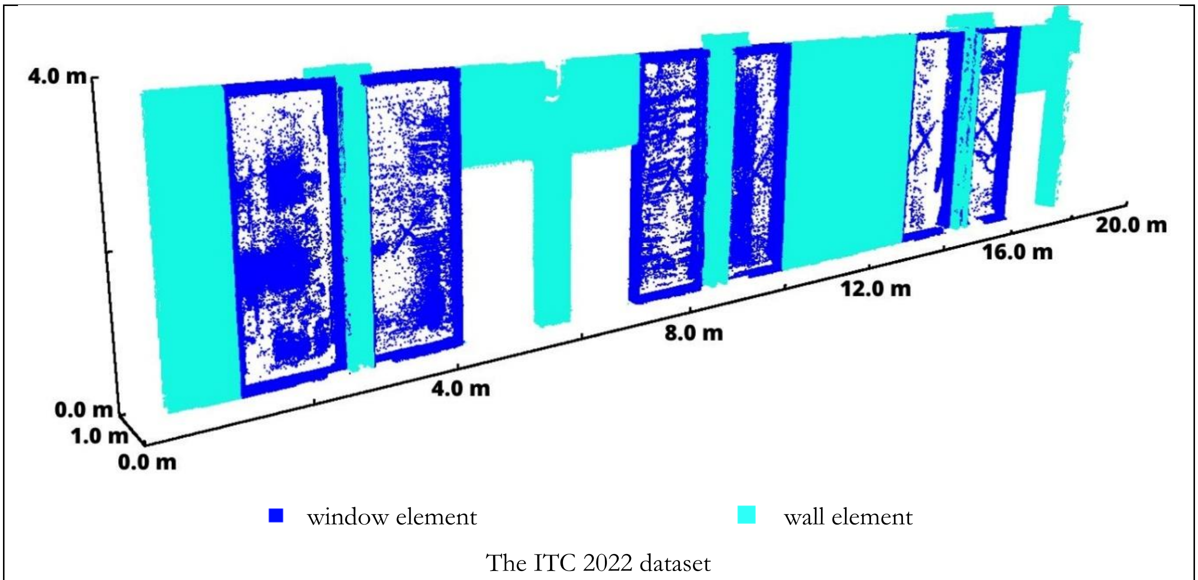


Figure 2.1 Window Element in the ITC 2022 dataset (Source: Author)

2.2. Deep Learning Networks on Point Clouds

Other than the challenge described at the beginning of the research and in Section 1.3, point cloud classification also can not directly use deep learning 2D convolution operation (Garcia-Garcia et al., 2017; Griffiths & Boehm, 2019a; Xie et al., 2019). The reason is due to the nature of point clouds. Unlike raster, the point cloud is unordered (invariant to permutations), unstructured (varying distances to neighboring point clouds), and irregular (unevenly sampled).

2.2.1. Indirect Methods

Multi-view-based methods (Su et al., 2015) overcome these challenges by converting point clouds into multiple 2D images using projection from several positions. Then, it applies a convolution operation with 2D kernels. But it suffers from occlusion sensitivity which can bring information loss. Then, voxel-based methods (C. Wang et al., 2019) convert point clouds into a fixed-size 3D grid structure and apply a convolution operation with 3D kernels. However, the methods convert the space not occupied by point clouds into voxels, leading to huge computation costs.

2.2.2. Direct Methods

Multiple networks from the point-based methods can directly take point clouds as input. PointNet (Qi et al., 2016) is the first method that applies deep learning convolution operation on point clouds. It is built on a multilayer perceptron (MLP) and a max-pooling function. The limitation of this method is that it does not capture local features. Then, the network can only handle 2048 point clouds at a time, making the method unable to handle large-scale point clouds.

PointNet++ (Qi et al., 2017) applies a PointNet in a hierarchical structure with a shared multilayer perceptron (MLP) function for local region computation. Then, the Pointwise Convolution method (Hua et al., 2017) creates multiple local neighborhoods where each input point cloud is used as the centroid. Then, the neighboring point clouds are sampled based on the adjusted radius value from the centroid. After that, the convolution operation is done independently for each local region. Unlike PointNet, PointNet++ and Pointwise Convolution learns individual point cloud features, which make them insensitive to varying density of point clouds. However, these methods do not explore local correlation, making them incapable of capturing small detailed features.

2.2.3. Direct Methods that Explore Local Correlation

Improved from the PointNet, some networks can capture the point cloud correlation in the local neighborhood. Graph-based methods (Klokov & Lempitsky, 2017) represents the input point clouds with a kd-tree graph structure and treats each point cloud as a node. The kd-network captures hierarchical relations between point clouds. However, the point clouds at the same depth level do not capture overlapping receptive fields.

PointCNN (Li et al., 2018) randomly samples the input point clouds and selects the neighboring ones based on the k - Nearest Neighbors (k-NN). Then, it utilized an X-transformation on the neighboring point clouds before applying a shared multilayer perceptron (MLP) function. The X-transformation explores the local correlations between point clouds in a local neighborhood to improve discriminative capability.

RandLa-Net (Hu et al., 2019) randomly samples the point clouds and does not use graph construction or kernelization, which requires less computation cost. Then, it captures and captures the local features of the point clouds using attentive pooling. However, it does not work effectively using small-scale point clouds since it does not learn point clouds independently.

2.3. Kernel Point Fully Convolutional Network (KP–FCNN) Deep Learning Network

As mentioned in Section 1.2, the Kernel Point Fully Convolutional Network or KP-FCNN proposed by Thomas et al. (2019) achieved a good point cloud classification performance in indoor scenes. Furthermore, the network does not have the limitations of previous networks mentioned in Section 2.2. Therefore, this research utilizes this network. KP-FCNN is a point-based deep learning method that directly learns the point clouds without converting them into intermediate data format, making it computationally efficient.

KP-FCNN has multiple layers with different receptive fields to learn input point clouds that vary in density and scale. The smaller features can only be captured at a lower receptive field, while the larger ones need higher receptive fields. KP-FCNN utilizes a pooling layer to increase the receptive field at every layer. It progressively downsamples the amount of the input point clouds using kernel point convolution of KPConv operation and grid subsampling operation. Table 1 explains multiple network parameters for the KP-FCNN need to be set during the network design.

Then, KP-FCNN has two different sampling strategies. The random picking strategy arbitrarily samples the input point clouds and samples the same number for each class. Contrarily, the regular picking strategy has a spatially consistent sampling method.

Table 1. Kernel Point Fully Convolutional Network Parameters

Network Parameters	Descriptions	Operation
The input features number Din	The number of input features (e.g., x, y, z, R, G, and B) with one additional default value	Kernel Point Convolution (KPConv)
The input sphere radius r	A radius of the local neighborhood N_x that controls which kernel point x_i computed for the convolution process of input point cloud x	
The convolution radius r	A radius of the kernel domain ball B_r^3 which controls the position of the kernel points $\bar{x}_k$	
The kernel influence distance σ	A correlation function at the given kernel point neighbor y_i and the kernel point $\bar{x}_k$ in the kernel domain ball B_r^3	
The kernel points number K	The number of kernel points $\bar{x}_k$ for the kernel domain ball B_r^3.	
The size of the first voxel grids dl_0	The cell size of the first grid subsampling	Grid Subsampling
Sampling Strategy	Method to reduce the number of point clouds	

2.3.1. Kernel Point Convolution (KPConv) Operation

Kernel point convolution (KPConv) operations are performed on each input point cloud at each network layer (Thomas et al., 2019). The convolution operation has kernel points arranged consistently spherically with a specified sphere radius and kernel point number. It used the specified distance from the neighboring point clouds to give each kernel a unique weight. From Figure 2.2, the constant distance guarantees a consistent receptive field that helps the network learn more meaningful representations compared to the k - Nearest Neighbors (k-NN) method used by PointCNN (Li et al., 2018). The results of this operation will update the features of the input point clouds in the next layer. Equation (1), Equation (2), and Equation (3) explain the KPConv operation on the point clouds.

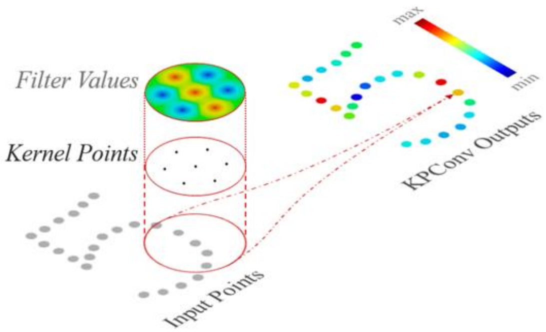


Figure 2.2. KPConv Network Kernels Fixed-size Radius (Source: Thomas et al. (2019))

$$(F * g)(x) = \sum_{x_i \in N_x} g(x_i - x) f_i \qquad \text{Equation (1)}$$

The convolution operation at the given input point cloud x requires the feature of the input point clouds F. A set of kernel points x_i is generated inside the local neighborhood N_x with the given input point cloud x at the center. The shape of the local neighborhood N_x is defined by the input sphere radius r. These kernel points x_i have the same position and feature properties as the corresponding input point cloud x. The convolution operation computes a weighted sum of the features f_i of these kernel points x_i using Equation (1). The weights are determined by the kernel function g, where it would be higher when the distance from the kernel points x_i to the given input point cloud x is closer.

$$g(y_i) = \sum_{k<K} h(y_i, \bar{x}_k) W_k \qquad \text{Equation (2)}$$

The kernel point neighbor y_i is defined as the kernel point x_i with a position relative to the given input point cloud x. A set of K-number kernel points $\bar{x}_k$ are generated inside the kernel domain ball B_r^3 with the given input point cloud x at the center. The shape of the kernel domain ball B_r^3 is defined by the convolution radius r. Each kernel point $\bar{x}_k$ is accompanied by a weight matrix W_k learned during the network training. The kernel function g at the given kernel point neighbor y_i is a weighted sum of the learned weight matrix W_k and is calculated using Equation (2). The weights are determined by the correlation function h, which takes the position of the kernel point neighbors y_i and the kernel points $\bar{x}_k$.

$$h(y_i, \bar{x}_k) = \max\left(0, 1 - \frac{\|y_i - \bar{x}_k\|}{\sigma}\right) \qquad \text{Equation (3)}$$

Equation (3) shows that the value of the correlation function h is ranged from zero to one. The value will be close to one when the distance from the kernel point neighbor y_i to the kernel point $\bar{x}_k$ is near. On the contrary, the value will be zero when the distance is the same as the kernel influence distance σ.

2.3.2. Grid Subsampling Operation

Grid subsampling operation is performed at each layer of the network, with the adjustable cell size of the first voxel grids dl_0 (Thomas et al., 2019). The operation divides the input point clouds into voxel grids with a consistent size and location. Then, a support point cloud is generated inside each voxel grid at the input point cloud closest to the voxel grid's centroid. It will not be generated without an input point cloud inside a voxel grid. After that, strided convolution is performed to the voxel grids, doubling the cell size of the voxel grids, reducing the number of input point clouds by a factor of four, and increasing the receptive field. A smaller voxel grid corresponds to a smaller receptive field allowing the network to capture local features.

3. METHODOLOGY

Based on the hypothesis described in Section 1.4, Section 1.5, and Section 1.6, this research performs the methodology with the workflow illustrated in Figure 3.1. First, this research defines the building element classes in classification based on the BIM models and the construction progress monitoring domain. Second, the BIM model preparation is defined in Section 3.2. Third, this research converts the BIM models into five sets of synthetic point clouds with different configurations in Section 3.3. Fourth, data preparation and normalization are done to the point clouds in Section 3.4 and Section 3.5. Fifth, the network training and testing process is described in Section 3.6. Last, the metrics used for the network performance evaluation are reviewed in Section 3.7.

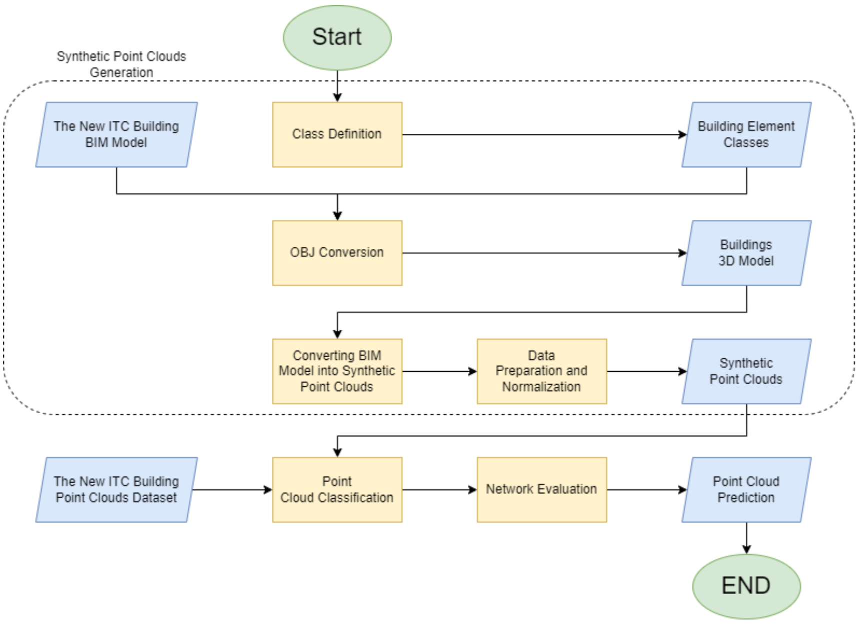


Figure 3.1 The Research Workflow

3.1. Class Definition

The initial step in the methodology is to determine the classes of building elements that should be considered for classification. There are two conditions for the class definition. The first condition is that it belongs to the main structural elements commonly built from the beginning of the construction process (Baldwin et al., 1971). The second condition corresponds with the BIM models used to design the building. The classes selected for this research are beam, ceiling, column, door, floor, railing, stair, wall, and window.

Section 1.1 describes the BIM models in the Industry Foundation Class (IFC) standards to define the element's attributes. Table 2 shows all IFC-related classes of the BIM models. Not all IFC classes are utilized for this research, as some need to be classified, and some are unused entirely. The IFC classes of the BIM models selected for this research are displayed in Table 3. Lastly, the IFC class is exported separately for further annotation processing. BIMVision (*BIMvision - Freeware IFC Model Viewer*, 2023) performs the IFC class export and classification.

Table 2. List of IFC Class Available in the BIM models

No	IFC Class	No	IFC Class	No	IFC Class
1	IFC Annotation	11	IFC FlowController	21	IFC Space
2	IFC Beam	12	IFC FlowFitting	22	IFC Stair
3	IFC Building Element Proxy	13	IFC FlowTerminal	23	IFC Standard Wall
4	IFC Column	14	IFC Member	24	IFC Stairway
5	IFC Covering	15	IFC Plate	25	IFC Wall
6	IFC Curtain Wall	16	IFC Railing	26	IFC Window
7	IFC Door	17	IFC Ramp		
8	IFC Element Assembly	18	IFC Ramp Flight		
9	IFC Element Furniture	19	IFC Roof		
10	IFC Footing	20	IFC Slab		

Table 3. List of IFC Class used in this Research

No	IFC class
1	IFC Beam
2	IFC Column
3	IFC Wall
4	IFC Door
5	IFC Window
6	IFC Slab
7	IFC Stair
8	IFC Railing
9	IFC Roof

Table 4 details the IFC classes unused in this research. These IFC classes are unused because they are not present in the actual building, not present in the indoor scenes, and do not belong to the structural building elements (e.g., beam, ceiling, column, door, floor, railing, stair, wall, and window). For example, Figure 3.2 shows that IFC Space and IFC Annotation are imaginary elements not present in the real world. Then, Figure 3.4 illustrates that IFC Flow Terminal, IFC Flow Controller, and IFC Flow Fitting do not belong to the structural building elements. After that, Figure 3.4 displays that the table, chair, and storage belonging to the IFC Element Furniture do not have detailed shapes. Last, Figure 3.3 shows that IFC Footing is located below the building and not in the indoor scenes.

Table 4. List of IFC Class Not Used in this Research

No	IFC Class	The reason why it is not used
1	IFC Annotation	Not present in the actual building
2	IFC Covering	
3	IFC Space	
4	IFC Element Assembly	Not present in the indoor scenes
5	IFC Footing	
6	IFC FlowController	Do not belong to the building's structural elements
7	IFC FlowFitting	
8	IFC FlowTerminal	
9	IFC Element Furniture	

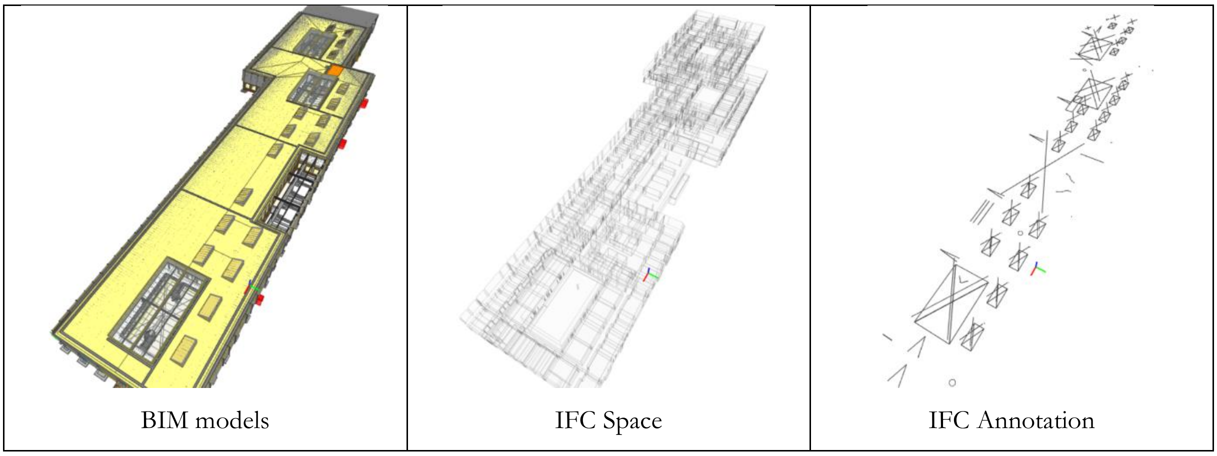


Figure 3.2 IFC Space and IFC Annotation in BIM models (Source: Author)

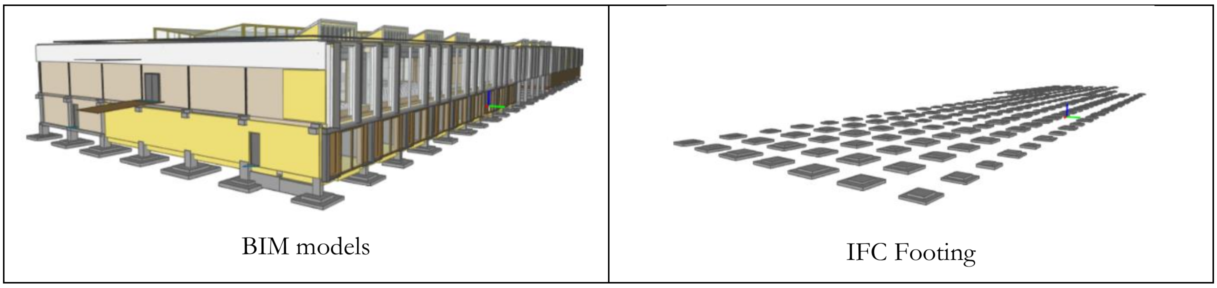


Figure 3.3 IFC Footing in BIM models (Source: Author)

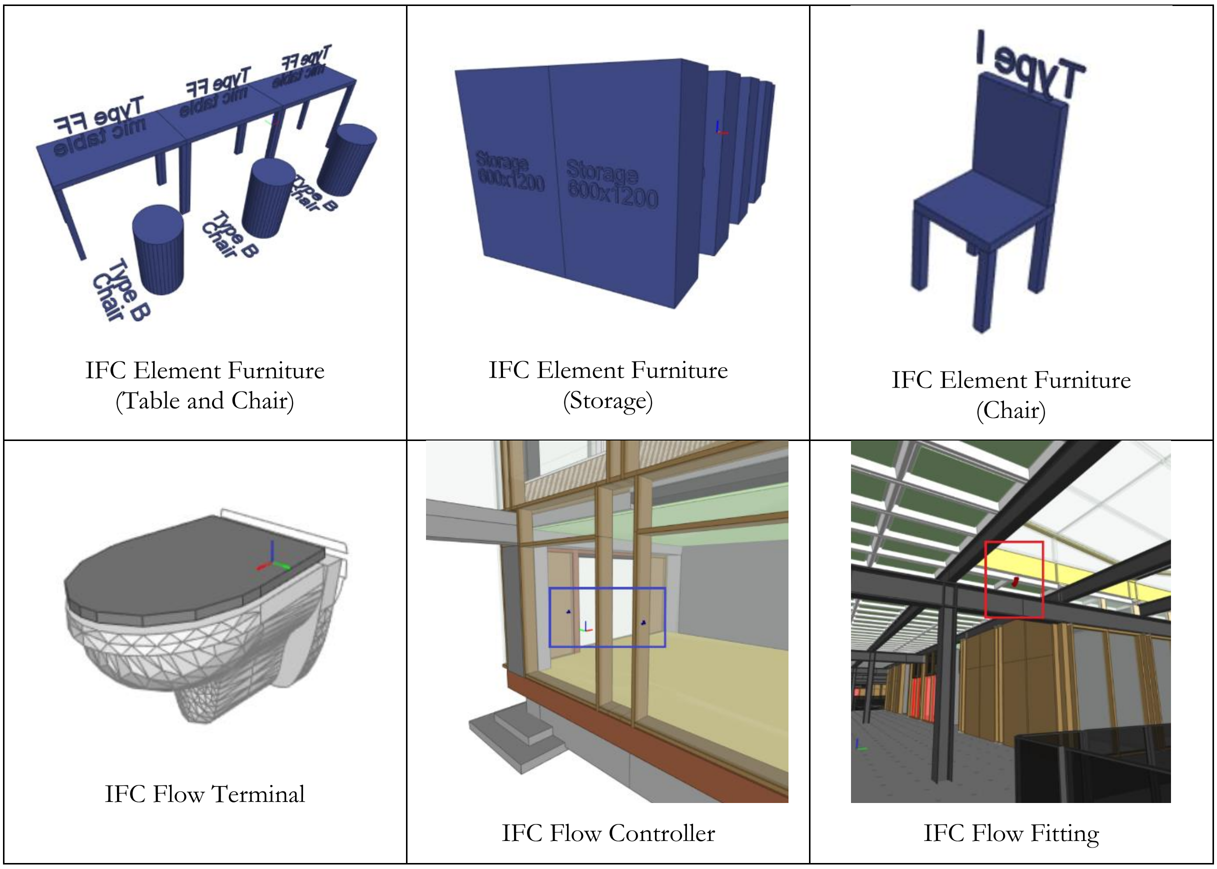


Figure 3.4 IFC Classes not Belong to the Structural Element (Source: Author)

Table 5 describes the IFC classes that need the classification process. The classification process is needed because one IFC class can include multiple other IFC classes (Son & Kim, 2010). Figure 5.1 shows IFC Curtain Wall contains multiple IFC classes, including IFC Door, IFC Wall, and IFC Window. Furthermore, Figure 3.6 shows that even IFC Stair does not contain IFC Slab and IFC Railing, it still includes elements similar to slab/floor and railing. Therefore, the IFC Curtain Wall rand IFC Stair require more classification to separate them into multiple IFC classes.

Table 5. A list of IFC class need to be classified

No	IFC Class	Activity
1	IFC Building Element Proxy	The object must be classified as IFC Door, IFC Stair, IFC Wall, IFC Railing, and IFC Window.
2	IFC Curtain Wall	
3	IFC Member	
4	IFC Plate	
5	IFC Ramp	The object needs to be classified as IFC Floor
6	IFC Stair	The object needs to be classified as IFC Floor, IFC Railing, and IFC Stair
7	IFC Standard Wall	The object needs to be classified as IFC Wall
8	IFC Stairway	The object needs to be classified as IFC Stair
9	IFC Ramp Flight	The object needs to be classified as IFC Floor

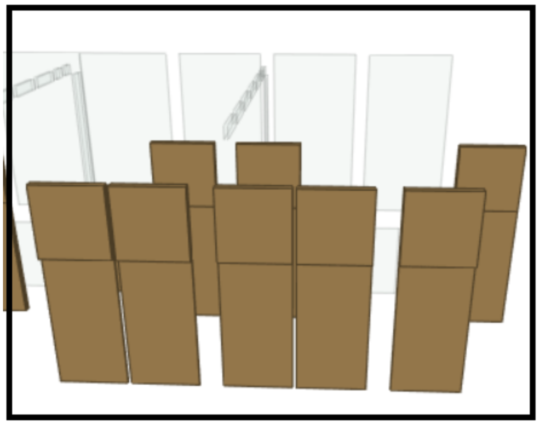


Figure 3.5. IFC Curtain Wall in BIM models
(Source: Author)

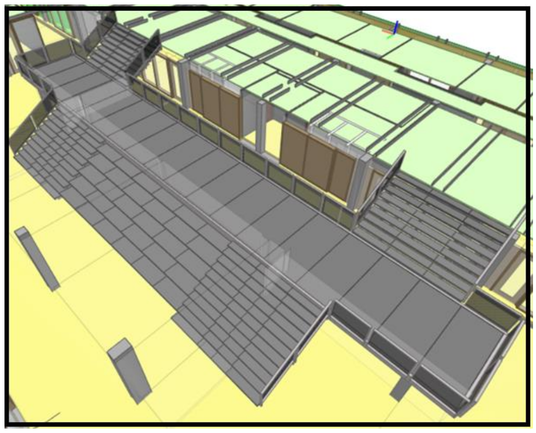


Figure 3.6. IFC Stair in BIM models
(Source: Author)

3.2. OBJ Conversion

The Wavefront OBJ format is the standard for the synthetic point clouds generation tool of Blensor (Gschwandtner et al., 2011) and CloudCompare (CloudCompare - Home, 2023). Therefore, the Blender BIM plugin (*BlenderBIM Add-on - Beautiful, Detailed, and Data-Rich OpenBIM*, 2023) converts the IFC format to OBJ format. Since the Wavefront OBJ format does not include semantic information, the element class label is put in the file name.

3.3. Synthetic Point Clouds Generation

This research utilizes four methods of synthetic point cloud generation: 1) Ideal method, 2) Simulated method, 3) Methods with noise, and 4) Methods with transparent glass.

3.3.1. Ideal Method

Derived from the method proposed by Ma et al. (2020) described in Section 2.1.3, this method directly converts the surface geometries of the BIM models into synthetic point clouds illustrated in Figure 3.7. Using the open-source CloudCompare (CloudCompare - Home, 2023), this method randomly samples the point clouds with the density based on the density of the point clouds to be predicted. Then, the process is performed individually per element class. The synthetic point clouds are stored in the .txt format with ASCII standard. The label corresponding to the element class is kept in the file name.

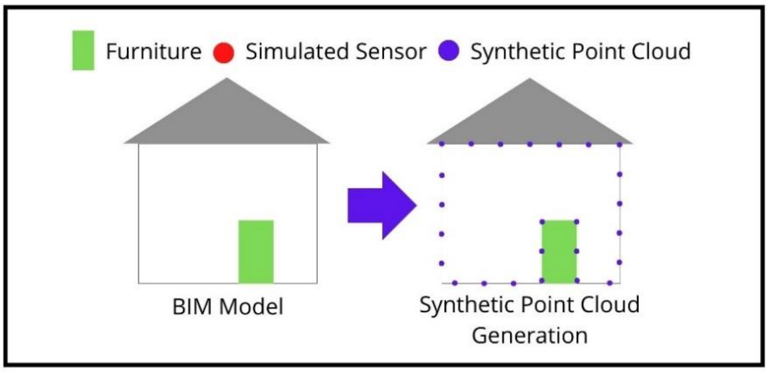


Figure 3.7. The Ideal Method of Synthetic Point Cloud Generation (Source: Author)

3.3.2. Simulated Method

Based on the method proposed by Griffiths & Boehm (2019b) described in Section 2.1.4, this method utilizes laser scanning simulation in the BIM models to generate synthetic point clouds, as illustrated in Figure 3.8. Using the open-source software Blensor (Gschwandtner et al., 2011), this method places multiple virtual sensor systems in the BIM models with the configuration based on the sensor system model that acquire the point clouds to be predicted. The configuration used in this research can be seen in Table 6. A set of point clouds is produced for each sensor system. Then, the point clouds derived from all scans are merged into one set of point clouds. Registration is unnecessary because the scans are aligned in the same coordinate system. The resulting synthetic point clouds are stored in the .txt format with ASCII standard.

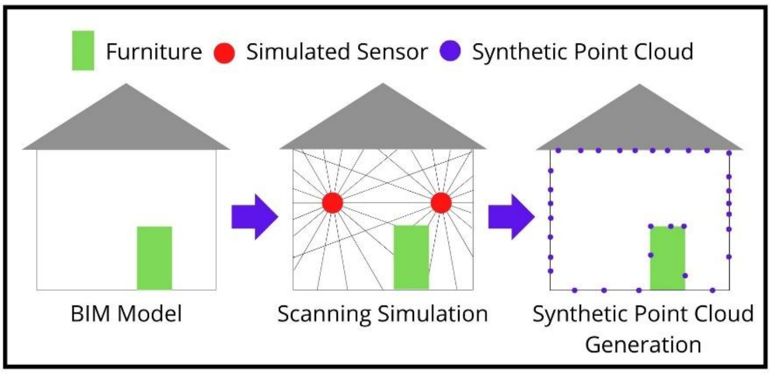


Figure 3.8. The Simulated Method of Synthetic Point Cloud Generation (Source: Author)

Table 6. Parameters for the Virtual Laser Scanner

Simulation Parameters	Value
Vertical Laser Angle's Range	-60.00 to 60.00 degree
Vertical Laser Angle's Resolution	0.25 degree
Horizontal Laser Angle's Range	-180.00 to 180.00 degree
Horizontal Laser Angle's Resolution	0.25 degree
Maximum Distance	50 meter
Sensor System Height	1.5 meter

Since Blensor only takes the Wavefront OBJ, which does not have semantic information, a semantic label can not be directly assigned to the synthetic point clouds during the laser scanning simulation. Hence, an annotation process is utilized. The annotation process uses the mesh-to-point nearest distance method from CloudCompare. It selects the point clouds with a 0.001 m distance from the 3D model with one building element class. Then, the selected point clouds are exported into a new set of point clouds with the label based on the building element class. This method is performed iteratively per each class so that point clouds with different classes are separated. The synthetic point clouds are stored in the .txt format with ASCII standard. Then, the label for the point cloud is stored in the file name.

3.3.3. Ideal and Simulated Method with Noise

For the Ideal Method with Noise, multiple noise value is defined using the normal distribution of the Gaussian function illustrated in Equation (4). Then, the x value of this function is derived from the random value ranging from 0.00 to 1.00. After that, the noise value is added to the X, Y, and Z coordinate feature of the Synthetic Point Clouds generated from the Ideal Method.

For the Simulated Method with Noise, the open-source software Blensor (Gschwandtner et al., 2011) offers noise simulation in synthetic point clouds. The Blensor simulated the sensor system noise in the distance between the sensor system and the 3D models of the building elements. It needs the configuration for the Gaussian function's mean and standard deviation, as seen in Equation (4).

$$y(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \qquad \text{Equation (4)}$$

3.3.4. Ideal and Simulated Method with Transparent Glass

This method redos the class definition from Section 3.1. It removes the IFC Window and utilizes IFC Member as the window's frame. After that, this method uses a similar approach to the Ideal Method or Simulated Method.

3.4. Data Preparation

This step synchronizes the file data composition for the synthetic point clouds with the S3DIS dataset (Armeni et al., 2016). The first data preparation process is to separate these files into multiple Rooms and save them in different folders. The process uses the point-to-mesh distance method of the open-source software CloudCompare. Then, the room assignment is based on the IFC Space from the BIM models.

The second data preparation process is to group the Room folders into six different Areas and save them in six different folders. One Area folder only takes the Room folders from the same floors. This research classified the Room into five types: Hallway, Office, Stair, Storage, and WC. The name for each Room folder is named based on its Room type combined with its index. The name for each synthetic point cloud file is named based on its building element class.

3.5. Data Normalization

This step is done because similar feature units make the network easily focus on the close relationships between values rather than their absolute values (K. Dhana Sree & C. Shoba Bindu, 2018). It converts the range in the point cloud's features, especially X, Y, and Z coordinate geometric features, to a common distribution. In particular, it moves all point clouds in one area so that a point cloud that originally has the lowest coordinate value is located at the coordinate origin (0,0,0). The normalization is done using the transformation method of the open-source software CloudCompare.

Moreover, point clouds for indoor scenes that include multiple stories of the element must be separated based on the floor and bring the elements from the upper floor to the ground floor. Most publicly available point cloud datasets, such as S3DIS (Armeni et al., 2016), only include single-story buildings. If these datasets directly classify the elements on the second floor, it can identify the floor element belonging to the second floor as a ceiling element, causing misclassifications.

3.6. Point Cloud Classification

With an adequate performance in indoor scenes described in Section 1.2, the Kernel Point Fully Convolutional Network, or KP-FCNN (Thomas et al., 2019), is chosen for point cloud classification. This research utilized the Rigid version. The reason is that this research focuses on the building elements (e.g., window, wall, and door) with simple shapes as it can accomplish more on simple shapes elements compared to Deform version. The network training parameters used in this research are summarized in Table 7. Then, this research utilized the network parameters listed in Table 8. The motivations behind the configuration of these parameters are explained in the sub-Section below.

Table 7. Network Training Parameter used in this Research

Training Parameter	Value
Maximum epoch	400
Optimizer	Adam
Momentum Gradient Descent	0.98
Initial Learning Rate	10^(-2)
Batch size	6

Table 8. Two Network Parameters of KP-FCNN used in this Research

Network Parameters	**Name**	
	Network Parameter – 1	**Network Parameter – 2**
the input features number Din	5	5
the size of the first voxel grids dl_0	0.04 m	0.04 m
the input sphere radius r	2.00 m	2.00 m
the convolution radius r	2.50 m	2.50 m
the kernel influence distance σ	1.20 m	1.20 m
the number of kernel points K	11	15
Sampling Strategy	Regular	Random

3.6.1. Simulated The Configuration of the Input Features Number Din

The geometry features give a relevant understanding of building elements' shapes and spatial distribution (Zhai et al., 2022). In particular, ceiling and beam elements are located at the top of other elements, while the floor is at the bottom. Then, window, wall, and door elements are usually adjacent. Therefore, the geometry features are included as the first three columns of the point clouds. Contrarily, the color features are insignificant because most elements have the same color of grey, as seen in Figure 3.9.

Consequently, it does not give unique information for each element. Nevertheless, since KPConv still requires the color feature, it is still included as the fourth column of the point cloud. They are assigned the constant value of 255.0. Additionally, the KPConv requires one standard constant feature, which only affects the value of input features number Din (Thomas et al., 2019). Therefore, the input features Din used in this research is 5.

Figure 3.9. Construction Scenes of the New ITC Building (Source: Author)

3.6.2. The Configuration of the Size of the First Voxel Grids dl_0

The building elements can vary in size. There are the large elements (e.g., door, wall, and window) and the small elements (e.g., beam, column, and railing). To be able to capture large elements while also being capable of capturing small elements, this research requires a larger size of the first voxel grids dl_0 compared to the default value. With that, the size of the first voxel grids dl_0 used in this research is 0.04 m.

3.6.3. The Configuration of the Input Sphere Radius r, the Convolution Radius r, and the Kernel Influence Distance σ

Thomas et al. (2019) advise that the input sphere radius r should be 50 times the size of the first voxel grids dl_0. It is done so that the voxel grid size on the last layer will not exceed the input sphere radius r. In addition, window, wall, and door elements have similar shapes but have a unique element distribution. As a result, this research requires a larger value of the input sphere radius r, the convolution radius r, and the kernel influence distance σ with 2 m, 2.5 m, and 1.2 m, respectively.

3.6.4. The Configuration of the Number of Kernel Points K, Sampling Method, and Training Parameters

In the indoor building scenes, there are majority elements (e.g., ceiling, floor, and wall) with a huge quantity of point clouds and minority elements (e.g., beam, column, and stair) with a small quantity. It makes the dataset unbalanced since the majority elements dominate the minority element. Smaller kernel points work best with a majority of large elements, while larger kernel points work best with a minority of small elements (Thomas et al., 2019). Then, the regular picking strategy works best with most large elements, and the random picking strategy works best with a minority of small elements. Therefore, to overcome the unbalanced dataset, this research used two configurations of these parameters. The first is 11 kernel points K with the random picking strategy. The second is 15 kernel points K with the regular picking strategy.

3.7. Evaluation

Based on the metrics used to evaluate most of the point cloud dataset, including the S3DIS dataset (Armeni et al., 2016), the quantitative performance for point cloud classification per class is evaluated using Precision, Recall, F-1 score, and Intersection Over Union or IoU. Then, the average values of IoU are computed to represent the overall performance. This research does not use Overall Accuracy since building elements in the indoor scene datasets can be unbalanced, as explained in Section 3.6.4. The reason is that Overall Accuracy measures the percentage of the correctly predicted datasets where the majority elements have a higher influence on the results than the minority elements (Everingham et al., 2010). It can lead to poor evaluation of the network with unbalanced datasets.

Precision, Recall, and F-1 Scores are appropriate for performance evaluation for each class (Everingham et al., 2010). These metrics are not meant for the whole class because the averaged value of all classes does not represent the datasets correctly predicted as negative. Precision measures the percentage of the actual datasets correctly predicted for one class. Then, Recall measures the percentage of the dataset's prediction that is correct for one class. After that, F-1 Score combines Precision and Recall using a harmonic average. Nevertheless, the average value of Intersection over Union or IoU for the whole class can represent the network classification's performance since it applies the same influence for all classes (Everingham et al., 2010). It measures the percentage of the union of predicted and actual datasets correctly predicted for one class.

4. DATA AND TOOLS

The primary datasets used in this research are the BIM model, the New ITC Building Point Clouds datasets, and the S3DIS dataset (Armeni et al., 2016). They are described in Section 4.1, Section 4.2, and Section 4.3. Then, this research used the hardware and the software with explanations under Section 4.4 and Section 4.5, respectively.

4.1. The BIM model for the New ITC Building

This BIM model covers multiple elements in indoor scenes, including beams, columns, doors, railing, roof, floor, stairs, walls, and windows. It has the geometric and semantic information of building elements standardized in Industry Foundation Class format or IFC. It is used as a design model for the construction phase of the new ITC building, as seen in Figure 4.1. The new ITC building is located at Hallenweg 21, 7522 NH Enschede, the Netherlands. The location of the building can be seen in Figure 4.2. It is a two-story building with 220m length and 50m width. It is possible to use the dataset since both are provided and owned by the ITC, Faculty Geo-Information Science and Earth Observation, University of Twente.

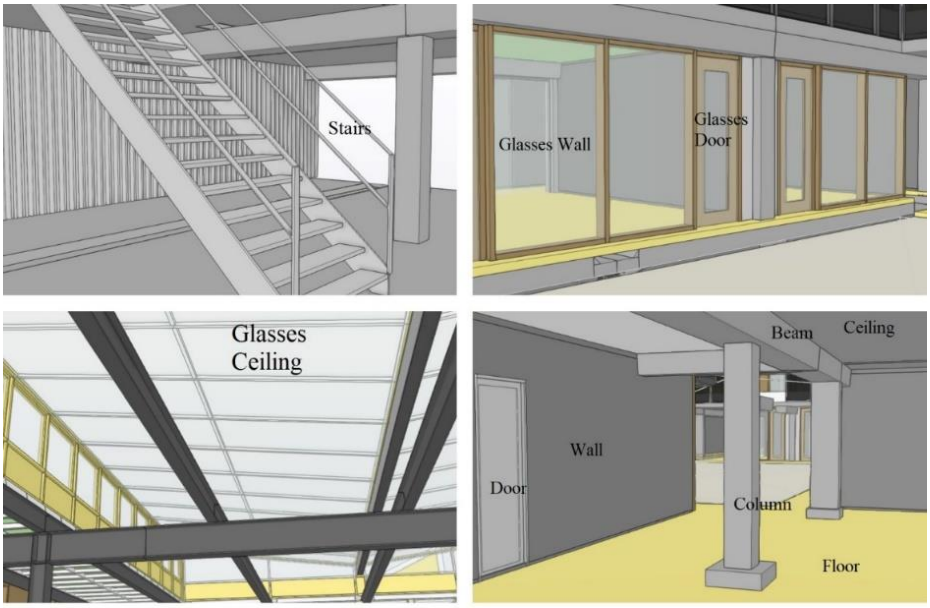


Figure 4.1. BIM model for the New ITC Building (Source: Author)

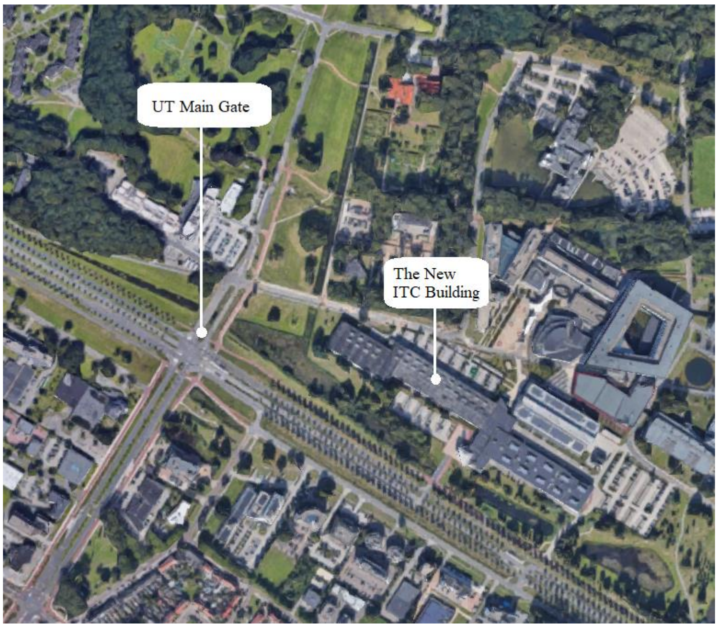


Figure 4.2. The Location for the New ITC Building (Source: Google Maps)

4.2. The New ITC Building Point Cloud Datasets

The point clouds are derived from the indoor scenes of the new ITC building during the construction phase. The sensor system model used is Riegl VZ-400i. 155 scan positions were used to measure the point clouds. Table 9 describes the specification of the sensor system during the data measurement. The point clouds are manually labeled within the Digital Twin@ITC project.

The New ITC Building Point Cloud Datasets consist of the ITC 2022 dataset acquired in April 2022 and the ITC 2021 dataset acquired in April 2021. Figure 4.3 illustrates both datasets. The figure shows that the ITC 2022 dataset consists of multiple building elements, including beams, columns, railing, roof, floor, stairs, walls, and windows. Then, unlike the ITC 2022 dataset, the ITC 2021 dataset does not have a stair element.

Table 9. Configurations of Terrestrial Laser Scanning Simulation

Specifications	Value
Measurement Range	1.5m to 600m
Accuracy	5mm
vertical Field of View	1000 deg
Horizontal Field of View	3600 deg
Pulse repetition rate	Up to 300,000 points per second
Laser wavelength	1550 nm

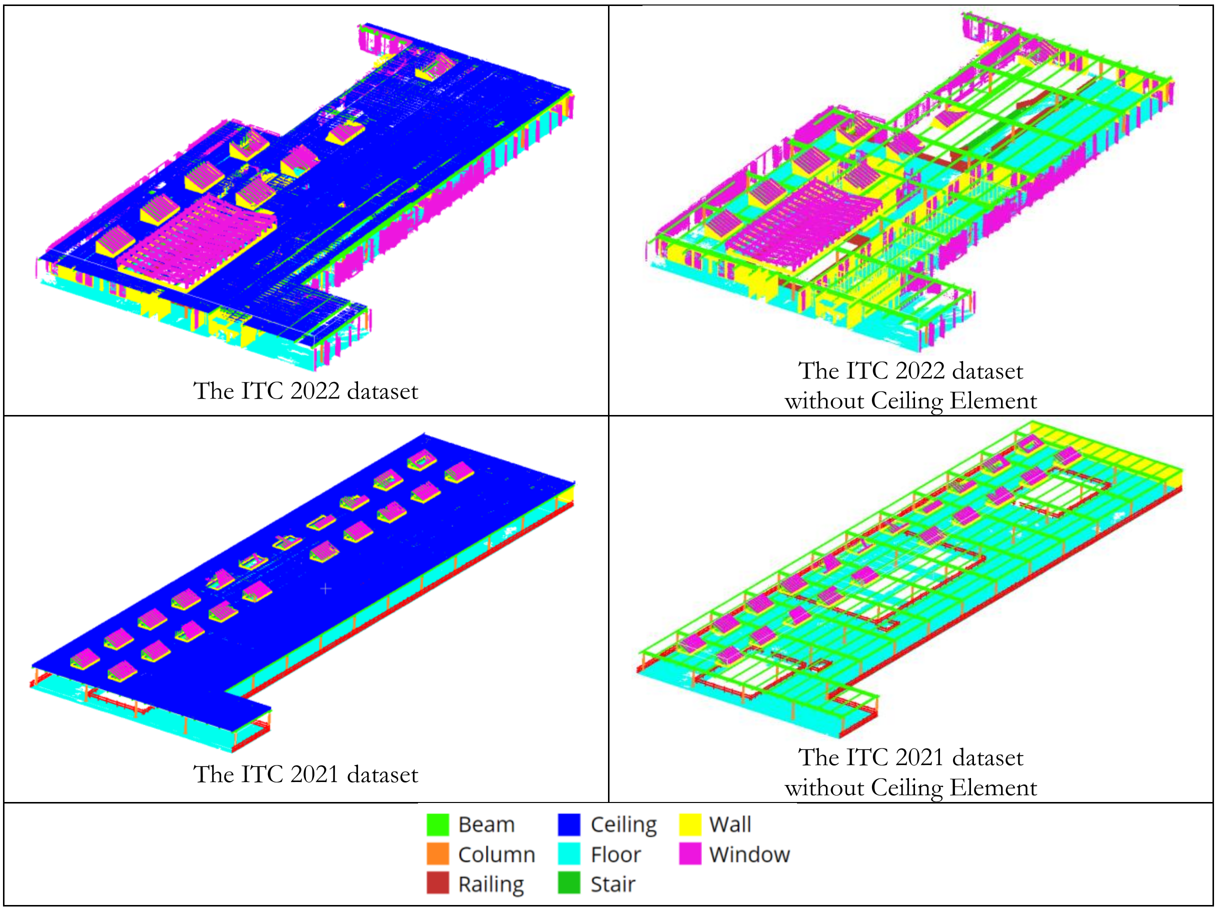


Figure 4.3 The New ITC Building Point Cloud Datasets (Source: Author)

4.3. The Stanford Large-Scale 3D Indoor Spaces or S3DIS dataset

As described in Section 1.3, this dataset is one of the benchmark point cloud datasets publicly available (Armeni et al., 2016). Stanford University developed the dataset. It contains over 215 million point clouds of 271 rooms from three different buildings. Each point cloud belongs to one of the multiple semantic classes, which regular indoor scenes have, including permanent elements (i.e., ceiling, floor, wall, beam, column, stairs, window, and door), furniture (i.e., table, sofa, board, bookcase, and chair), and clutter.

4.4. Hardware

BIM model conversion to synthetic point clouds is executed on a local computer. It has a 64-bit processor and 475.0 Giga Bytes memory. It has a Central Processing Unit (CPU) of AMD Ryzen 5 5600H with Radeon Graphics and a Graphics Processing Unit (GPU) of NVIDIA GeForce RTX 3050 Ti.

The classification process is performed on the remote Linux server provided by the Faculty of ITC, University of Twente. It has a 64-bit processor and 256 Giga Bytes memory. It has a Central Processing Unit (CPU) of Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz and a Graphics Processing Unit (GPU) of NVIDIA A40.

4.5. Software

Synthetic point clouds generation is executed using the open-source Blensor (Gschwandtner et al., 2011) and CloudCompare (*CloudCompare - Home*, 2023). Then, Python programming language with version 3.8 is used to manage the synthetic point clouds. It has multiple libraries, including Numpy, Scikit-learn, Pandas, and Matplotlib. These processes are carried out on a local computer.

For the classification of point clouds, this research used the open-source PyTorch implementation for the KP-FCNN network (Thomas et al., 2019). The network is carried out in a conda virtual environment, with Python3.8, PyTorch 1.10.2, CUDA 11.3, and cuDNN 8.2.0. The open-source PuTTY (*Download PuTTY: Latest Release (0.78)*, 2023) is used for accessing the remote Linux server, while WinSCP (*WinSCP :: Official Site :: Free SFTP and FTP Client for Windows*, 2023) is used for data transfer.

5. EXPERIMENTS

Based on the objective mentioned in Section 1.6, this research executes four experiments to answer the research questions.

5.1. Experiment 1 – Comparing the Synthetic Point Clouds and the S3DIS Dataset

Driven by the limitations explained in Section 1.3 and based on the hypothesis described in Section 1.4, an experiment is conducted to investigate the viability of the BIM models in point cloud classification. This experiment compares two networks trained with distinct datasets: 1) the synthetic point clouds generated from the BIM models and 2) the point clouds derived from the publicly available benchmark point clouds dataset, S3DIS (Armeni et al., 2016).

5.2. Experiment 2 – Comparing the Ideal and the Simulated Method of Synthetic Point Clouds Generation

As described in Section 2.1.4, no research has been conducted on the importance of the occlusion effect and the local point cloud distribution in synthetic point clouds. Then, using the second hypothesis made in Section 1.5, an experiment is held to compare two different methods for synthetic point cloud generation: 1) the Ideal Method that does not have the occlusion effect and has random local point cloud distribution and 2) the Simulated Method simulate the occlusion effect and the local point cloud distribution from the real point clouds.

5.3. Experiment 3 – Comparing the Synthetic Point Clouds on Varying Levels of Sensor System Noise

Motivated by the methods proposed by Griffiths & Boehm (2019b) explained in Section 2.1.5 and based on the second hypothesis made in Section 1.5, an experiment is carried out to confirm the influence of the sensor system noise of the synthetic point clouds in the classification performance. This experiment compares the network trained with the synthetic point clouds with seven different sensor system noise levels, including 0 m, 0.005 m, 0.01 m, 0.03 m, 0.05 m, 0.1 m, and 0.3 m.

5.4. Experiment 4 – Comparing the Synthetic Point Clouds that Consider the Glass as Transparent and Non-Transparent

This research presents an experiment that explores whether glass elements should be deemed transparent in synthetic point clouds and its impact on the classification performance of window elements. It is derived from the hypothesis and motivation described in Section 1.5 and Section 2.1.6, respectively. This experiment compares two networks trained with distinct datasets: 1) synthetic point clouds considering glass objects as transparent, and 2) synthetic point clouds considering glass objects as non-transparent. Both datasets are generated using the Ideal Method.

5.5. Experiment 5 – Augmenting the Synthetic Point Clouds and the S3DIS Dataset

With the third hypothesis made in Section 1.6, this research conducts an experiment to adapt the synthetic point clouds utilizing the S3DIS dataset. In particular, this experiment evaluates the network trained with the combination of the synthetic point clouds and the S3DIS dataset. Then, the results are compared with the network trained only on the synthetic point clouds or the S3DIS dataset alone.

5.6. Experiment 1 – 5 Execution

To experiment 1 to 5, this research used the methodology described in Section 3. The workflow converts the BIM model, used as a design model to construct the New ITC Building explained in Section 4.1, into 16 sets of synthetic point clouds. Table 10 describes the name and configurations of these sets. Then, the network is evaluated using the ITC 2022 Dataset described in Section 4.2.

Table 10. Sets of Synthetic Point Clouds Generated in this Research

No	Name	Generation Method	Noise	Transparent Glass
1	Synthetic Point Clouds - 1a	Ideal Method	0 m	No
2	Synthetic Point Clouds - 1b	Ideal Method with Noise	0.005 m	No
3	Synthetic Point Clouds - 1c	Ideal Method with Noise	0.01 m	No
4	Synthetic Point Clouds - 1d	Ideal Method with Noise	0.03 m	No
5	Synthetic Point Clouds - 1e	Ideal Method with Noise	0.05 m	No
6	Synthetic Point Clouds - 1f	Ideal Method with Noise	0.1 m	No
7	Synthetic Point Clouds - 1g	Ideal Method with Noise	0.3 m	No
8	Synthetic Point Clouds - 1h	Ideal Method with Transparent Window	0 m	Yes
9	Synthetic Point Clouds - 2a	Simulated Method	0 m	No
10	Synthetic Point Clouds - 2b	Simulated Method with Noise	0.005 m	No
11	Synthetic Point Clouds - 2c	Simulated Method with Noise	0.01 m	No
12	Synthetic Point Clouds - 2d	Simulated Method with Noise	0.03 m	No
13	Synthetic Point Clouds - 2e	Simulated Method with Noise	0.05 m	No
14	Synthetic Point Clouds - 2f	Simulated Method with Noise	0.1 m	No
15	Synthetic Point Clouds - 2g	Simulated Method with Noise	0.3 m	No
16	Synthetic Point Clouds - 2h	Simulated Method with Transparent Window	0 m	Yes

5.7. Experiment 6 – Applying the Proposed Approach on Different Construction Stages

Similar to Experiment – 1, this research experiment on how robust this approach is to different stages of building construction. This experiment compares two networks trained with distinct datasets: 1) the synthetic point clouds and 2) the S3DIS dataset. The networks are tested using the ITC 2021 dataset described in Section 4.2. Then, the results are compared with the networks tested using the ITC 2022 dataset.

6. RESULTS

This Chapter presents the classification results of the networks trained with every synthetic point cloud and S3DIS dataset (Armeni et al., 2016). The networks are evaluated using multiple measures, including Precision, Recall, F-1 Score, and Intersection over Union or IoU. Since the test dataset does not have a door element, the door element is excluded from the calculation for the network's overall performance.

6.1. Networks Trained using Network Parameter – 1 and Tested on the ITC 2022 dataset

6.1.1. Networks Trained with the Synthetic Point Clouds Generated from the Ideal Method

Table 11 shows the performance of the Ideal method, where it attains 39.98% m-IoU as the highest network overall performance from the Synthetic Point Clouds – 1e. Most network has floor element with high classification result. It has the IoU ranged from 7.66% to 92.87%. Then, huge amounts of stair elements are unidentified since 15.4% IoU is the highest classification result. After that, as the noise in the synthetic point clouds increased to 0.05 m, the network overall performance also increased. However, overall network performance is reduced as the noise increases beyond 0.05 m. After that, the synthetic point clouds that consider the glass a transparent object reduced the network's overall performance.

Table 11. Results of the Network Trained with the Synthetic Point Clouds Generated from the Ideal Method

Synthetic Point Clouds Dataset	Noise (m)	Transparent Glass	Evaluation									
			m-IoU (%)	IoU for each Building Element (%)								
				Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
– 1a	0.000	No	20.94	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
– 1b	0.005	No	23.78	47.26	11.28	0.00	11.49	0.92	49.62	15.04	42.31	12.31
– 1c	0.010	No	29.16	46.13	28.57	0.00	32.29	12.13	51.88	3.06	40.43	18.81
– 1d	0.030	No	35.62	60.72	7.17	0.00	36.20	46.69	64.14	7.70	34.40	27.97
– 1e	0.050	No	39.98	62.86	17.12	0.00	0.00	86.75	92.87	0.00	51.96	8.27
– 1f	0.100	No	36.56	62.83	0.00	0.00	8.92	72.40	73.54	4.83	43.82	26.17
– 1g	0.300	No	12.93	1.24	0.01	0.00	5.78	62.00	7.66	0.00	26.74	0.00
– 1h	0.000	Yes	25.43	40.78	11.92	0.00	23.36	41.97	41.81	6.65	31.95	5.03

6.1.2. Networks Trained with the Synthetic Point Clouds Generated from the Simulated Method

Table 12 shows that the Simulated method achieves an m-IoU of 51.54% as the highest network overall performance from the Synthetic Point Clouds – 2c. Like the Ideal method results, overall network performance increases as the noise in the synthetic point clouds increases to 0.01 m. Then, beyond 0.01 m noise level, the network's overall performance is reduced. Unlike the Ideal method results, considering glass as a transparent object does not reduce the network's overall performance. Instead, it has comparable results. The networks, up to 0.001 m noise level, accurately classified ceiling and floor elements where each element gained IoU higher than 80.00%. Meanwhile, the window element is poorly classified, with 24.24% IoU as the highest classification result.

Table 12. Results of the Network Trained with the Synthetic Point Clouds Generated from the Simulated Method

Synthetic Point Clouds Dataset	**Noise (m)**	**Transparent Glass**	**Evaluation**									
			m-IoU (%)	**IoU for each Building Element (%)**								
				Beam	**Column**	**Door**	**Railing**	**Ceiling**	**Floor**	**Stair**	**Wall**	**Window**
– 2a	0.000	No	38.35	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24
– 2b	0.005	No	40.47	47.13	5.08	0.00	21.98	84.03	85.46	24.01	32.24	23.81
– 2c	0.010	No	51.54	67.95	30.45	0.00	52.25	84.38	92.62	24.93	36.94	22.80
– 2d	0.030	No	44.91	72.20	35.08	0.00	32.17	62.28	72.24	25.48	48.22	11.59
– 2e	0.050	No	33.82	48.47	18.46	0.00	19.07	53.13	63.49	8.14	46.29	13.52
– 2f	0.100	No	28.10	50.42	14.19	0.00	34.51	8.76	43.41	28.53	44.21	0.74
– 2g	0.300	No	24.45	1.98	2.03	0.00	0.00	60.60	90.18	0.00	38.66	2.18
– 2h	0.000	No	38.50	35.26	15.49	0.00	60.09	84.39	54.01	13.76	36.39	8.60

6.1.3. Networks Trained with the S3DIS dataset

The overall performance of the network is shown in Table 13. It has m-IoU of 37.32%. It performs better than seven out of eight networks generated from the Ideal method. Contrarily, it performs less than most of the networks generated from the Simulated method (five out of eight networks). Like the networks generated from the Ideal and Simulated method, the ceiling and floor elements are the most well-classified, with IoU higher than 70.00%. Column element is the lowest classification result of the network, with 4.28% IoU. Then, the network also fails to identify the beam, wall, and window element, where it only achieves IoU lower than 20.00%. Nevertheless, unlike the networks generated from the Ideal and Simulated method, it has a high stair classification result, with 76.07% IoU. Then, it has 0.00% IoU for the railing element since the S3DIS dataset does not provide any point clouds.

Table 13. Results of the Network Trained with the S3DIS dataset

Dataset	Evaluation									
	m-IoU (%)	IoU for each Building Element (%)								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
S3DIS	37.32	17.80	4.28	0.00	0.00	72.88	92.80	76.07	17.77	16.93

6.1.4. Networks Trained with the Augmentation of Synthetic Point Clouds – 2c and S3DIS dataset

As the network with the highest overall performance compared to others, it has 51.5% m-IoU. The network has the highest overall performance compared to all previously mentioned networks. It has 55.01% m-IoU. Comparable to previous networks, the element with significant classification results is ceiling and floor elements, with IoU higher than 80.0%. Even though the network has the highest overall performance compared to others, the network is still unable to identify column elements completely.

Table 14. Results of the Network Trained with the Augmentation of Synthetic Point Clouds – 2c and S3DIS dataset

Dataset	Evaluation									
	m-IoU (%)	IoU for each Building Element (%)								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
Synthetic Point Clouds – 2c + S3DIS	55.01	69.09	24.97	0.00	64.71	86.28	92.88	32.96	38.14	31.06

6.2. Networks Trained using Network Parameter – 1 and Tested on the ITC 2021 dataset

ITC 2021 dataset does not include a stair element. As a result, the classification result for it is 0.00%. Therefore, the calculation of the m-IoU excludes the stair element. Table 15 shows the network attains 61.39% as the highest overall network performance. Similar to previous networks, elements with extensive coverage and simple geometry, like ceiling and floor, achieve higher classification results than others. Then, railing and window elements are the least well-classified, with IoU lower than 10%. Nevertheless, compared to the networks trained on the ITC 2022 dataset, it has higher classification results for beam and column elements.

Table 15. Results of the Network Trained using Network Parameter – 1 and Tested on the ITC 2021 dataset

Dataset	Evaluation									
	m-IoU (%)	IoU for each Building Element (%)								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
SPC – 1a	42.39	49.33	86.71	0.00	0.23	60.56	99.19	0.00	0.35	0.35
SPC – 1b	40.43	57.93	86.05	0.00	7.20	47.11	73.24	0.00	1.35	10.09
SPC – 1c	52.96	68.56	80.96	0.00	0.00	82.13	98.20	0.00	40.14	0.75
SPC – 1h	39.18	44.93	89.65	0.00	0.03	29.89	60.19	0.00	49.41	0.14
SPC – 2a	56.79	85.24	69.29	0.00	0.31	90.86	98.20	0.00	51.95	1.71
SPC – 2b	59.24	87.37	73.14	0.00	0.57	91.52	98.39	0.00	63.51	0.18
SPC – 2c	61.39	86.88	87.07	0.00	7.14	92.18	98.29	0.00	56.54	1.66
SPC – 2h	59.82	84.89	75.52	0.00	0.83	92.25	98.94	0.00	65.24	1.04
S3DIS	33.89	31.16	1.36	0.00	0.00	77.47	98.78	0.00	28.48	0.00

SPC = Synthetic Point Clouds

6.3. Networks Trained using Network Parameter – 2 and Tested on the ITC 2022 dataset

As seen from Table 16, similar to the networks trained using Network Parameter – 1, the highest network overall performance of these networks is achieved from the Synthetic Point Clouds – 2c, with m-IoU of 48.10%. The structural elements, like the ceiling and floor, are well-classified, and the column element is poorly identified. After that, the network's overall performance increases as the noise in the synthetic point clouds increases.

Table 16. Results of the Networks Trained using Network Parameter – 2 and Tested on the ITC 2022 dataset

Dataset	Evaluation									
	m-IoU (%)	IoU for each Building Element (%)								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
SPC – 1a	25.73	30.89	8.63	0.00	0.00	26.42	89.98	3.47	38.98	7.49
SPC – 1b	23.86	23.90	17.54	0.00	0.00	7.29	85.91	5.05	39.68	11.48
SPC – 1c	32.12	34.57	13.85	0.00	0.00	60.94	82.16	12.34	36.73	16.41
SPC – 1h	27.13	36.97	13.18	0.00	0.00	51.81	66.27	7.22	38.99	2.58
SPC – 2a	36.49	43.16	5.17	0.00	12.13	81.02	83.69	27.53	23.10	16.15
SPC – 2b	39.40	43.66	15.64	0.00	8.43	79.72	94.19	27.07	31.67	14.79
SPC – 2c	48.10	66.17	14.82	0.00	28.59	85.35	94.64	40.94	36.41	17.90
SPC – 2h	38.39	37.50	1.17	0.00	20.22	87.60	89.11	23.87	30.65	17.00
S3DIS	39.93	37.70	0.17	0.00	0.00	82.33	90.85	63.19	23.56	21.67

SPC = Synthetic Point Clouds

7. DISCUSSION

This Chapter analyses the results presented in Section 6 to answer the research questions mentioned in Section 1.7.

7.1. Common Performance for All Networks Trained using Network Parameter – 1 and Tested on ITC 2022 dataset

Based on the results reviewed in Section 6, the network trained on the Synthetic Point Clouds – 2c has the highest classification performance with 51.54% m-IoU. The network is generated from the Simulated method and includes a 0.01 m noise level. Then, most networks have the ceiling and floor as the highest classified elements, with the average IoU of 51.66% and 66.49%, respectively. On the contrary, most networks have the stair as the poorest-classified element, with an average IoU of 12.28%.

7.1.1. The Class Imbalance Problem

Low classification results of the networks can be attributed to the class imbalance problem. It occurs when there is an imbalance quantity of point clouds for each element that train the network (Zhang et al., 2020). KP-FCNN utilizes point-based cross entropy for the classification loss function. During the network training phase, it sums up the loss from each neighboring point cloud from any elements and updates the network parameters. However, the loss can be significantly influenced by the majority elements with larger point clouds than minority elements. As a result, the network captures fewer features from the minority elements and leads to poor classification performance for the minority elements.

As mentioned in Section 3.6.4, elements in indoor scenes of buildings are unbalanced where the majority elements (e.g., ceiling, floor, and wall) with a huge quantity of point clouds dominate the minority elements (e.g., column, railing, and stair) with a small quantity. Table 17 shows the number of point clouds and the classification result for each element for the network trained with Synthetic Point Clouds – 1a and Synthetic Point Clouds – 2a. From that table, it is apparent that there is a correlation between fewer point clouds and low classification results, especially for column, railing, and stair elements (marked in red). The ceiling and floor elements have the highest results as they have many point clouds (marked in green). Therefore, it can be concluded that the class imbalance problem in the datasets causes poor classification results.

Table 17. The Quantity and the Results of Network Trained with the Synthetic Point Clouds – 1a and the Synthetic Point Clouds – 2a

Dataset	Evaluation	Building Element								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
Synthetic Point Clouds – 1a	Quantity	8,972,447	1,860,634	9,516,129	772,409	43,554,213	20,977,986	1,044,355	44,280,252	12,433,382
	Quantity (%)	6.26	1.30	6.64	0.54	30.37	14.63	0.73	30.88	8.67
	IoU (%)	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
	m-IoU (%)	20.94								
Synthetic Point Clouds – 2a	Quantity	9,903,988	3,000,496	13,719,522	942,781	19,129,928	45,659,298	748,873	112,973,617	23,605,059
	Quantity (%)	4.31	1.31	5.97	0.41	8.33	19.88	0.33	49.19	10.28
	IoU (%)	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24
	m-IoU (%)	38.35								

7.1.2. The Inter-Class Similarity Problem

Low classification results of the networks can also be derived from the inter-class similarity problem. It can occur when different elements used to train the network exhibit similar appearances, which makes their geometry features indistinguishable (Venkataramanan et al., 2021). Since this research utilizes only the geometry features from the point clouds mentioned in Section 3.6.1, this condition confuses the network to differentiate between them. As a result, it leads to lower classification performance.

From Table 17, beam, column, and window elements have a high point cloud quantity but have lower classification results (marked in red). Table 18 shows the normalized precision confusion matrix for the network trained on the Synthetic Point Clouds – 2a. From that table, it can be seen that there is a high precision value between beam, wall, and window elements (marked in red), which means there is a huge confusion between them. Then, Figure 7.1, Figure 7.2, Figure 7.3, and Figure 7.4 illustrates the IFC data for column, beam, wall, and window elements. Figure 7.4 shows that the window element has multiple variations that share similar shapes to beam, column, and wall elements. As a result, it can be summarized that inter-class similarity problem in the datasets causes poor classification results.

Table 18. The Recall Matrix of the Network trained with the Synthetic Point Clouds – 2a

Recall Matrix (%)

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	24.41	0.00	0.00	0.00	0.59	0.00	0.00	0.39	0.74
column	0.02	2.15	0.00	0.08	0.00	0.00	0.00	0.51	3.12
door	0.00	25.81	0.00	36.68	0.00	0.02	0.05	16.92	11.66
railing	0.01	0.14	0.00	44.30	0.00	0.27	0.01	0.34	0.15
ceiling	11.60	0.00	0.00	0.00	95.04	0.00	0.00	0.37	10.47
floor	0.07	0.05	0.00	0.00	0.72	82.35	0.00	0.14	0.70
stair	1.52	0.27	0.00	6.82	0.20	7.59	99.83	0.49	1.64
wall	0.05	0.03	0.00	0.27	0.08	0.59	0.00	22.34	1.38
window	62.31	71.55	0.00	11.86	3.38	9.18	0.11	58.49	70.14

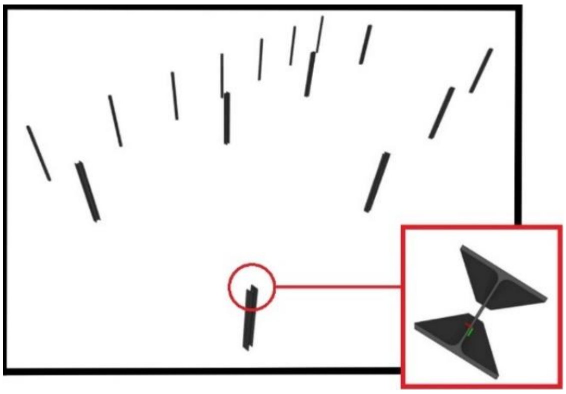


Figure 7.1. Column Element in the BIM models
(Source: Author)

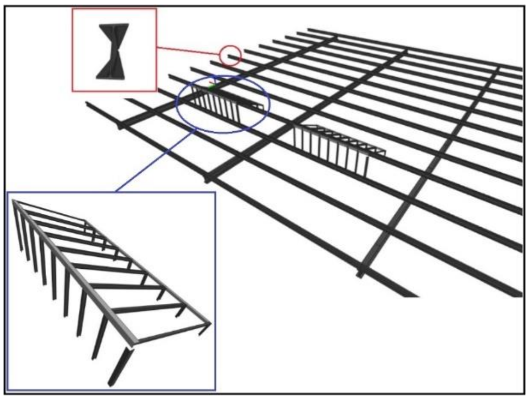


Figure 7.2. Beam Element in the BIM models
(Source: Author)

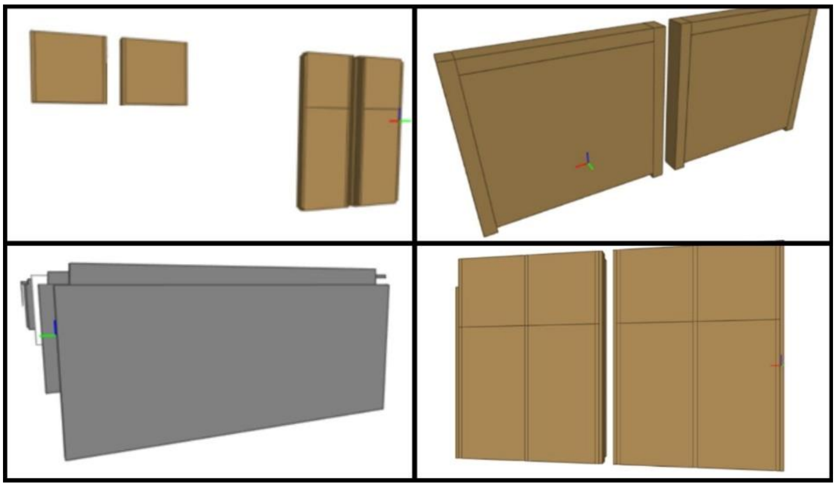


Figure 7.3. Wall Element in the BIM models
(Source: Author)

Figure 7.4. Window Element in the BIM models
(Source: Author)

7.1.3. Other Misclassification Problem

Additionally, certain wall elements are incorrectly classified as windows, not because of the class imbalance and the inter-class similarity problems. Figure 7.5 shows that this wall element has different shapes compared to the one in the synthetic point clouds or the BIM model. Instead, it has similar shapes to windows. This wall element is believed to be the same as the one presented in the BIM model but is still in construction. Thus, it can be inferred that incomplete elements pose a higher risk of low classification performance since they have a distinct appearance than finished ones.

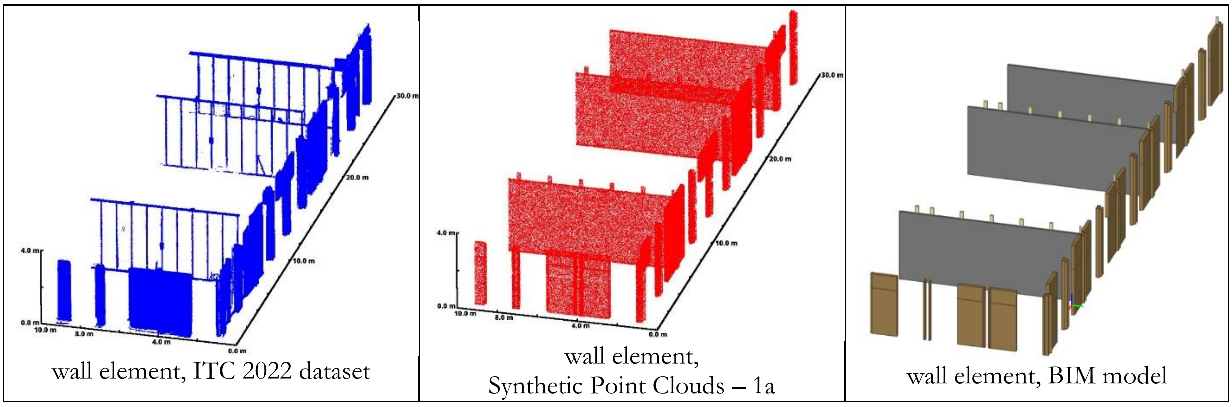


Figure 7.5. Wall Element in the ITC 2022 dataset, the Synthetic Point Clouds – 1a, and the BIM model
(Source: Author)

7.2. Experiment 1 – Comparing the Synthetic Point Clouds and the S3DIS Dataset

This Section presents a comparative analysis of the network trained with the synthetic point clouds and those trained with the S3DIS dataset (Armeni et al., 2016). As previewed in Section 6.1.3, most synthetic point clouds outperformed the S3DIS dataset, particularly the Synthetic Point Clouds – 2c with a 14.22% m-IoU difference. Table 17 provides a comprehensive comparison between these datasets for each element. It highlights that the classification result for each element in the Synthetic Point Clouds – 2c also outperformed the S3DIS dataset, especially for beam and column elements, with 50.15% and 26.17% IoU differences, respectively. However, even though the Synthetic Point Clouds – 2c has a higher quantity of point clouds for the stair element, the S3DIS dataset still has superior classification results, with a 51.14% IoU difference.

Table 19. The Quantity and the Results of Network Trained with S3DIS Dataset and the Synthetic Point Clouds – 2c

Dataset	**Evaluation**	**Building Element**								
		Beam	**Column**	**Door**	**Railing**	**Ceiling**	**Floor**	**Stair**	**Wall**	**Window**
S3DIS	Quantity	4,742,256	5,528,480	13,065,914	0	52,712,823	45,207,796	598,622	75,941,217	6,891,880
	Quantity (%)	2.32	2.70	6.38	0.00	25.75	22.09	0.29	37.10	3.37
	IoU (%)	17.80	4.28	0.00	0.00	72.88	92.80	76.07	17.77	16.93
	m-IoU (%)	37.31								
Synthetic Point Clouds – 2c	Quantity	9,818,542	2,874,775	16,501,670	962,921	8,712,074	52,394,079	742,708	102,187,077	21,775,139
	Quantity (%)	4.55	1.33	7.64	0.45	4.03	24.26	0.34	47.32	10.08
	IoU (%)	67.95	30.45	0.00	52.25	84.38	92.62	24.93	36.94	22.80
	m-IoU (%)	51.53								

As described in Section 1.3, the low performance of the S3DIS dataset happens as a consequence of the network's inability to classify unfamiliar elements, especially beam and column elements. Figure 7.6 and Figure 7.7 compares the appearance of beam and column elements in the S3DIS dataset, the Synthetic Point Clouds – 2c, and the ITC 2022 dataset. Unlike the Synthetic Point Clouds – 2c, those derived from the S3DIS dataset have distinct shapes from the ITC 2022 dataset. Beam elements in the S3DIS datasets are not continuous, with huge gaps that split the elements. Then, column elements in the S3DIS datasets have wider widths and have incomplete point clouds at the bottom. These gaps can be derived from the occlusion effect, where the furniture elements block the column from the sensor system.

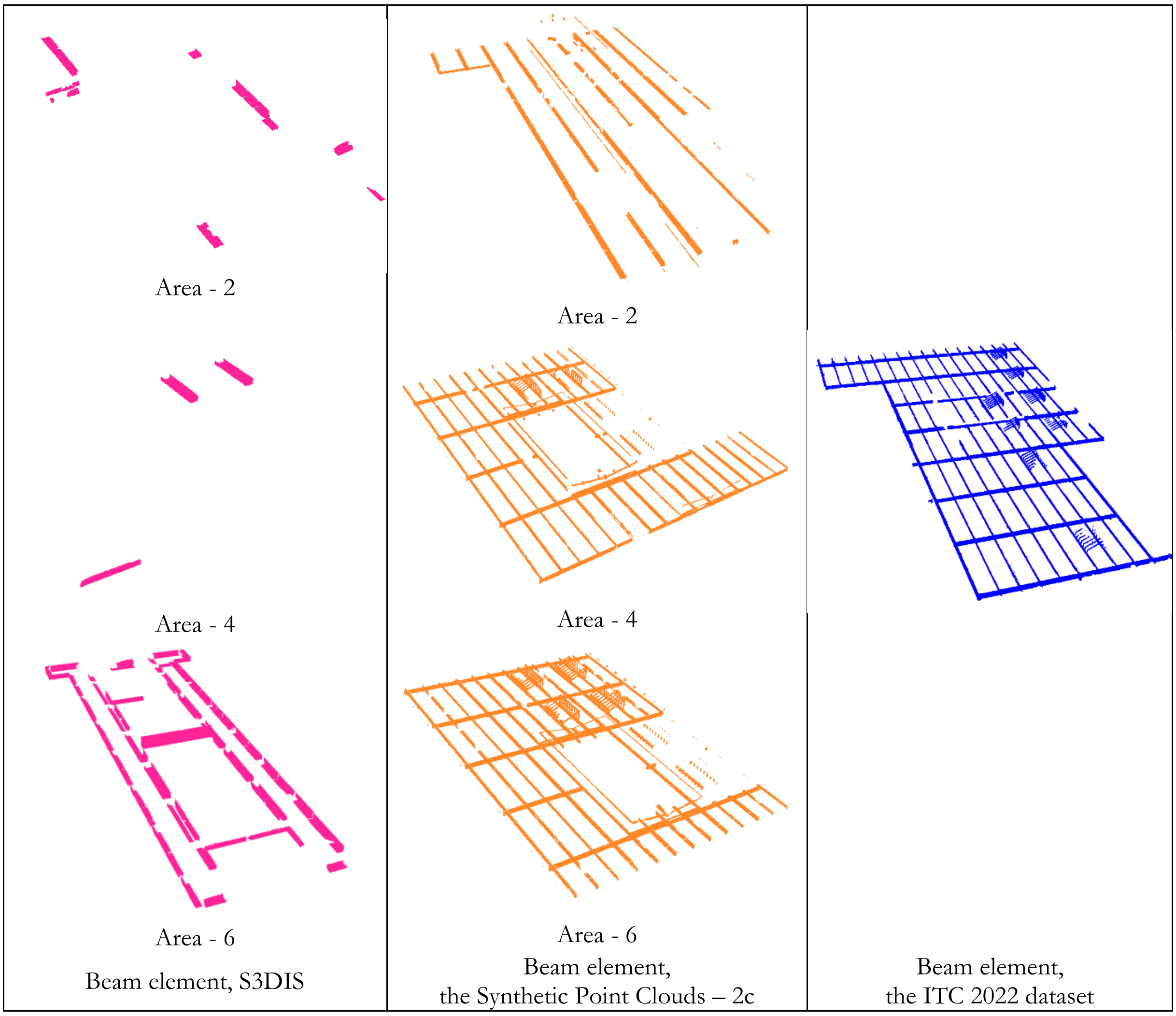


Figure 7.6. Beam Element in the S3DIS Dataset, Synthetic Point Clouds – 2c, and the ITC 2022 dataset (Source: Author)

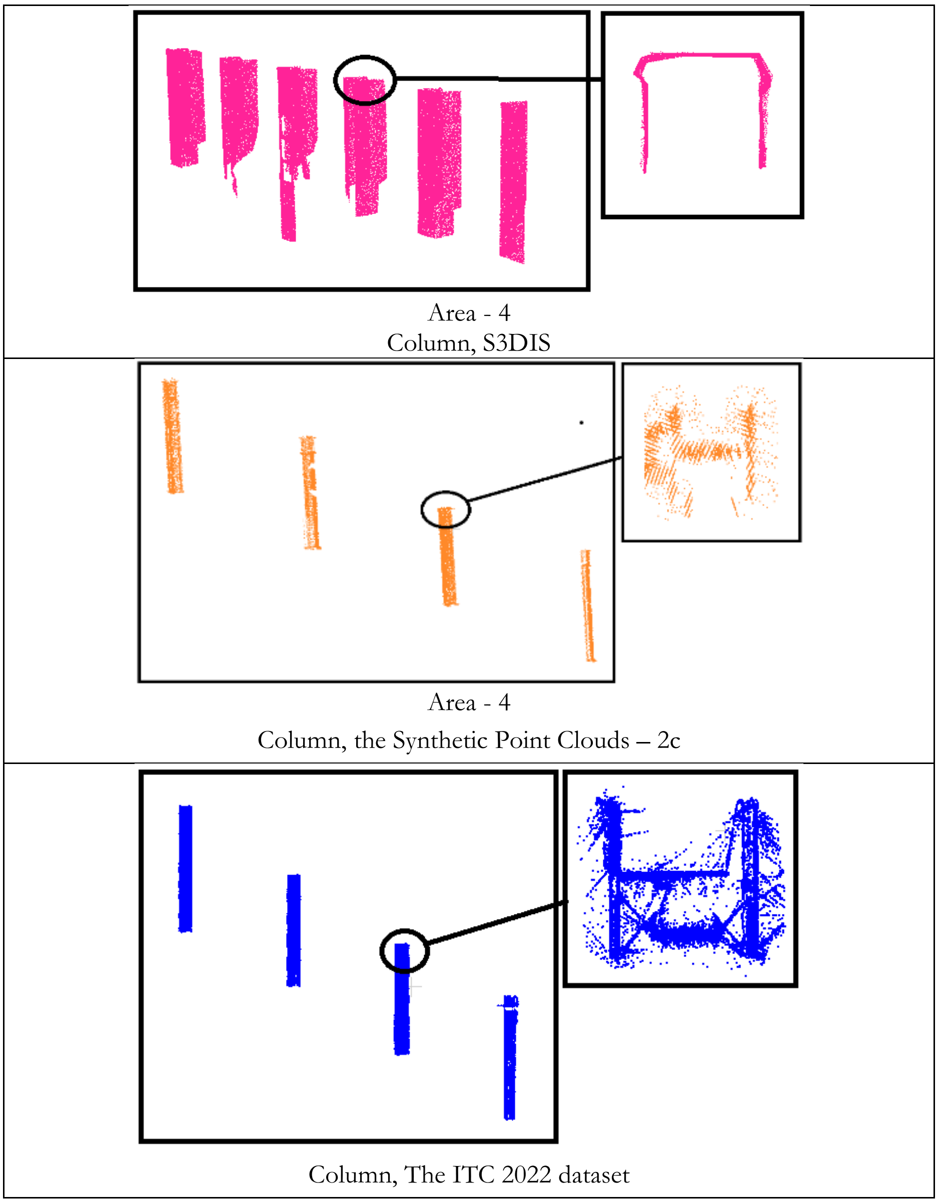
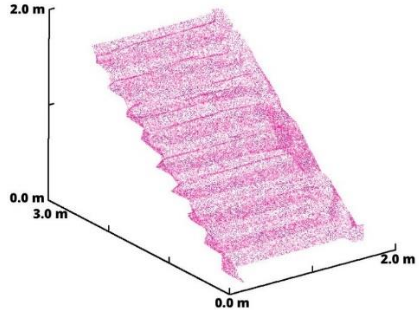
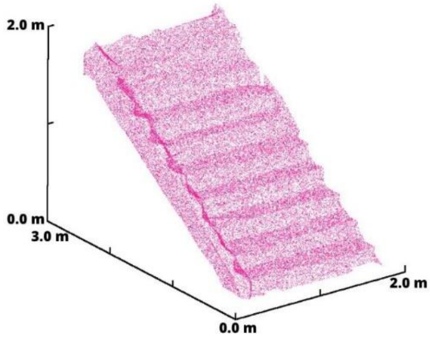


Figure 7.7. Column Element in the S3DIS Dataset, the Synthetic Point Clouds – 2c, and the ITC 2022 dataset (Source: Author)

The reason behind the higher stair classification performance accomplished by the S3DIS dataset can be derived from the difference in the density of point clouds. Figure 7.8 illustrates that the S3DIS dataset provides more quantity for a single stair than the Synthetic Point Clouds – 2c. In addition, the shape of the stair element in the S3DIS dataset does not differ from the ITC 2022 dataset.

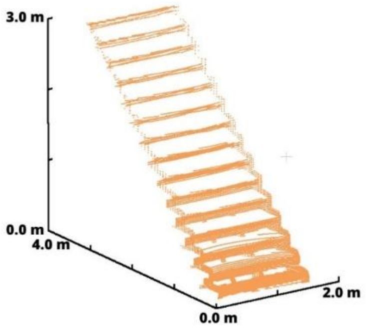


Quantity = 69,389 points

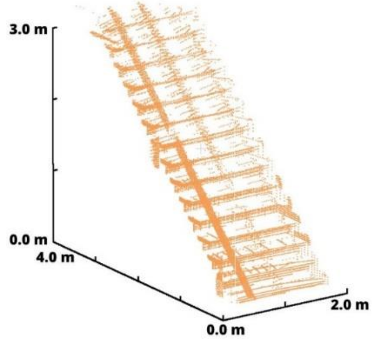


Quantity = 75,905 points

Stair, S3DIS

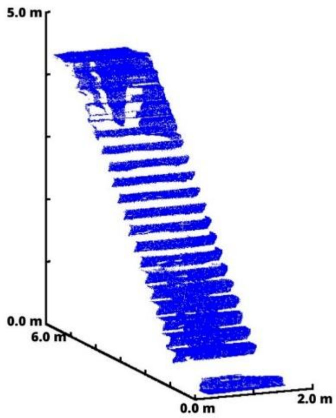


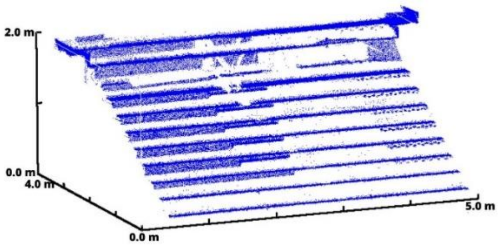
Quantity = 17,078 points



Quantity = 15,267 points

Stair, the Synthetic Point Clouds – 2c





Quantity = 80,275 points

Quantity = 97,809 points

Stair, the ITC 2022 dataset

Figure 7.8. The Quantity for Certain Stairs in the S3DIS Dataset, Synthetic Point Clouds – 2c, and the ITC 2022 dataset (Source: Author)

7.3. Experiment 2 – Comparing the Ideal and the Simulated Method of Synthetic Point Clouds Generation

This Section compares the classification performance of the network trained with two different datasets: 1) the Synthetic Point Clouds – 1a generated from the Ideal Method that does not have the occlusion effect and has random local point cloud distribution, and 2) the Synthetic Point Clouds – 2a generated from the Simulated Method that simulates the occlusion effect and the local point cloud distribution from the real point clouds.

Table 18 shows Synthetic Point Clouds – 2a outperforms Synthetic Point Clouds – 1 with 17.41% m-IoU differences. Figure 7.9 shows the front and side views of the local point cloud distributions obtained from three datasets: the ITC 2022 dataset, the Synthetic Point Clouds – 1a, and the Synthetic Point Clouds – 2a. These point clouds are derived from the flat surface of the wall element, all at the same location in the BIM model. These figures reveal that the point clouds in the ITC 2022 dataset primarily occupy the element surfaces, with a minor proportion positioned above the element surfaces. Particularly, the point clouds on the element surfaces exhibit a uniform distribution, while those above the element surfaces are randomly distributed.

In contrast, the Synthetic Point Clouds – 1a and the Synthetic Point Clouds – 2a only have point clouds on the element surfaces, and none are located above the element surfaces. Nevertheless, the Synthetic Point Clouds – 2a closely resemble the ITC 2022 dataset in terms of the distribution on the element surfaces, exhibiting a uniform distribution. Contrarily, the Synthetic Point Clouds – 1 exhibit a random distribution. As a result, it can be concluded that the high performance of the Synthetic Point Clouds – 2a is due to its resemblance with the local point cloud distribution of the real point cloud, especially those that exhibit on the element surfaces.

Table 20. the Results of Network Trained with the Synthetic Point Clouds – 1a and the Synthetic Point Clouds – 2a

Synthetic Point Clouds Dataset	Evaluation									
	m-IoU (%)	IoU for each Building Element (%)								
		Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
– 1a	20.94	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
– 2a	38.35	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24

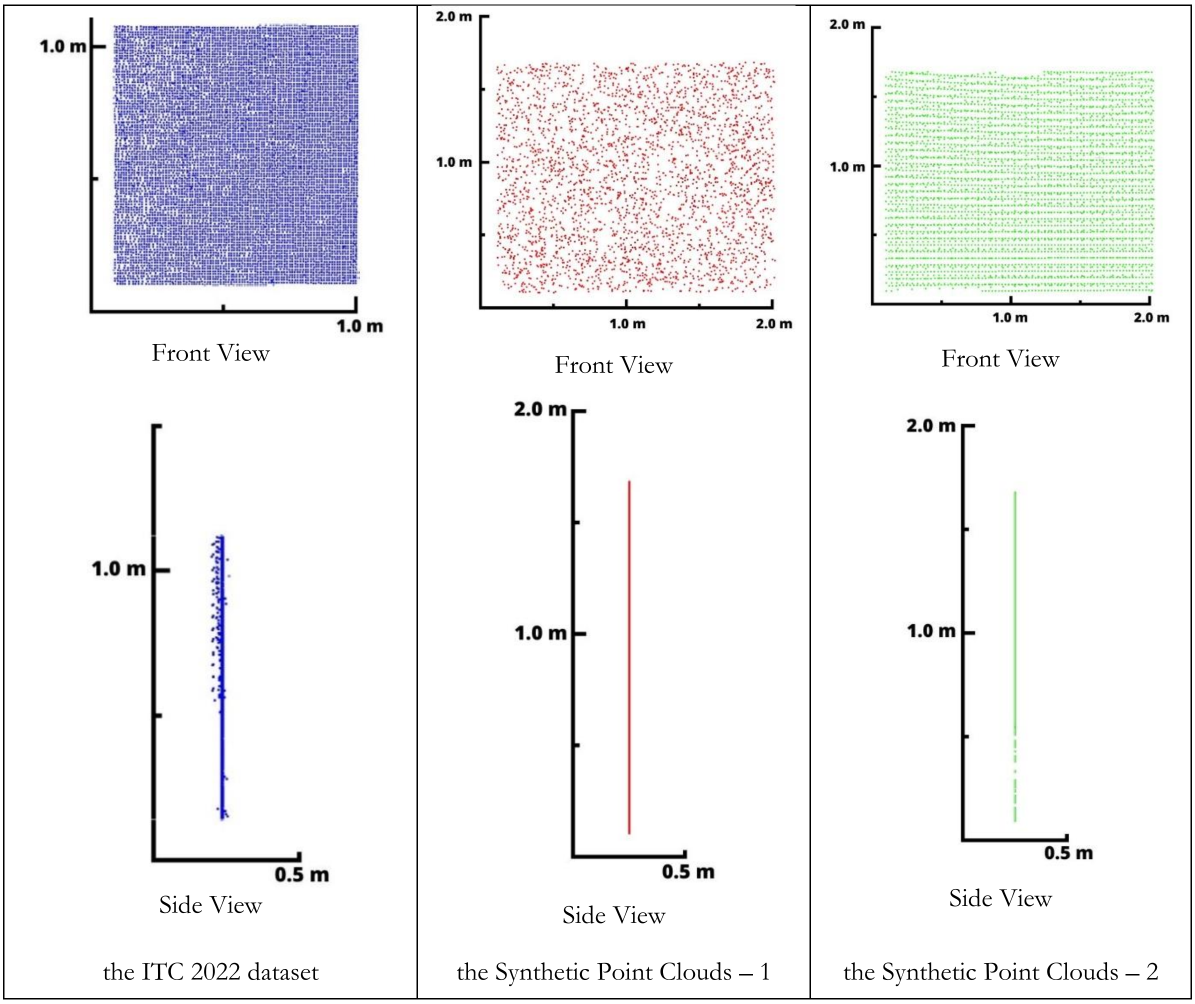


Figure 7.9. Point Cloud Distribution in the ITC 2022 dataset, the Synthetic Point Clouds – 1a and the Synthetic Point Clouds – 2a (Source: Author)

Even though Synthetic Point Clouds – 1a has lower overall network performance, it has higher classification results for the beam and column element (marked in red) than Synthetic Point Clouds – 2a. Figure 7.10 shows that the column element has higher noise than the ceiling element. Therefore, it can be assumed that the network trained on the Synthetic Point Clouds – 1a can generalize well with higher noise-level elements. The reason is that the random local point cloud distribution on the element surfaces from the Synthetic Point Clouds – 1a can simulate the local point cloud distribution of the real point cloud, especially those positioned above the element surfaces.

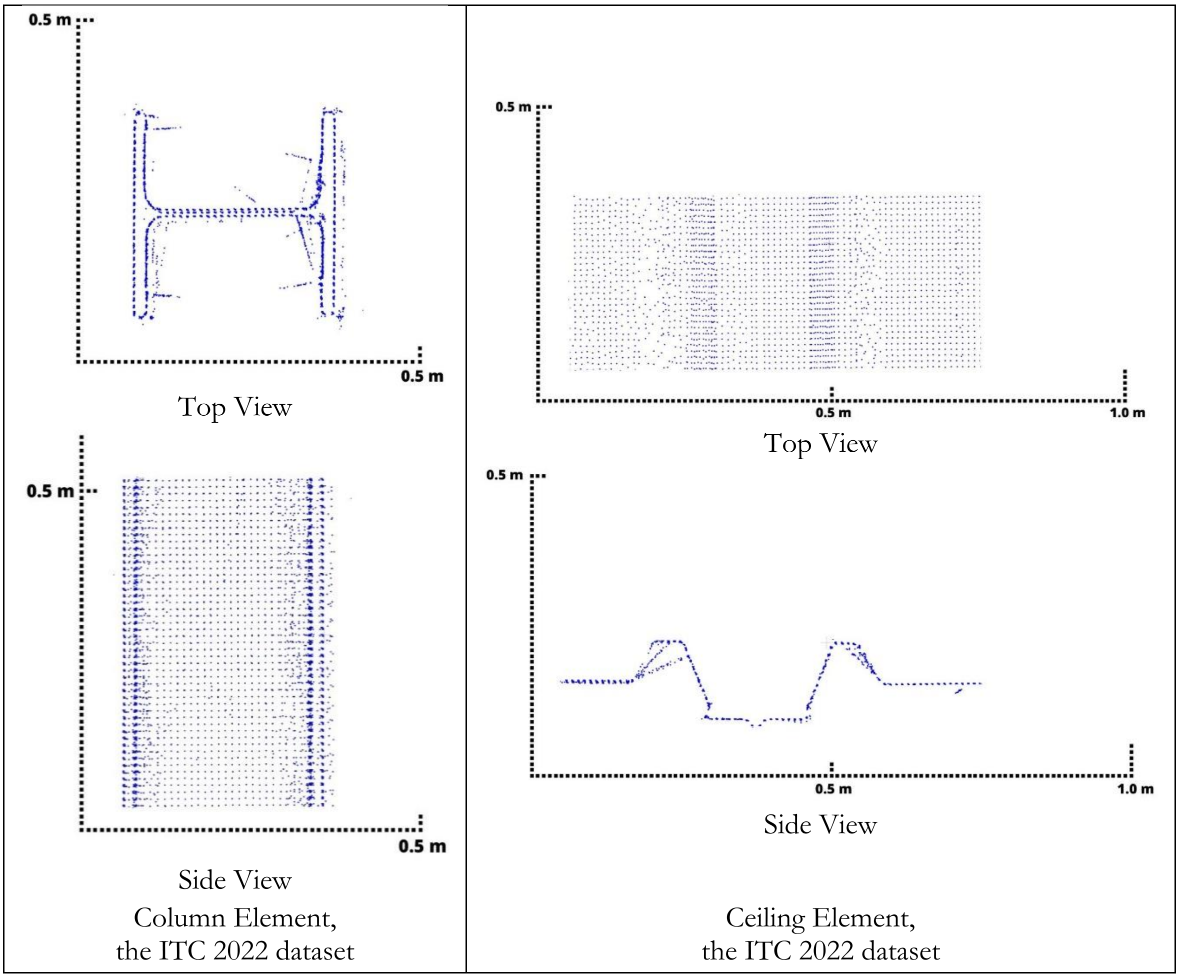


Figure 7.10. Local Point Cloud Distribution for Column and Ceiling Elements in the ITC 2022 dataset (Source: Author)

7.4. Experiment 3 – Comparing the Synthetic Point Clouds on Varying Levels of Sensor System Noise

This Section presents a comparative analysis of the classification performance of the network trained on three different point clouds with varying noise levels derived from the Ideal and Simulated methods.

According to Table 21 and Figure 7.11, increasing the noise to a certain level enhances the network's overall performance. Despite that, increasing the noise beyond that level can also reduce the network's overall performance. For the Ideal method, the Synthetic Point Clouds – 1e, with 0.05 m noise level, achieves the highest result with 39.98% m-IoU. Then, for the Simulated method, the Synthetic Point Clouds – 2c, with 0.01 m noise level, achieves the highest result with 51.54% m-IoU.

Table 21. The Results of the Network Trained with the Synthetic Point Clouds generated from the Ideal method and the Simulated method

Ideal Method Dataset	Noise (m)	m-IoU (%)
Synthetic Point Clouds – 1a	0	20.94
Synthetic Point Clouds – 1b	0.005	23.78
Synthetic Point Clouds – 1c	0.01	29.16
Synthetic Point Clouds – 1d	0.03	35.62
Synthetic Point Clouds – 1e	0.05	39.98
Synthetic Point Clouds – 1f	0.1	36.56
Synthetic Point Clouds – 1g	0.3	12.93

Simulated Method Dataset	Noise (m)	m-IoU (%)
Synthetic Point Clouds – 2a	0	38.35
Synthetic Point Clouds – 2b	0.005	40.47
Synthetic Point Clouds – 2c	0.01	51.54
Synthetic Point Clouds – 2d	0.03	44.91
Synthetic Point Clouds – 2e	0.05	33.82
Synthetic Point Clouds – 2f	0.1	28.10
Synthetic Point Clouds – 2g	0.3	24.45

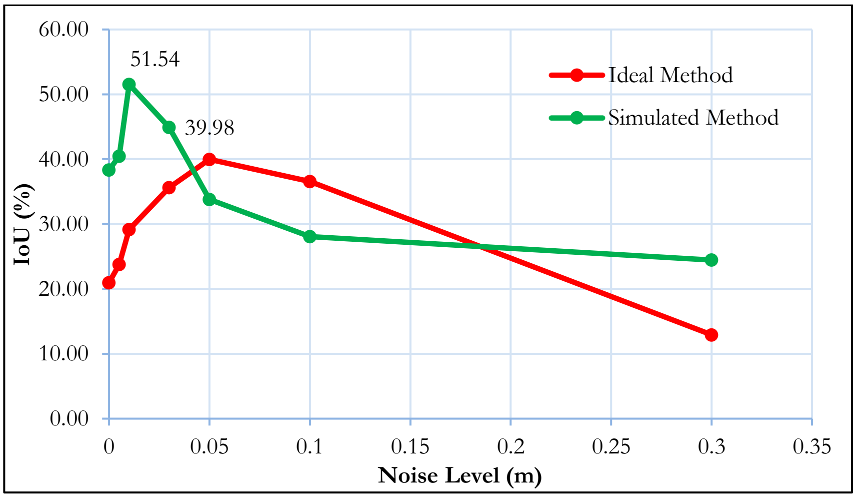


Figure 7.11. The Graph between the Noise Level and the Network's Overall Performance for the Ideal and Simulated Method (Source: Author)

The superior classification results of the Synthetic Point Clouds – 2c can be attributed to a greater resemblance to the local point cloud distribution of the ITC 2022 dataset. Figure 7.12 shows the front and side view for the point cloud distribution of these datasets and the ITC 2022 dataset. They are all derived from the flat surface of the wall element located in the same location in the BIM model. From the front view, up until 0.01 m noise level, these datasets' local point cloud distribution exhibits no significant differences, demonstrating a uniform distribution that resembles the ITC 2022 dataset. However, beyond the 0.01 m noise level, the local point cloud distribution starts to randomize, differing from the ITC 2022 dataset. Then, from the side view, these datasets' local point cloud distribution is different, especially those positioned above the element surfaces. The Synthetic Point Clouds – 2c have approximately 50.0% of point clouds above the element surfaces, aligning more closely with the local point cloud distribution observed in the ITC 2022 dataset. It has a length of 0.03 m from the farthest point cloud to the element surfaces.

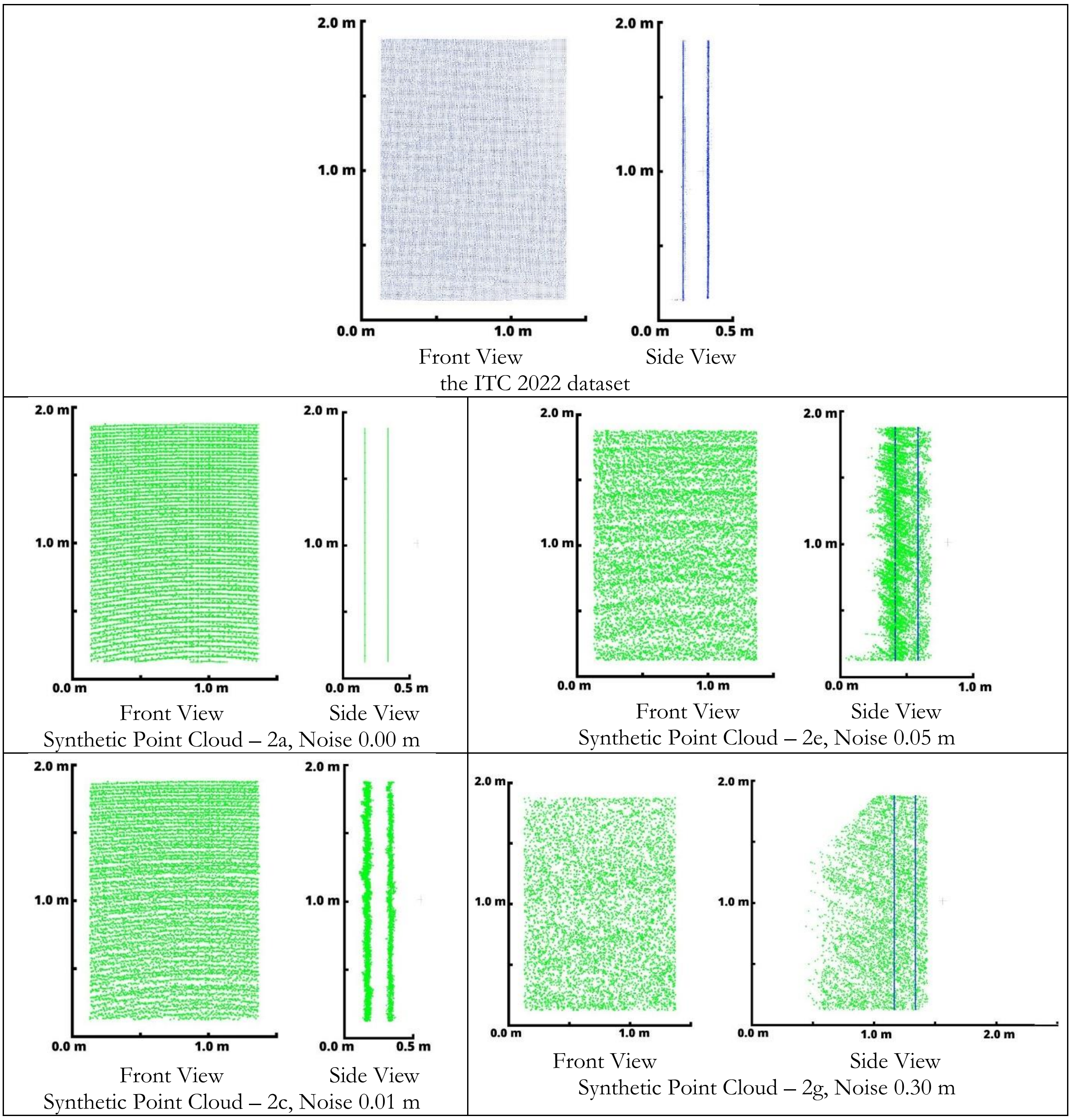


Figure 7.12. The Front and the Side View of the Wall Element in the Synthetic Point Clouds Generated from the Simulated Method (Source: Author)

The high classification results achieved by Synthetic Point Clouds – 2c are proven by its ability to classify point clouds with high noise levels effectively. Figure 7.13 presents both the oblique and side views of the network's predictions using each point cloud, with the point clouds from the ITC 2022 dataset represented in blue. These Figures are obtained from the flat surface of the beam element located in the same location within the BIM model. It is evident from the figures that Synthetic Point Clouds – 2a fails to classify point clouds with noise. Then, the Synthetic Point Clouds – 2b correctly classify the point clouds with small noise. After that, the Synthetic Point Clouds – 2c successfully classify all point clouds affected by noise. Consequently, it can be concluded that the network exhibits higher classification performance when utilizing synthetic point clouds with noise levels akin to the configurations found in Terrestrial Laser Scanning.

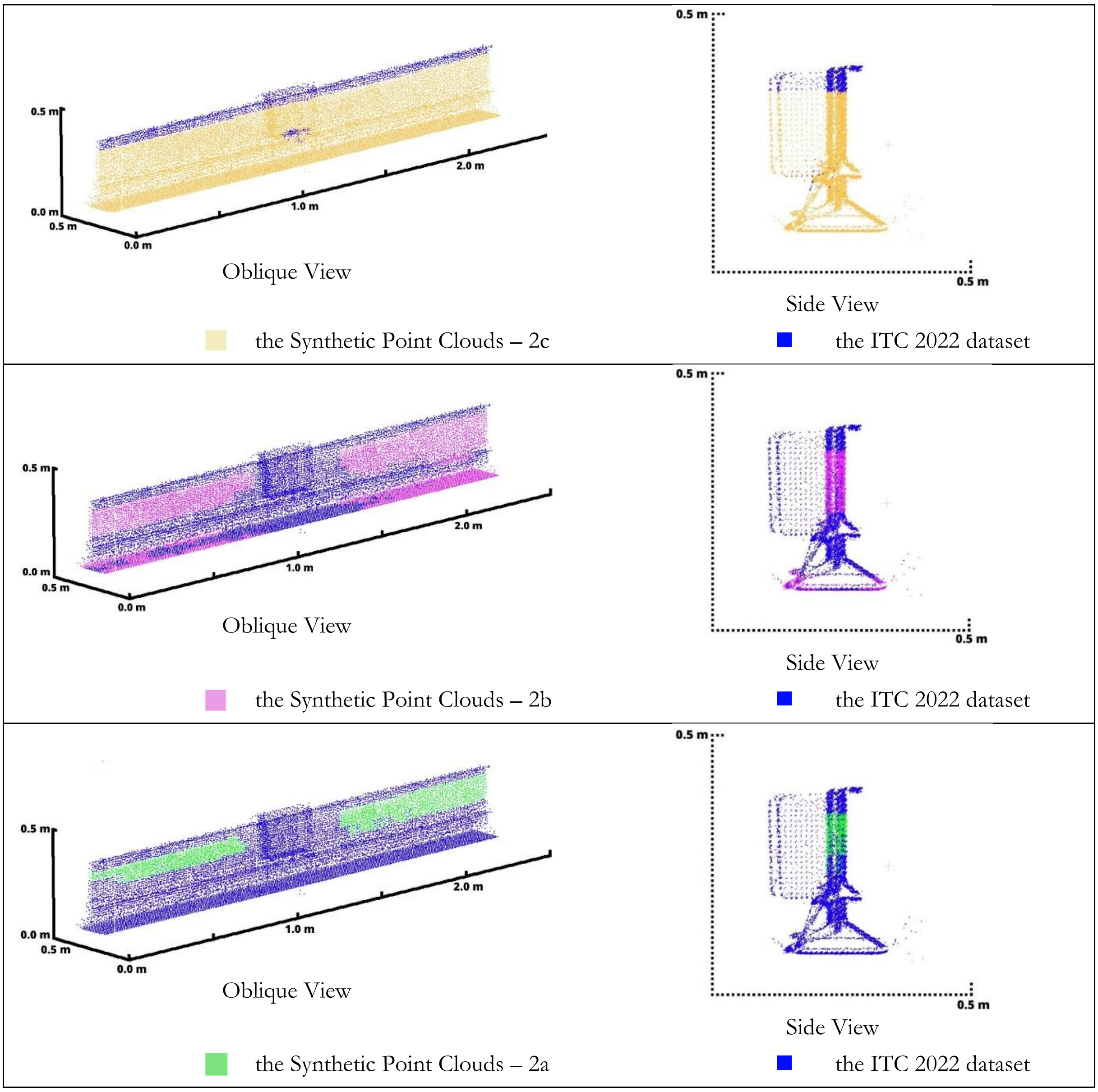


Figure 7.13. The Prediction for Beam Element in the Synthetic Point Clouds – 2c, Synthetic Point Clouds – 2b, and Synthetic Point Clouds – 2a (Source: Author)

7.5. Experiment 4 – Comparing the Synthetic Point Clouds that Consider the Glass as Transparent and Non-Transparent

This Section compares the classification performance of the network trained on the synthetic point clouds that consider the glass a non-transparent object and the synthetic point clouds that consider it a transparent object.

Window elements in the New ITC Building consist of transparent glass and a non-transparent frame. Figure 7.14 displays the prediction for the window element using the Synthetic Point Clouds – 2a and the Synthetic Point Clouds – 2h. It can be seen that the network trained on the Synthetic Point Clouds – 2a classifies the glass and the frame part of the window. Contrarily, the network trained on the Synthetic Point Clouds – 2h classifies only the frame part of the window.

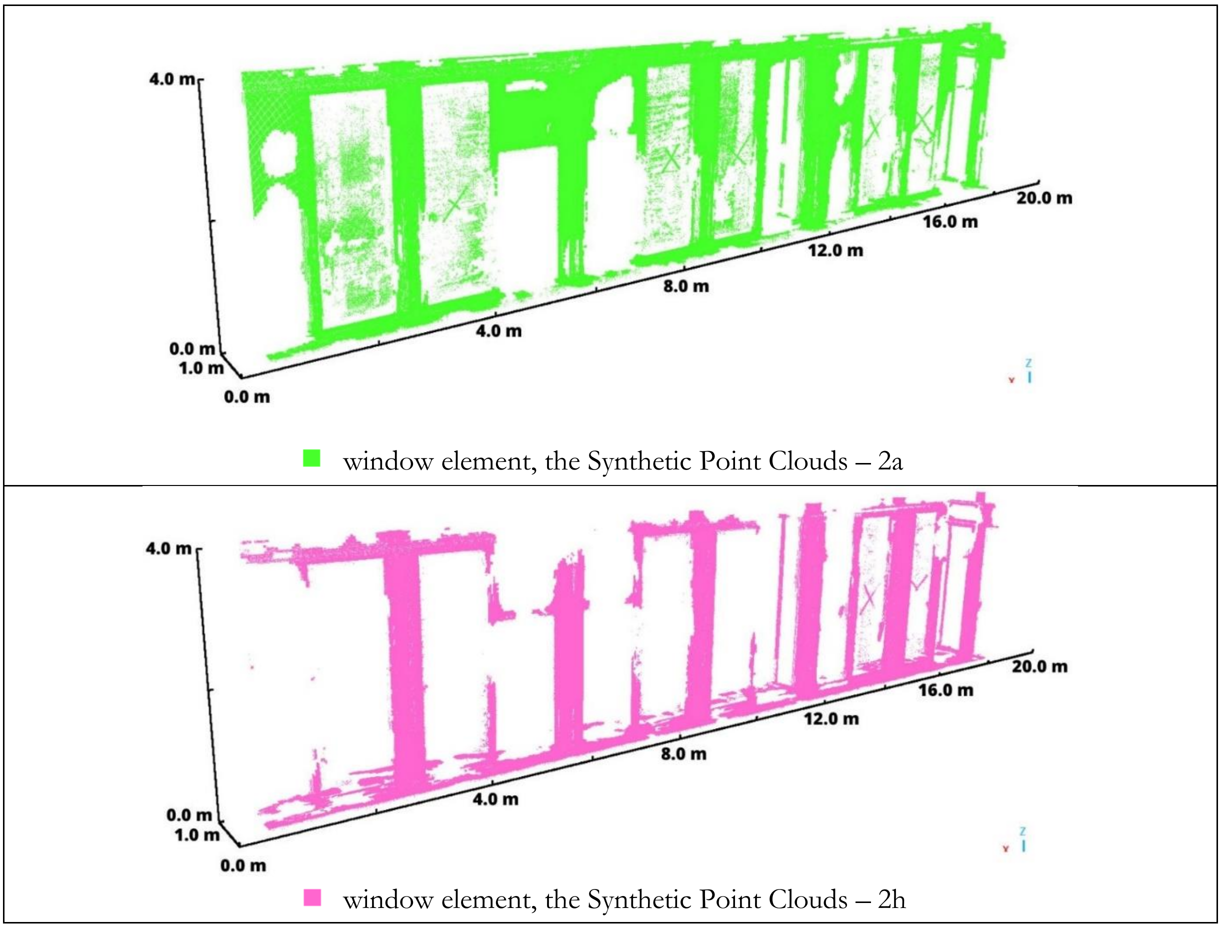


Figure 7.14. The Prediction of Window Element using Synthetic Point Clouds – 2a and Synthetic Point Clouds – 2h (Source: Author)

Table 20 shows that the beam and wall element classification result is increased when using the synthetic point clouds that consider glass as a transparent object. The reason is that removing the glass part from the window in the synthetic point clouds can reduce the similarity between the window with the beam and wall elements. As a result, this condition reduces the influence of the inter-class similarity problem described in Section 7.1.2.

Despite having different classification results for beam and wall elements, both datasets have similar m-IoU values as the overall classification performance. Additionally, the classification result of the window element using the Synthetic Point Clouds – 2h is lower than that of Synthetic Point Clouds – 2a. The reason is that the ITC 2022 dataset used to test the network still has point clouds for the glass part of the window element, which is classified as a window. Then, the network can not learn the features of the glass part of the window element since the Synthetic Point Clouds – 2h do not provide any. As a result, the point clouds located at the glass part are misclassified as door and wall elements, decreasing the classification result for window elements.

Table 22. The Results of the Network Trained with the Synthetic Point Clouds that consider the Glass Object as Transparent and Non-transparent

Synthetic Point Clouds Dataset	Transparent Glass	Evaluation									
		m-IoU (%)	IoU for each Building Element (%)								
			Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
– 1a	**No**	20.94	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
– 1h	**Yes**	25.434	40.78	11.92	0.00	23.36	41.97	41.81	6.65	31.95	5.03
– 2a	**No**	38.35	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24
– 2h	**Yes**	38.50	35.26	15.49	0.00	60.09	84.39	54.01	13.76	36.39	8.60

7.6. Experiment 5 – Augmenting the Synthetic Point Clouds and the S3DIS Dataset

This Section compares the classification performance of the network trained on three different datasets: the Synthetic Point Clouds – 2c, the S3DIS datasets, and the augmentation of Synthetic Point Clouds – 2c and S3DIS datasets.

As seen in Table 23Table 23. The Results of Network Trained with the Synthetic Point Clouds – 2c, the S3DIS dataset,, the network's overall performance has a 12.3% m-IoU increase by augmenting Synthetic Point Clouds – 2c and S3DIS datasets, compared to only S3DIS datasets. Despite that, not all elements experienced an enhancement in classification results. Only window and railing elements have higher classification results, while column and stair elements have reduced classification results. It assumed that the network can not learn different variations of column and stair elements from Synthetic Point Clouds – 2c and S3DIS datasets, resulting in lower classification results.

Table 23. The Results of Network Trained with the Synthetic Point Clouds – 2c, the S3DIS dataset, and the Combination of the Synthetic Point Clouds – 2c and the S3DIS dataset

Dataset	**Evaluation**									
	m-IoU (%)	**IoU for each Building Element (%)**								
		Beam	**Column**	**Door**	**Railing**	**Ceiling**	**Floor**	**Stair**	**Wall**	**Window**
Synthetic Point Clouds – 2c	51.54	67.95	30.45	0.00	52.25	84.38	92.62	24.93	36.94	22.80
S3DIS	37.32	17.80	4.28	0.00	0.00	72.88	92.80	76.07	17.77	16.93
Synthetic Point Clouds – 2c + S3DIS	55.01	69.09	24.97	0.00	64.71	86.28	92.88	32.96	38.14	31.06

7.7. Experiment 6 – Applying the Proposed Approach on Different Construction Stages

This Section compares the classification performance of the networks tested on the ITC 2021 and ITC 2022 datasets.

Based on Table 24, four similarities are found in the results of the networks tested on the ITC 2021 and ITC 2022 datasets. First, mirroring the findings revealed in Section 7.3, the Synthetic Point Clouds – 2a has better overall network performance than the Synthetic Point Clouds – 1a, proving that Simulated methods generate the synthetic point clouds with more resemble than the Ideal method. Second, similar to findings in Section 7.4, the classification results for each element are enhanced by increasing the noise level as the Synthetic Point Clouds – 2c has better results than the Synthetic Point Clouds – 2a. Third, parallels with the conclusions explained in Section 7.5, the classification result for the window element is lower in the Synthetic Point Clouds – 2h compared to the Synthetic Point Clouds – 2a. Fourth, reminiscent of the observations described in Section 7.2, the Synthetic Point Clouds – 2c has better overall network performance than the S3DIS dataset. It proved that the BIM model can help provide the synthetic point clouds that are more relevant to the point cloud classification for this building than the S3DIS dataset.

Despite that, unlike the networks tested in the ITC 2022 dataset, the network tested on the ITC 2021 dataset has higher classification results for beam, column, and wall elements. As mentioned in Section 7.1.2, the window on the ITC 2022 dataset has multiple variations that lead to inter-class similarity problems and reduces the discriminative power of the network. Nevertheless, the ITC 2021 dataset only consists of a single window variation located on the ceiling, as seen in Figure 7.16. Additionally, they have distinct shapes with the beam, column, and wall elements, avoiding inter-class similarity problems.

Additionally, the network tested on the ITC 2021 dataset has poor railing and window elements results. The low classification result in the railing element is primarily because of the significant disparity of the railing element in the training and test datasets. Figure 7.15 illustrates that the railing element in the ITC 2021 dataset is more complex than in synthetic point clouds. As a result, the network failed to identify the railing element in the ITC 2021 dataset. After that, the low classification result in the window element is due to the limited quantity of training datasets. As previously mentioned, the ITC 2021 dataset only consists of a single window variation on the ceiling. However, this window variation is only represented by a few point clouds. For example, in the synthetic point clouds – 2a, 591,387 out of 23,605,059 points, or 2.5% from the window element in the synthetic point clouds. As a result, this window variation is under-represented, and the network can not capture enough features to identify it in the test dataset.

Therefore, it can be concluded that utilizing the proposed approach in different building construction stages may provide different classification results.

Table 24. The Results of the Network Tested on the ITC 2021 and ITC 2022 datasets

Dataset	Test Dataset	Evaluation									
		m-IoU (%)	IoU for each Building Element (%)								
			Beam	Column	Door	Railing	Ceiling	Floor	Stair	Wall	Window
Synthetic Point Clouds – 1a	ITC 2022	20.94	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
	ITC 2021	42.39	49.33	86.71	0.00	0.23	60.56	99.19	0.00	0.35	0.35
Synthetic Point Clouds – 2a	ITC 2022	38.35	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24
	ITC 2021	56.79	85.24	69.29	0.00	0.31	90.86	98.20	0.00	51.95	1.71
Synthetic Point Clouds – 2c	ITC 2022	51.54	67.95	30.45	0.00	52.25	84.38	92.62	24.93	36.94	22.80
	ITC 2021	61.39	86.88	87.07	0.00	7.14	92.18	98.29	0.00	56.54	1.66
Synthetic Point Clouds – 2h	ITC 2022	38.50	35.26	15.49	0.00	60.09	84.39	54.01	13.76	36.39	8.60
	ITC 2021	59.82	84.89	75.52	0.00	0.83	92.25	98.94	0.00	65.24	1.04
S3DIS	ITC 2022	37.32	17.80	4.28	0.00	0.00	72.88	92.80	76.07	17.77	16.93
	ITC 2021	33.89	31.16	1.36	0.00	0.00	77.47	98.78	0.00	28.48	0.00

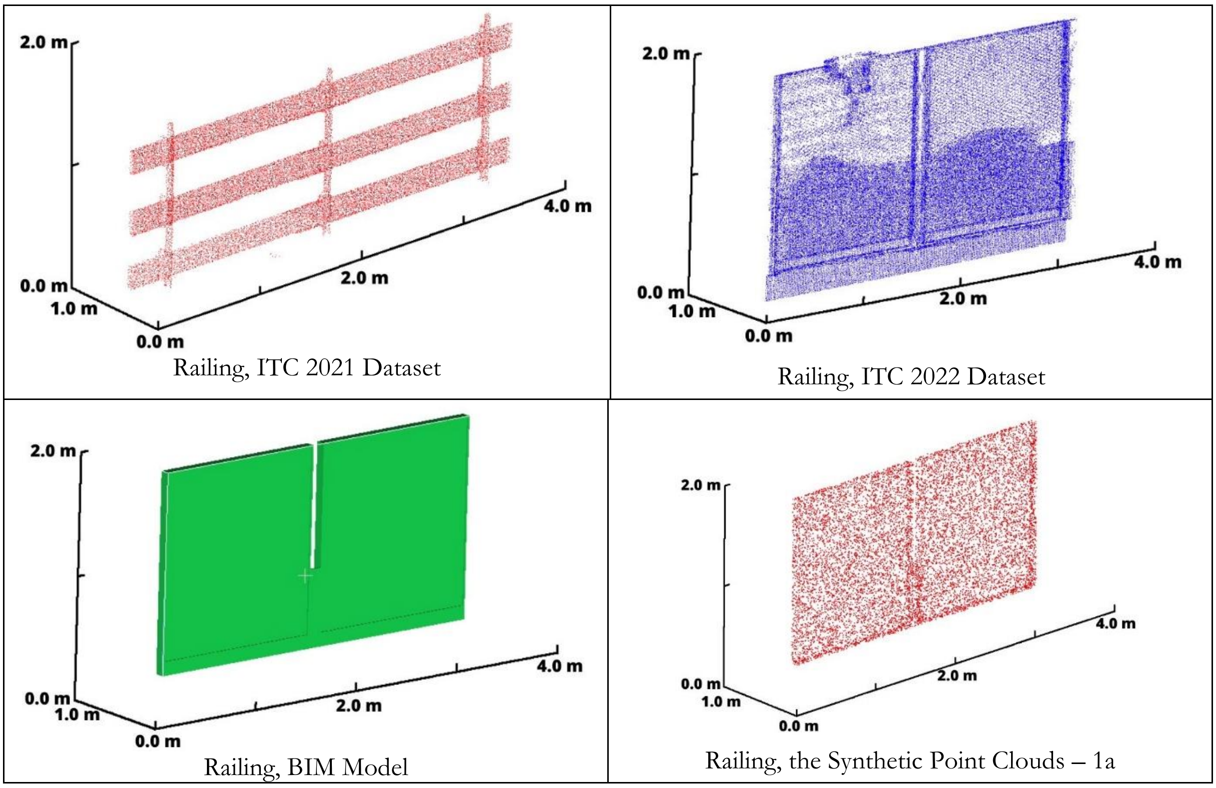


Figure 7.15. Railing Element in the ITC 2021 dataset, the ITC 2022 dataset, the BIM model, and the Synthetic Point Clouds – 1a (Source: Author)

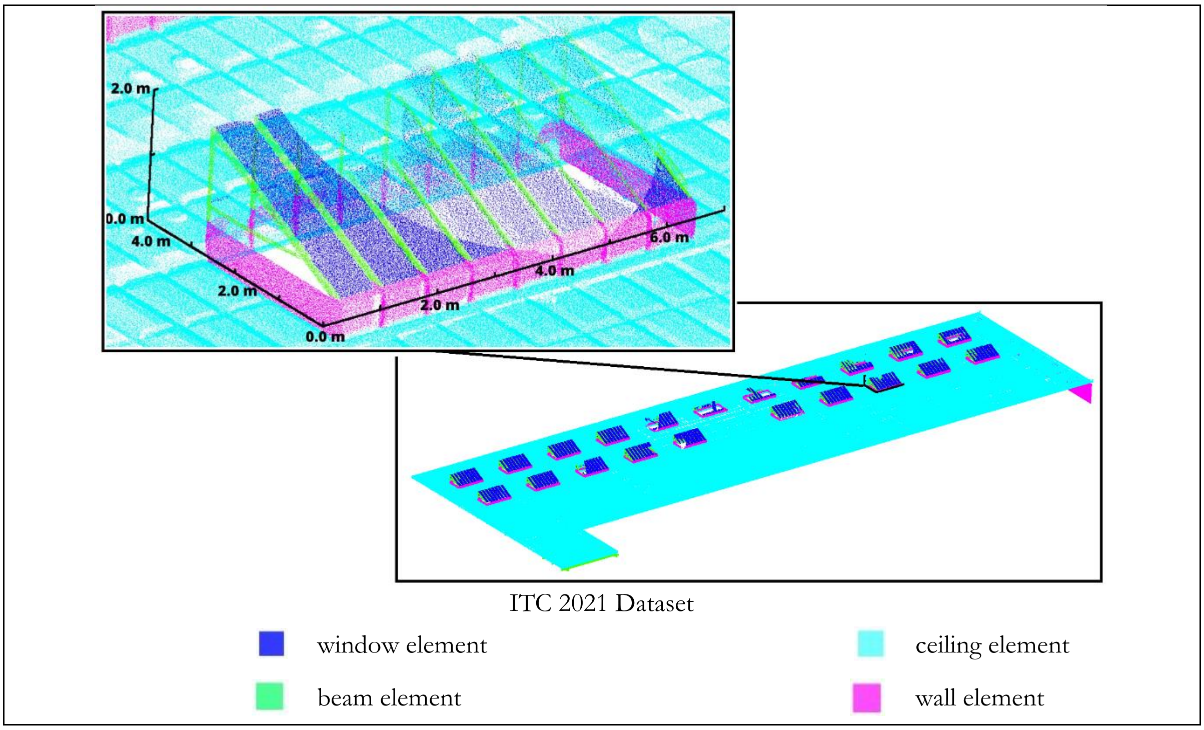


Figure 7.16. Window Element in the ITC 2021 Dataset (Source: Author)

7.8. The Uses of KP-FCNN

This section explores the utilization of KP-FCNN deep learning networks in this research. First, the classification performance of the networks trained using Network Parameter – 1 and those trained using

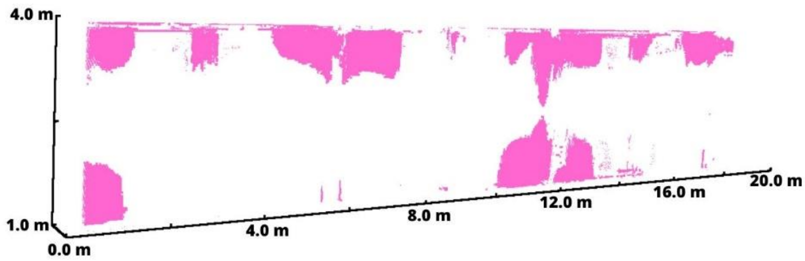
Network Parameter – 2 are compared. Then, utilizing a large voxel size for grid subsampling is also explored.

As explained in Section 2.3, Network Parameter – 2 has a random picking strategy that arbitrarily samples the input point clouds and samples the same number for each class. It makes the network loss during the network training has balanced influences from each element, resulting in higher classification performance. Despite that, as seen in Table 25, there are no significant differences in the overall classification performance between the networks that used these parameters. It assumed that the inter-class similarities problems described in Section 7.1.2 conflict with the influences of the random picking method. Hence, the random picking method fails to increase the influence of the minority elements.

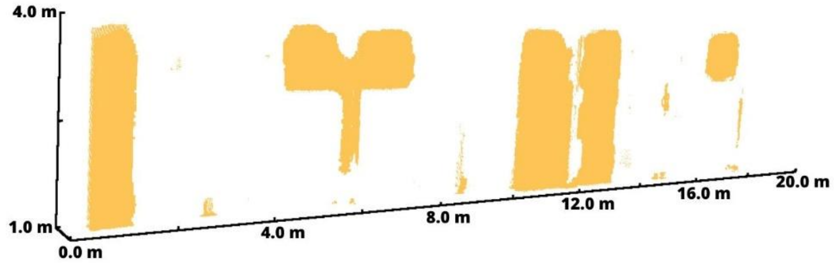
Additionally, as mentioned in Section 7.1.2, wall and window elements share similar shapes that confuse the network and lead to poor classification performance for the S3DIS dataset. Nevertheless, utilizing a large voxel size for grid subsampling in KP-FCNN, executed in Section 3.6.2, can make the network learn the position distribution of wall and window elements. As a result, the Synthetic Point Clouds – 2c has higher classification performance than the S3DIS dataset, especially for wall elements with 19.17% IoU differences.

Table 25. The Results of Network Trained using Network Parameter – 1 and Network Parameter – 2

Dataset	**Network Parameter**	**Evaluation**									
		m-IoU (%)	**IoU for each Building Element (%)**								
			Beam	**Column**	**Door**	**Railing**	**Ceiling**	**Floor**	**Stair**	**Wall**	**Window**
Synthetic Point Clouds – 1a	1	20.94	30.69	19.69	0.00	8.07	1.81	62.53	5.95	20.21	18.57
	2	25.73	30.89	8.63	0.00	0.00	26.42	89.98	3.47	38.98	7.49
Synthetic Point Clouds – 2a	1	38.35	23.80	1.51	0.00	42.53	87.44	81.22	24.21	21.87	24.24
	2	36.49	43.16	5.17	0.00	12.13	81.02	83.69	27.53	23.10	16.15
Synthetic Point Clouds – 2c	1	51.54	67.95	30.45	0.00	52.25	84.38	92.62	24.93	36.94	22.80
	2	48.10	66.17	14.82	0.00	28.59	85.35	94.64	40.94	36.41	17.90
Synthetic Point Clouds – 2h	1	38.50	35.26	15.49	0.00	60.09	84.39	54.01	13.76	36.39	8.60
	2	38.39	37.50	1.17	0.00	20.22	87.60	89.11	23.87	30.65	17.00
S3DIS	1	37.32	17.80	4.28	0.00	0.00	72.88	92.80	76.07	17.77	16.93
	2	39.93	37.70	0.17	0.00	0.00	82.33	90.85	63.19	23.56	21.67

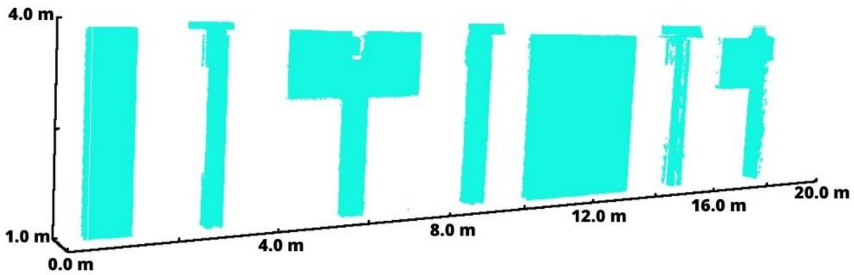


Wall Element Prediction using
S3DIS



Wall Element Prediction using
the Synthetic Point Clouds – 2c

Figure 7.17. The Prediction for the Wall Element using S3DIS Dataset and Synthetic Point Clouds – 2c (Source: Author)



Wall Element in
the ITC 2022 dataset

Figure 7.18. Wall Element in the ITC 2022 dataset (Source: Author)

7.9. Limitations

Below are the limitations of the approach used in this research.

The Inconsistency of the IFC Class in the BIM Model

The BIM model is converted into synthetic point clouds with semantic information based on the IFC class. However, as explained in Section 3.1, the IFC class can not be used directly. For example, although the IFC stair does not include other IFC classes, it contains other elements, including floor and railing. Consequently, it can introduce inter-class similarity problems to the network, causing misclassifications. Therefore, the classification of the IFC class needs to be done as pre-processing.

The Need for the Normalization in the Training Dataset

Then, as explained in Section 3.5, normalization needs to be done to the point clouds before using it to train the deep learning network. The limitation derived from this process is that the classified point clouds, with the acquired semantic information, can not be directly utilized on the target application. Specifically, the normalized point clouds must be returned to their original position. Then, the point clouds separated based on their floor must be combined back. Construction progress monitoring can not be done if the design model and the classified point clouds are in different positions. Therefore, another post-processing procedure is needed.

Coordinate system transformation or registration methods can be utilized for this problem. The transformation parameters can be defined by comparing the normalized position with the original position of point clouds. Additionally, before the semantic classification process, adding an index feature to four point clouds can facilitate this process. It assists in the search for the corresponding point clouds in the normalized and the original point clouds.

8. CONCLUSION AND RECOMMENDATION

This chapter concludes all of the research findings in Section 8.1. Then, the research questions mentioned in Section 1.7 are answered in Section 8.2. Lastly, multiple recommendations are described in Section 0 to improve the approach used in this research.

8.1. Conclusion

The main objective of this research is to confirm the effectiveness of of the BIM models for point cloud classification to overcome the problem of limited availability of labeled indoor point cloud datasets. This research focused to classify building elements found in the construction progress monitoring, including beam, ceiling, column, door, floor, railing, stair, wall, and window. The BIM models are converted into labeled synthetic point clouds with relevant shapes of the architectural layouts to the buildings to be predicted. Then, the synthetic point clouds are used to train the deep learning network. The results of this approach are evaluated based on the comparison with the benchmark point clouds dataset publicly available, S3DIS (Armeni et al., 2016).

Leveraging existing BIM models are proven to be helpful for the point cloud classification at indoor scenes. The networks trained on the synthetic point clouds has a better network's overall performance than the S3DIS dataset. In Section 7.2, the Synthetic Point Clouds – 2c, generated from the Simulated method with 0.01 m noise level, has 14.22% mean – Intersection over Union (m-IoU) differences with the S3DIS dataset when tested on the ITC 2022 dataset. Section 7.7 also has similar results when tested on earlier stages of the construction, the ITC 2021 dataset. The Synthetic Point Clouds – 2c has 27.50% m-IoU differences with the S3DIS dataset. Then, based on Section 3.3, this approach does not utilize manual data collection and classification, which makes it an inexpensive and non-subjective process.

There is a case where the S3DIS dataset remains superior compared to the Synthetic Point Clouds. The S3DIS dataset has higher classification results for the stair element by 51.14% IoU since it has higher point cloud density. Therefore, synthetic point cloud generation methods should be configured to have a sufficient density of point clouds.

Additionally, as mentioned in Section 7.1.1 and Section 7.1.2, this research encountered class-imbalance and inter-class similarity problems that degraded the network performance. Small amounts of stair element, with less than one percent of overall point clouds, are dominated by floor and ceiling elements. As a result, it only has 25.48% IoU as the higest classification results. After that, the network struggles to differentiate between beam, wall, and window elements. There are multiple variations of window elements that share similar shapes to others, resulting in low window classification result with 27.97% IoU as the highest result.

8.2. Answer to Research Questions

This section answers the research questions mentioned in Section 1.7

1. **How will the results changed when the same synthetic point clouds are used in different construction stages of the buildings?**
   Utilizing the same synthetic point clouds in different construction stages can have different results. In Section 7.7, when the networks are tested on the ITC 2021 dataset, the classification performance is higher than those tested on the ITC 2022 dataset. For example, the Synthetic Point Clouds – 1a and Synthetic Point Clouds – 2a can have higher results by 21.45% m-IoU and 18.44% m-IoU, respectively. The reason is that, unlike the ITC 2022 dataset, the ITC 2021 dataset does not have multiple variations of window elements, reducing the influence of the inter-class similarity problem described in Section 7.1.2. Contrarily, in Section 7.1.3, the networks completely unidentified certain wall elements. The reason is that this wall element is unfinished, which has distinct shapes from one in the BIM model. Instead, it resembles the window elements. Therefore, synthetic point clouds should provide elements for all conditions.

2. **How can the augmentation of the synthetic point clouds and the S3DIS dataset can improve the classification results?**
   The third hypothesis of this research is proven as the combination of the synthetic point clouds and the real point clouds can increase the classification performance compared to only using a single dataset. In Section 7.6, the classification performance of the networks trained on the Synthetic Point Clouds – 2c and the S3DIS dataset are 51.54% m-IoU and 37.32% m-IoU, respectively. Nevertheless, augmenting the Synthetic Point Clouds – 2c and the S3DIS dataset has 55.01% m-IoU, much higher than using the Synthetic Point Clouds – 2c or the S3DIS dataset alone. The reason is that the network can learn real point cloud characteristics other than local point cloud distribution, occlusion effect, and sensor system noise in the S3DIS dataset, not included in the Synthetic Point Clouds – 2c.

3. **What is the right way to simulate the local point cloud distribution and occlusion effect to help the point cloud classification?**
   Part of the second hypothesis of this research is proven as including the real point cloud characteristics of the local point cloud distribution and occlusion effect in the synthetic point clouds can increase the point cloud classification performance. In Section 7.3, utilizing the Simulated method, the Synthetic Point Clouds – 2a closely resemble the ITC 2022 dataset regarding the distribution on the element surfaces, exhibiting a uniform distribution. As a result, it has a higher classification result by 17.41% m-IoU compared to the Synthetic Point Clouds – 1a,

generated from the Ideal method. Then, unlike the Synthetic Point Clouds – 1a, the Synthetic Point Clouds – 2a has comparable results with the S3DIS dataset. It only has 1.03% m-IoU differences.

Also, in Section 7.4, when the sensor noise is added to the synthetic point clouds, all synthetic point clouds generated from the Simulated method outperform those generated from the Ideal method. The Synthetic Point Clouds – 2c, with the highest classification result from the Simulated method, outperforms the Synthetic Point Clouds – 1e, with the highest classification result from the Ideal method, by 11.56% m-IoU differences. Additionally, in Section 7.7, this condition also occured when the network is tested on earlier stages of the construction, the ITC 2021 dataset. The Synthetic Point Clouds – 2a has 14.40% m-IoU differences with the Synthetic Point Clouds – 1a.

4. **How can including sensor system noise in the synthetic point clouds help the point cloud classification?**

The second hypothesis of this research is also proven as including the real point cloud characteristics of sensor system noise in the synthetic point clouds can increase the classification performance. In Section 7.4, when the sensor system noise is added, the synthetic point clouds closely resemble the ITC 2022 dataset regarding the distribution above the element surfaces, exhibiting a random distribution. As a result, the classification results are enhanced. The Synthetic Point Clouds – 1c, generated from the Ideal method with 0.01 m noise, has a higher classification result by 8.22% m-IoU than the Synthetic Point Clouds – 1a, with 0.00 m noise. Similarly, the Synthetic Point Clouds – 2c, generated from the Simulated method with 0.01 m noise, has higher classification results by 13.19% m-IoU than the Synthetic Point Clouds – 2a, with 0.00 m noise. Additionally, in Section 7.7, this condition also occured when the network is tested on earlier stages of the construction, the ITC 2021 dataset. The Synthetic Point Clouds – 1c and the Synthetic Point Clouds – 2c are higher than those without noise, with 10.57% m-IoU and 4.6% m-IoU differences, respectively.

However, increasing the noise level beyond a certain level can degrade the classification performance. For the Ideal method, this noise level is 0.005 m while the Simulated method is 0.01 m. The reason is that the synthetic point clouds no longer resemble the ITC 2022 dataset. Therefore, the sensor system noise level should be configured when it is added to the synthetic point clouds, where it should be similar to those from the real point clouds it will be predicted.

5. **How can the synthetic point clouds that consider the glass as transparent object help the point cloud classification?**

   The second hypothesis of this research is not proven, as considering glass as a transparent object does not increase the window element's classification result. In Section 7.5, the window element in the Synthetic Point Clouds – 1h, where it considers the glass as a transparent object, has a lower classification result than those in the Synthetic Point Clouds – 1a, where it does not consider the glass as a transparent object. It has 13.54% IoU differences. Similarly, the window element in the Synthetic Point Clouds – 2h also has lower classification results than those in the Synthetic Point Clouds – 2a, with 15.64% IoU differences. The reason is that the ITC 2022 dataset has point clouds at the window element's glass part. Since the synthetic point cloud does not provide any point clouds, the network misclassifies it to door and wall elements. Therefore, the synthetic point clouds should provide the feature for this object to the network.

6. **How robust is the KP-FCNN deep learning network in point cloud classification in indoor scenes?**

   Utilizing a large voxel size for grid subsampling in KP-FCNN, executed in Section 3.6.2, can make the network learn the position distribution of wall and window elements, reducing the influence of the inter-class similarity problem described in Section 7.8. For example, the Synthetic Point Clouds – 2c has higher classification performance than the S3DIS dataset, especially for wall elements with 19.17% IoU differences. However, in Section 7.8, utilizing Network Parameter – 2 of KP-FCNN to train the networks has similar results compared to using Network Parameter – 1. It assumed that the inter-class similarities problems described in Section 7.1.2 conflict with the influences of the random picking method. Hence, the random picking method fails to increase the influence of the minority elements.

8.3. Recommendation

With the target application of construction progress monitoring, this research converts the BIM model into point clouds representing multiple elements in indoor construction scenes. It comprises a beam, ceiling, column, door, floor, railing, stair, wall, and window. However, the current research does not emphasize the clutter elements since the BIM model does not have this information. It contradicts the fact that indoor construction scenes have various clutter elements, including humans, scaffolding, big machinery, etc. Therefore, the synthetic point clouds generated in this research can not be applied to the clutter point clouds. It would be valuable to explore the semantic segmentation of point clouds with clutter at indoor scenes, decreasing the pre-processing procedures.

LIST OF REFERENCES

Arditi, D., & Gunaydin, H. M. (1997). Total quality management in the construction process. *International Journal of Project Management*, *15*(4), 235–243. https://doi.org/https://doi.org/10.1016/S0263-7863(96)00076-2

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Baldwin, J. R., Manthei, J. M., Rothbart, H., & Harris, R. B. (1971). Causes of Delay in the Construction Industry. *Journal of the Construction Division*, *97*(2), 177–187. https://doi.org/10.1061/JCCEAZ.0000305

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). *SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences*.

BIMvision - freeware IFC model viewer. (2023). https://bimvision.eu/

BlenderBIM Add-on - beautiful, detailed, and data-rich OpenBIM. (2023). https://blenderbim.org/

Chen, Y., Hu, V. T., Gavves, E., Mensink, T., Mettes, P., Yang, P., & Snoek, C. G. M. (2020). *PointMixup: Augmentation for Point Clouds*.

CloudCompare - home. (2023). https://www.cloudcompare.org/main.html

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). *CARLA: An Open Urban Driving Simulator*. https://arxiv.org/abs/1711.03938v1

Download PuTTY: latest release (0.78). (2023). https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Emunds, C., Pauen, N., Richter, V., Frisch, J., & van Treeck, C. (2021). *IFCNet: A Benchmark Dataset for IFC Entity Classification*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Gao, B., Pan, Y., Li, C., Geng, S., & Zhao, H. (2020). *Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey and Experimental Study*.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). *A Review on Deep Learning Techniques Applied to Semantic Segmentation*.

Griffiths, D., & Boehm, J. (2019a). A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote Sensing*, *11*(12). https://doi.org/10.3390/rs11121499

Griffiths, D., & Boehm, J. (2019b). *SynthCity: A large scale synthetic point cloud*. https://arxiv.org/abs/1907.04758v1

Gschwandtner, M., Kwitt, R., Uhl, A., & Pree, W. (2011). BlenSor: Blender sensor simulation toolbox. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6939 LNCS*(PART 2), 199–208. https://doi.org/10.1007/978-3-642-24031-7_20/COVER

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2019). *Deep Learning for 3D Point Clouds: A Survey*.

Hajian, H., & Becerik-Gerber, B. (2010). Scan to BIM: Factors Affecting Operational and Computational Errors and Productivity Loss. In T. Brno (Ed.), *Proceedings – The 27th International Symposium on Automation and Robotics in Construction* (pp. 265–272). International Association for Automation and Robotics in Construction (IAARC). https://doi.org/10.22260/ISARC2010/0028

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2019). *RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds*.

Hua, B.-S., Tran, M.-K., & Yeung, S.-K. (2017). *Pointwise Convolutional Neural Networks*.

K. Dhana Sree, Dr., & C. Shoba Bindu, Dr. (2018). Data Analytics: Why Data Normalization. *International Journal of Engineering & Technology*, *7*(4.6), 209. https://doi.org/10.14419/ijet.v7i4.6.20464

Kim, C., Son, H., & Kim, C. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, *31*, 75–82. https://doi.org/https://doi.org/10.1016/j.autcon.2012.11.041

Klokov, R., & Lempitsky, V. (2017). *Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models*.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). *PointCNN: Convolution On $\mathcal{X}$-Transformed Points*.

Ma, J. W., Czerniawski, T., & Leite, F. (2020). Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Automation in Construction*, *113*, 103144. https://doi.org/10.1016/J.AUTCON.2020.103144

Marcus, G. (2018). *Deep Learning: A Critical Appraisal*.

Noichl, F., Braun, A., & Borrmann, A. (2021). BIM-to-Scan for Scan-to-BIM: Generating Realistic Synthetic Ground Truth Point Clouds based on Industrial 3D Models. *Proceedings of the 2021 European Conference on Computing in Construction*, *2*, 164–172. https://doi.org/10.35490/EC3.2021.166

Pan, Y., Gao, B., Mei, J., Geng, S., Li, C., & Zhao, H. (2020). *SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances*.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2016). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 77–85. https://doi.org/10.48550/arxiv.1612.00593

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*.

Sauder, J., & Sievers, B. (2019). *Self-Supervised Deep Learning on Point Clouds by Reconstructing Space*.

Son, H., & Kim, C. (2010). 3D structural component recognition and modeling method using color and 3D data for construction progress monitoring. *Automation in Construction*, *19*(7), 844–854. https://doi.org/https://doi.org/10.1016/j.autcon.2010.03.003

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). *Multi-view Convolutional Neural Networks for 3D Shape Recognition*.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). *Revisiting Unreasonable Effectiveness of Data in Deep Learning Era*.

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., & Guibas, L. (2019). KPConv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*, 6410–6419. https://doi.org/10.1109/ICCV.2019.00651

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. *CVPR 2011*, 1521–1528. https://doi.org/10.1109/CVPR.2011.5995347

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, *22*, 414–421. https://doi.org/https://doi.org/10.1016/j.autcon.2011.10.003

Venkataramanan, A., Laviale, M., Figus, C., Usseglio-Polatera, P., & Pradalier, C. (2021). *Tackling Inter-Class Similarity and Intra-Class Variance for Microscopic Image-based Classification*.

Vosselman, G. (George), & Maas, H.-Gerd. (2010). *Airborne and terrestrial laser scanning*. 318. https://www.researchgate.net/publication/220690347_Airborne_and_Terrestrial_Laser_Scanning

Wang, C., Cheng, M., Sohel, F., Bennamoun, M., & Li, J. (2019). NormalNet: A voxel-based CNN for 3D object classification and retrieval. *Neurocomputing*, *323*, 139–147. https://doi.org/https://doi.org/10.1016/j.neucom.2018.09.075

Wang, F., Zhuang, Y., Gu, H., & Hu, H. (2019). Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis. *IEEE Transactions on Instrumentation and Measurement*, *68*(7), 2671–2673. https://doi.org/10.1109/TIM.2019.2906416

WinSCP :: Official Site :: Free SFTP and FTP client for Windows. (2023). https://winscp.net/eng/index.php

Wu, B., Zhou, X., Zhao, S., Yue, X., & Keutzer, K. (2018). *SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud*.

Xie, Y., Tian, J., & Zhu, X. X. (2019). *Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation*. https://doi.org/10.1109/MGRS.2019.2937630

Xiong, X., Adan, A., Akinci, B., & Huber, D. (2013). Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, *31*, 325–337. https://doi.org/https://doi.org/10.1016/j.autcon.2012.10.006

Xu, X., & Lee, G. H. (2020). *Weakly Supervised Semantic Point Cloud Segmentation:Towards 10X Fewer Labels*.

Yue, X., Wu, B., Seshia, S. A., Keutzer, K., & Sangiovanni-Vincentelli, A. L. (2018). A LiDAR Point Cloud Generator: from a Virtual World to Autonomous Driving. *ICMR 2018 - Proceedings of the 2018 ACM International Conference on Multimedia Retrieval*, 458–464. https://doi.org/10.1145/3206025.3206080

Zhai, R., Zou, J., He, Y., & Meng, L. (2022). BIM-driven data augmentation method for semantic segmentation in superpoint-based deep learning network. *Automation in Construction*, *140*, 104373. https://doi.org/10.1016/J.AUTCON.2022.104373

Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., & Foroosh, H. (2020). *PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation*.

Zhou, H., Zhu, X., Song, X., Ma, Y., Wang, Z., Li, H., & Lin, D. (2020). *Cylinder3D: An Effective 3D Framework for Driving-scene LiDAR Semantic Segmentation*.

APPENDIX :

1. Networks trained using Network Parameter – 1 and tested on the ITC 2022 dataset

a. Networks trained on the Synthetic Point Clouds – 1a

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	3401747 46.97%	29143 0.58%	9348 0.16%	549945 8.99%	6339 0.13%	45934 0.27%	1319951 22.17%	48111 0.80%	3698667 19.76%
column	199 0.01%	271581 32.90%	163474 9.89%	1503 0.08%	0 0.00%	1324 0.01%	7489 0.42%	34849 1.89%	267266 1.84%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	6456 0.40%	559427 22.86%	407619 14.94%	0 0.00%	25407 0.18%	254089 9.90%	15618 0.59%	1065667 6.95%
ceiling	1650352 11.88%	30 0.00%	0 0.00%	82225 0.64%	406796 3.56%	8347717 35.07%	88087 0.70%	28990 0.23%	11794276 46.51%
floor	4250 0.00%	1507 0.02%	13748 0.14%	1212 0.01%	0 0.00%	16330388 76.94%	630741 6.30%	120542 1.20%	139195 0.61%
stair	0 0.00%	6101 0.82%	403 0.03%	123978 6.68%	0 0.00%	258654 0.00%	190041 11.22%	9152 0.00%	1552 0.01%
wall	70320 0.85%	265994 4.42%	1314371 19.19%	1246244 17.48%	1714 0.03%	81724 0.45%	141589 2.03%	2366369 33.63%	5652198 28.65%
window	249994 3.71%	322367 7.16%	498508 9.35%	709398 12.64%	32673 0.76%	115585 0.69%	164259 3.01%	309092 5.60%	5704212 31.32%

b. Networks trained on the Synthetic Point Clouds – 1b

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	5805673 64.19%	125027 1.82%	10756 0.19%	54456 0.96%	26369 0.54%	42077 0.20%	424377 7.20%	243356 2.93%	2377094 23.72%
column	962 0.02%	546296 20.28%	43372 3.20%	0 0.00%	0 0.00%	1409 0.01%	280 0.02%	150270 3.63%	5108 0.09%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	6 0.00%	89300 2.56%	513970 23.92%	467580 20.61%	0 0.00%	13075 0.08%	699452 27.90%	80093 1.62%	470773 7.10%
ceiling	1725233 11.00%	1042 0.01%	0 0.00%	231874 1.89%	209374 1.82%	15309045 56.15%	220979 1.76%	127144 0.85%	4573782 27.44%
floor	164 0.00%	32599 0.30%	14353 0.15%	8369 0.09%	0 0.00%	16373379 66.33%	727471 7.30%	52205 0.42%	33063 0.23%
stair	16 0.00%	27732 1.06%	106 0.01%	0 0.00%	0 0.00%	134324 0.00%	427534 26.15%	149 0.00%	22 0.00%
wall	306156 3.04%	1567820 19.87%	936869 14.30%	1221821 18.31%	2365 0.04%	115500 0.53%	53899 0.78%	5549690 59.46%	1386403 12.56%
window	1142502 13.37%	2250157 35.31%	444017 8.82%	219214 4.25%	371911 8.53%	141933 0.71%	125597 2.33%	1323133 16.93%	2087624 21.93%

c. Networks trained on the Synthetic Point Clouds – 1c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4951449 63.14%	30220 0.60%	1808 0.04%	140630 2.09%	133609 2.08%	111864 0.54%	433677 8.61%	585002 6.48%	2720926 24.44%
column	1320 0.04%	375391 44.44%	35023 5.27%	20443 0.80%	0 0.00%	1561 0.01%	2274 0.26%	271461 5.60%	40220 0.58%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	123 0.00%	5223 0.32%	194069 13.31%	1632478 48.81%	0 0.00%	23822 0.14%	193508 11.71%	23027 0.41%	262051 3.38%
ceiling	667913 4.61%	91 0.00%	0 0.00%	79706 0.60%	2829152 21.64%	14710758 53.69%	31999 0.27%	316514 2.02%	3762340 21.16%
floor	965 0.00%	5745 0.06%	746 0.01%	199450 1.85%	0 0.00%	16954860 68.32%	50568 0.56%	26752 0.20%	2509 0.02%
stair	15 0.00%	10532 1.38%	0 0.00%	230246 9.31%	0 0.00%	302786 0.00%	46264 5.93%	0 0.00%	0 0.00%
wall	215179 2.43%	191097 3.16%	242010 4.13%	1447745 18.69%	50444 0.68%	140611 0.65%	69110 1.14%	5781000 57.58%	3003327 24.72%
window	738695 10.06%	323537 7.15%	108592 2.50%	603891 9.69%	736235 12.42%	149062 0.74%	142051 3.13%	1936969 22.73%	3367056 31.67%

d. Networks trained on the Synthetic Point Clouds – 1d

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	7094814 75.56%	483770 5.21%	0 0.00%	640 0.01%	145728 1.40%	34902 0.20%	12696 0.27%	927262 11.38%	409373 5.46%
column	954 0.02%	682548 13.38%	1785 0.38%	657 0.07%	0 0.00%	1339 0.01%	12 0.00%	55382 1.40%	5020 0.15%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	11 0.00%	714938 12.13%	123601 9.74%	938628 53.15%	0 0.00%	20401 0.14%	43521 3.43%	212707 4.47%	280522 6.83%
ceiling	1984447 12.38%	365 0.00%	0 0.00%	611 0.01%	10847738 63.66%	8567813 35.26%	520 0.00%	308875 2.09%	688104 4.87%
floor	89 0.00%	3917 0.03%	187 0.00%	420 0.00%	3880 0.00%	16972648 78.15%	36265 0.42%	223243 1.83%	954 0.01%
stair	3 0.00%	99201 1.98%	0 0.00%	7014 0.78%	0 0.00%	369278 0.00%	56990 14.31%	57277 0.00%	40 0.00%
wall	349765 3.36%	4330128 42.05%	59388 1.05%	113099 1.83%	29496 0.26%	115763 0.62%	16482 0.29%	4689636 51.19%	1436766 16.89%
window	240075 2.70%	3138643 35.75%	19175 0.46%	136424 2.93%	654665 6.62%	112385 0.66%	40354 0.97%	708486 9.27%	3055881 43.71%

e. Networks trained on the Synthetic Point Clouds – 1e

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	8365448 77.20%	25256 0.52%	260 0.01%	0 0.00%	289982 1.86%	64471 0.47%	0 0.00%	329322 2.72%	34446 0.59%
column	4462 0.07%	199214 29.23%	1446 0.26%	0 0.00%	0 0.00%	4724 0.05%	0 0.00%	531579 6.72%	6272 0.39%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	2 0.00%	4444 0.30%	331830 24.41%	0 0.00%	0 0.00%	73872 0.71%	0 0.00%	1138084 13.08%	786067 32.55%
ceiling	1404028 8.03%	25 0.00%	0 0.00%	0 0.00%	20696030 92.91%	227960 1.12%	0 0.00%	26907 0.14%	43523 0.35%
floor	23 0.00%	360 0.00%	75 0.00%	0 0.00%	19895 0.00%	17155565 96.30%	0 0.00%	63210 0.39%	2385 0.02%
stair	0 0.00%	44 0.01%	203 0.04%	0 0.00%	0 0.00%	508629 0.00%	0 0.00%	79731 0.00%	1316 0.09%
wall	637832 5.38%	226386 3.85%	37488 0.65%	49 0.00%	253425 1.52%	212467 1.44%	0 0.00%	8961150 68.39%	811726 11.91%
window	2152434 20.83%	159760 3.66%	12789 0.30%	133 0.00%	894462 5.91%	139506 1.05%	1278 0.03%	3936322 33.97%	809404 15.27%

f. Networks trained on the Synthetic Point Clouds – 1f

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6554091 77.17%	0 0.00%	682646 10.18%	28690 0.59%	425872 3.08%	20330 0.14%	264279 4.68%	884262 7.72%	249015 3.82%
column	1035 0.02%	0 0.00%	281834 11.17%	741 0.11%	0 0.00%	2536 0.02%	3459 0.24%	457918 6.30%	174 0.01%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	1 0.00%	0 0.00%	1000285 30.17%	241015 16.38%	0 0.00%	29700 0.26%	274426 12.12%	788864 9.78%	0 0.00%
ceiling	875366 5.78%	0 0.00%	604 0.00%	2081 0.02%	17210036 83.99%	3840679 17.95%	36454 0.30%	335442 1.85%	97811 0.74%
floor	0 0.00%	0 0.00%	12280 0.11%	6067 0.07%	0 0.00%	15946516 84.76%	1224342 12.60%	52372 0.34%	0 0.00%
stair	0 0.00%	0 0.00%	1444 0.06%	185876 31.01%	0 0.00%	271147 0.00%	128314 9.22%	3086 0.00%	0 0.00%
wall	344488 3.62%	0 0.00%	1571473 20.36%	46939 0.80%	217735 1.47%	188805 1.20%	91034 1.37%	7598872 60.94%	1081177 14.35%
window	102045 1.28%	0 0.00%	746478 12.04%	97544 2.24%	727963 5.46%	88258 0.62%	171353 3.33%	3677396 33.58%	2495051 41.48%

g. Networks trained on the Synthetic Point Clouds – 1g

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	120396 2.45%	7 0.00%	14940 0.26%	47074 0.99%	8065695 36.21%	4795 0.09%	0 0.00%	856272 4.26%	6 0.00%
column	153 0.02%	58 0.02%	12211 0.79%	59 0.01%	6607 0.00%	0 0.00%	0 0.00%	728609 4.57%	0 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	6811 0.58%	633359 26.98%	147012 10.93%	0 0.00%	44373 2.22%	0 0.00%	1502734 8.98%	0 0.00%
ceiling	1504 0.01%	0 0.00%	0 0.00%	0 0.00%	22134343 76.54%	0 0.00%	0 0.00%	262614 0.98%	12 0.00%
floor	0 0.00%	66 0.00%	1499918 15.30%	7796 0.09%	0 0.00%	1346277 14.24%	0 0.00%	14387466 59.50%	0 0.00%
stair	0 0.00%	209 0.07%	106013 7.18%	137539 29.12%	0 0.00%	264850 0.00%	0 0.00%	81312 0.00%	0 0.00%
wall	254620 4.30%	8 0.00%	53038 0.79%	70 0.00%	1915802 8.23%	1478 0.02%	0 0.00%	8915491 42.19%	16 0.00%
window	338400 7.67%	9 0.00%	41889 0.80%	15134 0.36%	3315877 15.23%	10414 0.21%	0 0.00%	4384302 22.35%	63 0.00%

h. Networks trained on the Synthetic Point Clouds – 1h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6424081 57.93%	189532 2.77%	70561 0.96%	37103 0.64%	34833 0.36%	12985 0.09%	442094 5.08%	247204 3.55%	1650792 25.64%
column	642 0.01%	569040 21.31%	87889 2.75%	3213 0.20%	0 0.00%	349 0.00%	9353 0.21%	50902 1.82%	26305 1.17%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	45 0.00%	7872 0.23%	1076139 26.99%	903827 37.88%	0 0.00%	15930 0.15%	292520 5.51%	37878 1.06%	92 0.00%
ceiling	4150135 23.40%	610 0.00%	2727 0.02%	11525 0.09%	9723355 59.13%	7839135 38.29%	18509 0.12%	61622 0.45%	590855 4.52%
floor	3796 0.00%	6103 0.06%	21113 0.18%	2266 0.02%	0 0.00%	10551127 58.97%	6557780 51.38%	98305 0.89%	1077 0.01%
stair	0 0.00%	557 0.02%	542 0.02%	40 0.00%	0 0.00%	34853 0.00%	553285 12.47%	592 0.00%	0 0.00%
wall	433482 3.58%	1755928 22.32%	2649189 31.57%	1256142 18.50%	39712 0.37%	21888 0.15%	186114 1.92%	3868842 48.43%	929226 12.47%
window	2057616 19.43%	2063710 32.50%	1732254 25.20%	223808 4.25%	693644 7.46%	67916 0.51%	227431 2.77%	471592 7.29%	568117 9.57%

i. Networks trained on the Synthetic Point Clouds – 2a

Coonfusion and F1-Score Matrix

Ground Truth

Prediction		beam	column	door	railing	ceiling	floor	stair	wall	window
	beam	2223970 38.45%	2209 0.05%	2 0.00%	607 0.01%	1056528 6.53%	6703 0.06%	138825 2.41%	4425 0.07%	5675916 37.67%
	column	9 0.00%	16091 2.98%	192974 8.34%	1058 0.11%	0 0.00%	344 0.00%	2034 0.13%	226 0.01%	534957 4.91%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	0 0.00%	1763 0.13%	856101 27.54%	1033925 59.68%	0 0.00%	30 0.00%	159106 6.68%	6286 0.25%	276924 2.37%
	ceiling	132488 1.07%	10 0.00%	1 0.00%	6 0.00%	21286725 93.30%	161020 0.87%	44608 0.36%	16815 0.13%	756800 3.49%
	floor	0 0.00%	222 0.00%	3394 0.03%	45896 0.50%	0 0.00%	14198894 89.64%	1308961 13.31%	101943 1.02%	1582571 8.27%
	stair	0 0.00%	7 0.00%	278 0.01%	49 0.01%	0 0.00%	0 0.00%	588744 38.99%	0 0.00%	645 0.01%
	wall	43458 0.64%	57372 1.00%	1885020 25.09%	37426 0.61%	41764 0.24%	15777 0.12%	54842 0.81%	2489076 35.89%	6515788 40.51%
	window	60271 1.14%	253284 6.00%	944883 15.76%	11935 0.26%	848653 5.42%	56602 0.50%	133333 2.53%	111785 2.06%	5685342 39.03%

j. Networks trained on the Synthetic Point Clouds – 2b

Coonfusion and F1-Score Matrix

Ground Truth

Prediction		beam	column	door	railing	ceiling	floor	stair	wall	window
	beam	4412489 0.00%	2880 0.00%	2114 0.00%	16776 0.00%	836171 0.00%	57841 0.00%	6478 0.00%	74836 0.00%	3699600 0.00%
	column	264 0.00%	50136 0.00%	18432 0.00%	1445 0.00%	1 0.00%	895 0.00%	301 0.00%	904 0.00%	675317 0.00%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	275 0.00%	0 0.00%	70434 0.00%	537389 0.00%	0 0.00%	23135 0.00%	138412 0.00%	50669 0.00%	1513783 0.00%
	ceiling	65942 0.00%	0 0.00%	0 0.00%	0 0.00%	21919635 0.00%	179557 0.00%	2558 0.00%	10832 0.00%	219949 0.00%
	floor	46 0.00%	448 0.00%	1 0.00%	172 0.00%	1765 0.00%	15364836 0.00%	51726 0.00%	122486 0.00%	1700287 0.00%
	stair	13764 0.00%	519 0.00%	0 0.00%	65 0.00%	40 0.00%	358694 0.00%	195836 0.00%	6310 0.00%	14645 0.00%
	wall	51127 0.00%	50630 0.00%	933923 0.00%	78482 0.00%	760358 0.00%	33089 0.00%	9771 0.00%	3769850 0.00%	5453293 0.00%
	window	120938 0.00%	184354 0.00%	218624 0.00%	14313 0.00%	2089809 0.00%	83786 0.00%	16414 0.00%	286572 0.00%	5091278 0.00%

k. Networks trained on the Synthetic Point Clouds – 2c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6957597 0.00%	135370 0.00%	4303 0.00%	1005 0.00%	574189 0.00%	61796 0.00%	60672 0.00%	101670 0.00%	1212583 0.00%
column	576 0.00%	545369 0.00%	665 0.00%	58019 0.00%	0 0.00%	1828 0.00%	47 0.00%	19845 0.00%	121350 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	10 0.00%	2512 0.00%	2265 0.00%	1470732 0.00%	0 0.00%	16627 0.00%	65678 0.00%	367387 0.00%	408728 0.00%
ceiling	245488 0.00%	0 0.00%	0 0.00%	21 0.00%	21676867 0.00%	244838 0.00%	121203 0.00%	15519 0.00%	94537 0.00%
floor	1 0.00%	4175 0.00%	2 0.00%	6740 0.00%	1589 0.00%	16787144 0.00%	33773 0.00%	292301 0.00%	116174 0.00%
stair	70 0.00%	12053 0.00%	0 0.00%	703 0.00%	0 0.00%	349944 0.00%	220048 0.00%	3089 0.00%	3988 0.00%
wall	264389 0.00%	209581 0.00%	906135 0.00%	369207 0.00%	579297 0.00%	80337 0.00%	2026 0.00%	4859591 0.00%	3869960 0.00%
window	620106 0.00%	679933 0.00%	95500 0.00%	44924 0.00%	2137002 0.00%	126953 0.00%	9515 0.00%	1215264 0.00%	3176891 0.00%

l. Networks trained on the Synthetic Point Clouds – 2d

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	7963307 0.00%	149546 0.00%	0 0.00%	247 0.00%	477949 0.00%	43213 0.00%	81532 0.00%	199916 0.00%	193475 0.00%
column	3188 0.00%	526443 0.00%	5829 0.00%	0 0.00%	0 0.00%	1684 0.00%	797 0.00%	178522 0.00%	31234 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	3 0.00%	54050 0.00%	51 0.00%	850026 0.00%	0 0.00%	23293 0.00%	456653 0.00%	949733 0.00%	348 0.00%
ceiling	446178 0.00%	0 0.00%	0 0.00%	27 0.00%	15718133 0.00%	5591318 0.00%	608911 0.00%	21213 0.00%	12693 0.00%
floor	2 0.00%	1466 0.00%	0 0.00%	36 0.00%	0 0.00%	16718230 0.00%	312228 0.00%	209591 0.00%	112 0.00%
stair	0 0.00%	1300 0.00%	0 0.00%	0 0.00%	0 0.00%	19542 0.00%	568489 0.00%	570 0.00%	12 0.00%
wall	502182 0.00%	199698 0.00%	295638 0.00%	227165 0.00%	486262 0.00%	121888 0.00%	39944 0.00%	7688568 0.00%	1579178 0.00%
window	968039 0.00%	347065 0.00%	199440 0.00%	80744 0.00%	1873775 0.00%	101462 0.00%	140956 0.00%	3244523 0.00%	1150084 0.00%

m. Networks trained on the Synthetic Point Clouds – 2e

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	5088481 0.00%	344885 0.00%	0 0.00%	0 0.00%	2059275 0.00%	282383 0.00%	5977 0.00%	717542 0.00%	610642 0.00%
column	2 0.00%	602689 0.00%	0 0.00%	0 0.00%	0 0.00%	504 0.00%	0 0.00%	136193 0.00%	8297 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	7 0.00%	11337 0.00%	0 0.00%	453612 0.00%	0 0.00%	317 0.00%	381 0.00%	1860980 0.00%	7665 0.00%
ceiling	417417 0.00%	1628 0.00%	0 0.00%	0 0.00%	13704054 0.00%	7641203 0.00%	0 0.00%	369595 0.00%	264576 0.00%
floor	33 0.00%	345 0.00%	0 0.00%	0 0.00%	0 0.00%	16305211 0.00%	9650 0.00%	896340 0.00%	30074 0.00%
stair	0 0.00%	98 0.00%	0 0.00%	0 0.00%	0 0.00%	371659 0.00%	50841 0.00%	164649 0.00%	2548 0.00%
wall	178766 0.00%	1146542 0.00%	0 0.00%	36521 0.00%	28638 0.00%	53613 0.00%	464 0.00%	8705587 0.00%	990392 0.00%
window	792240 0.00%	1011994 0.00%	0 0.00%	7542 0.00%	1308717 0.00%	91259 0.00%	18233 0.00%	3521111 0.00%	1354992 0.00%

n. Networks trained on the Synthetic Point Clouds – 2f

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6204201 0.00%	6965 0.00%	0 0.00%	63893 0.00%	1574080 0.00%	562472 0.00%	444 0.00%	285792 0.00%	411338 0.00%
column	21 0.00%	64038 0.00%	0 0.00%	9750 0.00%	0 0.00%	2280 0.00%	0 0.00%	647673 0.00%	23931 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	2 0.00%	0 0.00%	88422 0.00%	0 0.00%	26721 0.00%	0 0.00%	2219228 0.00%	0 0.00%
ceiling	224850 0.00%	0 0.00%	0 0.00%	12834 0.00%	6534715 0.00%	15450683 0.00%	0 0.00%	153140 0.00%	22251 0.00%
floor	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	16957071 0.00%	1838 0.00%	282654 0.00%	0 0.00%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	526550 0.00%	0 0.00%	63253 0.00%	0 0.00%
wall	339870 0.00%	27755 0.00%	0 0.00%	561710 0.00%	545355 0.00%	151162 0.00%	0 0.00%	8650409 0.00%	864262 0.00%
window	1268057 0.00%	280 0.00%	0 0.00%	134328 0.00%	1874196 0.00%	158721 0.00%	60 0.00%	4194696 0.00%	475750 0.00%

o. Networks trained on the Synthetic Point Clouds – 2g

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	216588 0.00%	0 0.00%	0 0.00%	33753 0.00%	8709146 0.00%	31748 0.00%	0 0.00%	101299 0.00%	16651 0.00%
column	80243 0.00%	15574 0.00%	0 0.00%	11 0.00%	8577 0.00%	752 0.00%	0 0.00%	619919 0.00%	22609 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	1 0.00%	7 0.00%	0 0.00%	0 0.00%	0 0.00%	12820 0.00%	0 0.00%	2321250 0.00%	1 0.00%
ceiling	26378 0.00%	2 0.00%	0 0.00%	204 0.00%	22163482 0.00%	145740 0.00%	0 0.00%	62657 0.00%	10 0.00%
floor	226 0.00%	7 0.00%	0 0.00%	0 0.00%	46677 0.00%	16232530 0.00%	0 0.00%	962361 0.00%	0 0.00%
stair	0 0.00%	16 0.00%	0 0.00%	0 0.00%	0 0.00%	373449 0.00%	0 0.00%	216402 0.00%	0 0.00%
wall	878838 0.00%	8791 0.00%	0 0.00%	221777 0.00%	2080480 0.00%	92667 0.00%	0 0.00%	7324900 0.00%	533070 0.00%
window	848492 0.00%	12490 0.00%	0 0.00%	102780 0.00%	3329540 0.00%	100684 0.00%	0 0.00%	3523302 0.00%	188800 0.00%

p. Networks trained on the Synthetic Point Clouds – 2h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	3432912 52.13%	22154 0.43%	165962 2.54%	13488 0.24%	569232 3.41%	62469 0.65%	460145 8.10%	397370 5.25%	3985453 29.77%
column	453 0.02%	273300 26.82%	21429 0.91%	24068 1.67%	0 0.00%	1284 0.02%	53279 3.56%	13878 0.41%	360000 3.91%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	1 0.00%	1799 0.10%	158942 5.04%	1678535 75.07%	0 0.00%	12668 0.21%	459688 20.07%	14790 0.35%	7654 0.08%
ceiling	146456 1.11%	1108 0.01%	12454 0.09%	42 0.00%	21369014 91.54%	446 0.00%	272555 2.21%	97594 0.69%	498804 2.49%
floor	0 0.00%	1500 0.02%	608 0.01%	180307 1.86%	88932 0.00%	9542335 70.14%	284744 2.92%	103315 0.89%	7040120 40.34%
stair	0 0.00%	328 0.03%	0 0.00%	10739 0.79%	0 0.00%	235192 0.00%	343091 24.19%	0 0.00%	453 0.00%
wall	81224 1.07%	161864 2.60%	1885540 24.96%	68486 1.03%	519748 2.93%	51212 0.49%	56594 0.85%	4582721 53.36%	3733134 25.92%
window	399686 6.57%	828390 17.63%	1725676 28.58%	162161 3.17%	1744923 10.77%	62836 0.70%	317001 6.12%	824913 11.67%	2040502 15.83%

q. Networks trained on the S3DIS dataset

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	1738055 30.22%	81852 1.36%	28806 0.59%	0 0.00%	4874162 25.24%	37377 0.29%	10446 0.22%	1503567 20.65%	834920 7.27%
column	0 0.00%	149888 8.21%	32310 4.74%	0 0.00%	357 0.00%	841 0.01%	0 0.00%	26965 0.87%	537334 7.36%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
ceiling	104476 0.84%	10523 0.08%	5 0.00%	0 0.00%	21884656 84.31%	0 0.00%	0 0.00%	276303 1.98%	122510 0.68%
floor	0 0.00%	12 0.00%	1419 0.02%	0 0.00%	196231 0.00%	16175425 96.27%	21381 0.24%	50098 0.44%	797223 5.13%
stair	0 0.00%	0 0.00%	114 0.02%	0 0.00%	0 0.00%	35785 0.00%	500184 86.41%	1410 0.00%	52398 0.73%
wall	311538 3.93%	1060128 12.94%	487236 6.92%	0 0.00%	346987 1.61%	47779 0.32%	35355 0.50%	2856202 30.18%	8329355 60.96%
window	240022 4.57%	1602354 29.11%	65885 1.51%	0 0.00%	2211225 11.76%	66806 0.55%	495 0.01%	740402 10.92%	3178899 28.95%

r. Networks trained on the augmentation of the Synthetic Point Clouds – 2c and the S3DIS dataset

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6735713 81.72%	83784 1.56%	35124 0.72%	3391 0.06%	908735 5.31%	25599 0.20%	61634 1.13%	71538 1.00%	1183667 11.53%
column	595 0.01%	477781 39.96%	1935 0.29%	47530 3.37%	0 0.00%	1029 0.01%	5625 0.44%	16800 0.57%	196402 3.23%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	3134 0.16%	0 0.00%	1731203 78.57%	0 0.00%	2183 0.02%	411565 19.96%	140187 3.73%	46065 0.67%
ceiling	177348 1.19%	0 0.00%	28 0.00%	0 0.00%	21995924 92.64%	72325 0.37%	719 0.01%	6463 0.05%	145666 0.86%
floor	0 0.00%	8105 0.09%	2548 0.03%	187198 1.94%	94916 0.00%	16246055 96.31%	519069 5.45%	79185 0.71%	104457 0.73%
stair	0 0.00%	0 0.00%	0 0.00%	3 0.00%	0 0.00%	24 0.00%	589838 49.58%	0 0.00%	0 0.00%
wall	144608 1.56%	162163 2.54%	510297 8.70%	54169 0.82%	515209 2.84%	77629 0.56%	44865 0.69%	4506567 55.21%	5125016 45.41%
window	316889 4.09%	908904 18.64%	36473 0.84%	48712 0.96%	1574984 9.49%	71225 0.58%	156266 3.16%	362451 5.45%	4630184 47.40%

2. Networks trained using Network Parameter – 1 and tested on the ITC 2021 dataset

a. Networks trained on the Synthetic Point Clouds – 1a

Coonfusion and F1-Score Matrix

Ground Truth

		beam	column	door	railing	ceiling	floor	stair	wall	window
Prediction	beam	5230909 66.07%	78951 2.48%	0 0.00%	69643 2.23%	1612 0.02%	0 0.00%	0 0.00%	11588 0.36%	150071 3.82%
	column	269 0.00%	736232 92.88%	320 0.05%	0 0.00%	0 0.00%	7942 0.15%	8430 2.12%	2365 0.27%	11281 0.73%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	0 0.00%	1484 0.11%	459932 40.19%	5810 0.46%	0 0.00%	1338 0.02%	370 0.04%	2155 0.15%	1357179 65.58%
	ceiling	4890217 36.86%	0 0.00%	0 0.00%	203450 2.40%	10099309 75.44%	1432 0.01%	0 0.00%	920798 10.71%	125309 1.35%
	floor	123 0.00%	1768 0.03%	20 0.00%	3996 0.08%	0 0.00%	9688396 99.60%	17805 0.36%	8449 0.16%	35838 0.59%
	stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	wall	30884 0.54%	0 0.00%	9 0.00%	423129 47.11%	1891 0.03%	3 0.00%	0 0.00%	7179 0.70%	620906 36.58%
	window	139587 2.57%	0 0.00%	0 0.00%	6244 0.96%	432717 7.78%	0 0.00%	0 0.00%	1318 0.17%	10126 0.70%

b. Networks trained on the Synthetic Point Clouds – 1b

Coonfusion and F1-Score Matrix

Ground Truth

		beam	column	door	railing	ceiling	floor	stair	wall	window
Prediction	beam	5181911 73.36%	90201 2.82%	0 0.00%	88292 2.52%	28 0.00%	0 0.00%	0 0.00%	8228 0.29%	174114 3.93%
	column	609 0.01%	747182 92.50%	419 0.08%	8 0.00%	0 0.00%	10809 0.15%	760 0.19%	4787 0.96%	2265 0.11%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	0 0.00%	7955 0.59%	222304 21.68%	220533 13.43%	0 0.00%	1157 0.02%	20403 2.21%	13634 1.33%	1342282 52.21%
	ceiling	3293483 26.53%	394 0.00%	0 0.00%	179824 2.03%	7747010 64.04%	3472318 23.60%	0 0.00%	162170 1.97%	1385316 14.17%
	floor	3 0.00%	1497 0.03%	3 0.00%	38373 0.68%	0 0.00%	9698064 84.56%	1142 0.02%	13922 0.28%	3391 0.05%
	stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	wall	90196 1.87%	1395 0.14%	0 0.00%	926758 72.96%	77 0.00%	0 0.00%	0 0.00%	17477 2.67%	48098 2.19%
	window	17722 0.39%	0 0.00%	0 0.00%	2710 0.26%	205502 4.81%	0 0.00%	0 0.00%	6248 1.53%	357810 18.33%

c. Networks trained on the Synthetic Point Clouds – 1c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	**beam**	**column**	**door**	**railing**	**ceiling**	**floor**	**stair**	**wall**	**window**
beam	5284980 81.35%	135021 4.18%	0 0.00%	7771 0.26%	43033 0.43%	0 0.00%	0 0.00%	65202 2.01%	6767 0.20%
column	55 0.00%	754527 89.48%	3 0.00%	0 0.00%	0 0.00%	12205 0.23%	0 0.00%	36 0.00%	13 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	28 0.00%	20252 1.47%	520638 44.31%	0 0.00%	0 0.00%	12227 0.21%	0 0.00%	98600 7.10%	1176523 77.39%
ceiling	2031602 17.15%	77 0.00%	0 0.00%	10607 0.13%	13845831 90.19%	142426 1.09%	0 0.00%	198540 2.31%	11432 0.13%
floor	4 0.00%	4225 0.08%	1006 0.02%	0 0.00%	0 0.00%	9745615 99.09%	0 0.00%	4478 0.08%	1067 0.02%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
wall	115814 2.71%	4950 0.49%	0 0.00%	357644 48.96%	19283 0.25%	655 0.01%	0 0.00%	582843 57.29%	2812 0.24%
window	17856 0.44%	614 0.08%	0 0.00%	1003 0.21%	556046 7.39%	0 0.00%	0 0.00%	1013 0.13%	13460 1.49%

d. Networks trained on the Synthetic Point Clouds – 1h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	**beam**	**column**	**door**	**railing**	**ceiling**	**floor**	**stair**	**wall**	**window**
beam	5425512 62.00%	58045 1.83%	0 0.00%	6699 0.23%	794 0.02%	6 0.00%	20 0.00%	36076 1.17%	15622 0.56%
column	2452 0.04%	748564 94.54%	289 0.02%	0 0.00%	0 0.00%	3782 0.05%	10909 1.43%	841 0.12%	2 0.00%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	7813 0.59%	1560633 91.44%	524 0.05%	0 0.00%	550 0.01%	256693 19.84%	2052 0.17%	3 0.00%
ceiling	5741213 40.72%	0 0.00%	0 0.00%	2320 0.03%	4882128 46.02%	5613523 36.09%	1 0.00%	1018 0.01%	312 0.00%
floor	0 0.00%	1850 0.03%	6697 0.12%	62 0.00%	0 0.00%	9253810 75.15%	486360 9.25%	7613 0.15%	3 0.00%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
wall	293437 4.50%	442 0.05%	17545 1.31%	207558 31.88%	360 0.01%	0 0.00%	4890 0.53%	559139 66.14%	630 0.11%
window	496603 7.91%	0 0.00%	0 0.00%	986 0.24%	91458 3.29%	9 0.00%	40 0.01%	52 0.01%	844 0.28%

e. Networks trained on the Synthetic Point Clouds – 2a

Coonfusion and F1-Score Matrix

Ground Truth

		beam	**column**	**door**	**railing**	**ceiling**	**floor**	**stair**	**wall**	**window**
Prediction	**beam**	5136918 92.03%	65940 2.13%	7430 0.25%	0 0.00%	129556 1.18%	0 0.00%	0 0.00%	8304 0.27%	194626 4.82%
	column	215 0.01%	579465 81.86%	0 0.00%	0 0.00%	0 0.00%	1733 0.03%	0 0.00%	0 0.00%	185426 11.24%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	27 0.00%	1313 0.11%	457209 39.76%	5758 0.63%	0 0.00%	514 0.01%	0 0.00%	16 0.00%	1363431 62.52%
	ceiling	435694 3.99%	0 0.00%	0 0.00%	0 0.00%	15513549 95.21%	0 0.00%	0 0.00%	0 0.00%	291272 3.10%
	floor	0 0.00%	0 0.00%	8 0.00%	0 0.00%	0 0.00%	9582541 99.09%	0 0.00%	13123 0.25%	160723 2.62%
	stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	wall	45163 1.35%	2173 0.25%	7140 0.92%	0 0.00%	169684 1.95%	0 0.00%	0 0.00%	574272 68.38%	285569 15.79%
	window	2371 0.08%	0 0.00%	0 0.00%	0 0.00%	535107 6.32%	0 0.00%	0 0.00%	0 0.00%	52514 3.36%

f. Networks trained on the Synthetic Point Clouds – 2b

Coonfusion and F1-Score Matrix

Ground Truth

		beam	**column**	**door**	**railing**	**ceiling**	**floor**	**stair**	**wall**	**window**
Prediction	**beam**	5238755 93.26%	26500 0.86%	1 0.00%	4 0.00%	144832 1.30%	0 0.00%	0 0.00%	50871 1.58%	81811 2.38%
	column	537 0.02%	580214 84.48%	1910 0.22%	0 0.00%	0 0.00%	190 0.00%	0 0.00%	0 0.00%	183988 17.44%
	door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	railing	0 0.00%	0 0.00%	935367 67.55%	10446 1.14%	0 0.00%	505 0.01%	200 0.02%	2123 0.16%	879627 55.46%
	ceiling	429608 3.92%	0 0.00%	0 0.00%	0 0.00%	15744109 95.57%	0 0.00%	0 0.00%	58363 0.68%	8435 0.10%
	floor	0 0.00%	0 0.00%	0 0.00%	25 0.00%	0 0.00%	9599628 99.19%	9 0.00%	21973 0.41%	134760 2.43%
	stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
	wall	16681 0.49%	0 0.00%	3692 0.36%	0 0.00%	238521 2.68%	0 0.00%	0 0.00%	773609 77.69%	51498 4.24%
	window	6363 0.20%	0 0.00%	0 0.00%	0 0.00%	579373 6.70%	0 0.00%	0 0.00%	712 0.10%	3544 0.37%

g. Networks trained on the Synthetic Point Clouds – 2c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	5138214 92.98%	42296 1.34%	1 0.00%	19 0.00%	144324 1.29%	0 0.00%	0 0.00%	47975 1.52%	169945 4.79%
column	1 0.00%	716140 93.09%	1 0.00%	0 0.00%	0 0.00%	8483 0.16%	76 0.02%	14340 1.85%	27798 2.39%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	1 0.00%	12942 1.00%	488917 42.12%	130667 13.34%	0 0.00%	86614 1.49%	30724 3.31%	10046 0.77%	1068357 63.08%
ceiling	364448 3.35%	0 0.00%	0 0.00%	0 0.00%	15825631 95.93%	0 0.00%	8 0.00%	21535 0.25%	28893 0.32%
floor	0 0.00%	11 0.00%	0 0.00%	63 0.00%	0 0.00%	9682726 99.14%	118 0.00%	14097 0.27%	59380 1.05%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
wall	4378 0.13%	380 0.04%	4515 0.57%	141 0.02%	230823 2.59%	3 0.00%	0 0.00%	673984 72.24%	169777 12.85%
window	2870 0.09%	0 0.00%	0 0.00%	357 0.10%	551650 6.36%	0 0.00%	0 0.00%	43 0.01%	35072 3.26%

h. Networks trained on the Synthetic Point Clouds – 2h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4991485 91.83%	131192 4.13%	9683 0.26%	0 0.00%	156367 1.40%	0 0.00%	165 0.01%	60230 1.85%	193652 6.64%
column	1 0.00%	678295 86.05%	16149 1.23%	3663 0.93%	0 0.00%	23530 0.45%	45 0.01%	43098 4.98%	2058 0.39%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	1 0.00%	31 0.00%	1790284 97.33%	15297 1.66%	0 0.00%	11617 0.20%	97 0.01%	1642 0.12%	9299 0.88%
ceiling	319772 2.97%	39 0.00%	0 0.00%	0 0.00%	15859104 95.97%	0 0.00%	9535 0.12%	34301 0.40%	17764 0.21%
floor	0 0.00%	0 0.00%	617 0.01%	391 0.01%	0 0.00%	9687900 99.47%	0 0.00%	14562 0.27%	52925 1.05%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
wall	16128 0.50%	67 0.01%	33879 2.31%	410 0.07%	215717 2.41%	278 0.01%	245 0.04%	808260 78.96%	9017 1.31%
window	1446 0.05%	0 0.00%	0 0.00%	0 0.00%	578348 6.65%	0 0.00%	0 0.00%	1096 0.14%	9102 2.06%

i. Networks trained on the S3DIS dataset

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	1745810 47.51%	0 0.00%	6 0.00%	0 0.00%	3670219 27.97%	0 0.00%	0 0.00%	126739 3.65%	0 0.00%
column	116 0.01%	10432 2.68%	5620 1.45%	0 0.00%	11081 0.00%	21 0.00%	262 0.07%	230587 21.29%	508720 33.75%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
ceiling	59716 0.66%	0 0.00%	5 0.00%	0 0.00%	16125751 87.31%	0 0.00%	0 0.00%	55043 0.62%	0 0.00%
floor	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	9637996 99.39%	423 0.01%	30017 0.54%	87959 1.47%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
wall	858 0.04%	0 0.00%	92 0.01%	0 0.00%	303972 2.57%	659 0.01%	54 0.00%	955815 44.34%	1650819 63.99%
window	7 0.00%	0 0.00%	0 0.00%	0 0.00%	588964 5.53%	0 0.00%	0 0.00%	1021 0.10%	0 0.00%

3. Networks trained using Network Parameter – 2 and tested on the ITC 2022 dataset

a. Networks trained on the Synthetic Point Clouds – 1a

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	5065862 47.20%	13886 0.28%	3485 0.06%	132 0.00%	71511 0.89%	15654 0.12%	3073618 29.26%	672718 7.21%	192319 2.11%
column	420 0.01%	136807 15.88%	93061 4.35%	0 0.00%	0 0.00%	2093 0.02%	27347 0.43%	474460 9.20%	13505 0.27%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	352 0.00%	7333 0.44%	784517 26.77%	0 0.00%	0 0.00%	36685 0.38%	684666 9.62%	112141 1.89%	708603 12.39%
ceiling	4726199 27.20%	320 0.00%	0 0.00%	1608 0.01%	6135328 41.80%	320334 1.63%	5831244 34.00%	447573 2.80%	4935867 31.34%
floor	2259 0.00%	9876 0.11%	7784 0.07%	0 0.00%	0 0.00%	16221743 94.72%	910852 6.25%	23904 0.18%	65197 0.50%
stair	556 0.00%	4271 0.55%	2 0.00%	0 0.00%	0 0.00%	165429 0.00%	418631 6.70%	938 0.00%	0 0.00%
wall	493733 4.20%	213481 3.52%	1681616 22.93%	278908 4.88%	25861 0.29%	135916 0.97%	519065 4.51%	5806210 56.09%	1985733 19.62%
window	2064766 20.18%	588864 12.97%	957478 16.46%	2210 0.05%	723961 9.61%	111117 0.88%	434721 4.35%	2023884 22.91%	1199087 13.94%

b. Networks trained on the Synthetic Point Clouds – 1b

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4333068 38.58%	2833 0.06%	0 0.00%	135439 2.28%	59849 1.09%	65004 0.46%	714669 12.47%	135947 1.58%	3662376 24.68%
column	2594 0.04%	271432 29.85%	88793 6.08%	4205 0.24%	82 0.00%	2121 0.02%	55333 3.57%	205000 4.66%	118137 1.11%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	63788 3.75%	947753 42.07%	0 0.00%	0 0.00%	176738 1.62%	594609 25.37%	403440 7.77%	147963 1.29%
ceiling	7949709 44.47%	126 0.00%	0 0.00%	67842 0.54%	1647825 13.58%	1513608 7.24%	138388 1.12%	126618 0.83%	10954357 50.99%
floor	16 0.00%	7266 0.08%	7597 0.08%	0 0.00%	226 0.00%	16949693 92.42%	212826 2.17%	55396 0.44%	8499 0.04%
stair	0 0.00%	14621 1.76%	34 0.00%	0 0.00%	0 0.00%	433481 0.00%	141509 9.62%	276 0.00%	4 0.00%
wall	168722 1.38%	255513 4.18%	865966 13.01%	1314187 18.87%	33918 0.52%	157685 1.03%	167113 2.48%	5452525 56.81%	2724894 17.19%
window	900819 8.39%	455256 9.92%	260810 5.08%	1269087 23.29%	124270 2.49%	139309 1.01%	328050 6.27%	1675699 20.74%	2952788 20.59%

c. Networks trained on the Synthetic Point Clouds – 1c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	3743798 51.38%	8385 0.14%	0 0.00%	53476 0.98%	113678 0.94%	29830 0.21%	429192 7.53%	104338 1.37%	4626488 34.65%
column	1386 0.04%	420385 24.34%	23963 3.24%	0 0.00%	0 0.00%	1926 0.02%	4648 0.31%	83215 2.42%	212170 2.31%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	195 0.00%	29934 1.19%	269084 17.55%	0 0.00%	0 0.00%	59684 0.54%	734713 31.75%	147162 3.48%	1093555 10.98%
ceiling	1005597 7.22%	2 0.00%	0 0.00%	76974 0.64%	14187922 75.73%	2521067 11.93%	15437 0.13%	237195 1.66%	4354279 21.78%
floor	1771 0.00%	17617 0.18%	1698 0.02%	0 0.00%	200 0.00%	16730431 90.21%	423343 4.33%	33936 0.29%	32601 0.19%
stair	140 0.00%	24657 1.50%	0 0.00%	0 0.00%	0 0.00%	248352 0.00%	316619 21.96%	42 0.00%	5 0.00%
wall	184913 2.23%	873911 12.62%	278730 4.70%	1213542 18.71%	8835 0.07%	148831 0.96%	142945 2.13%	4638679 53.73%	3650137 25.41%
window	526854 7.76%	1332315 24.64%	157851 3.57%	484678 9.76%	760985 6.57%	111159 0.80%	226712 4.36%	883084 12.41%	3622450 28.19%

d. Networks trained on the Synthetic Point Clouds – 1h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4674770 53.99%	72334 1.17%	174024 2.41%	91270 1.70%	816895 6.91%	180847 1.16%	1547557 22.87%	829353 9.92%	722135 10.59%
column	859 0.02%	466357 23.29%	117093 3.86%	0 0.00%	152 0.00%	1379 0.01%	5841 0.23%	90719 2.17%	65297 2.47%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	3 0.00%	27836 1.00%	1591532 41.61%	0 0.00%	0 0.00%	65042 0.53%	484036 14.33%	149800 3.01%	16086 0.47%
ceiling	1405383 9.18%	303 0.00%	3183 0.02%	32074 0.27%	12606425 68.26%	5752346 25.83%	502743 3.75%	170497 1.14%	1925519 14.30%
floor	723 0.00%	35580 0.35%	27590 0.24%	0 0.00%	221 0.00%	15693423 79.71%	1195598 11.04%	31228 0.25%	257156 2.36%
stair	5 0.00%	14464 0.75%	384 0.01%	0 0.00%	0 0.00%	237082 0.00%	337533 13.47%	365 0.00%	48 0.00%
wall	272479 2.82%	972931 13.52%	2016032 24.50%	1027405 16.06%	164147 1.28%	96199 0.58%	104600 1.34%	5259910 56.11%	1226820 15.66%
window	1854901 22.74%	1666447 29.33%	1386152 20.65%	499554 10.24%	951831 8.41%	106989 0.71%	245478 3.92%	1076775 13.70%	317961 5.03%

e. Networks trained on the Synthetic Point Clouds – 2a

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4270596 60.29%	13673 0.28%	1036919 15.27%	245 0.01%	1149496 6.50%	27422 0.23%	79478 1.45%	93285 1.50%	2438071 20.68%
column	94 0.00%	71471 9.84%	112083 4.30%	1 0.00%	16 0.00%	2131 0.03%	17602 1.37%	26422 1.30%	517873 6.81%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	139 0.00%	358 0.02%	320936 9.43%	306454 21.63%	0 0.00%	15022 0.17%	732084 35.25%	76373 2.70%	882849 10.51%
ceiling	137667 1.00%	0 0.00%	3657 0.03%	0 0.00%	21782595 89.51%	117750 0.63%	3498 0.03%	19759 0.15%	333547 1.81%
floor	122 0.00%	1766 0.02%	3697 0.03%	0 0.00%	12323 0.00%	14714272 91.12%	67457 0.71%	89034 0.87%	2353146 14.84%
stair	6445 0.00%	2495 0.39%	536 0.02%	0 0.00%	0 0.00%	55382 0.00%	520206 43.18%	826 0.00%	3817 0.05%
wall	76270 0.94%	110588 1.87%	2231238 28.59%	169440 2.91%	856433 4.58%	52767 0.40%	132692 2.05%	2714266 37.52%	4796829 37.47%
window	565939 8.60%	504986 11.46%	760725 12.10%	22742 0.53%	2470254 14.37%	69871 0.60%	266918 5.38%	306279 5.36%	3138374 27.81%

f. Networks trained on the Synthetic Point Clouds – 2b

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	4205828 60.78%	74963 1.26%	4 0.00%	24 0.00%	1885655 10.42%	31795 0.24%	221601 4.06%	195536 2.73%	2493779 26.67%
column	37 0.00%	482904 27.04%	38398 2.30%	0 0.00%	74 0.00%	4466 0.05%	25805 2.02%	12937 0.43%	183078 3.54%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	35084 1.36%	193692 7.86%	199570 15.55%	120 0.00%	92902 0.93%	642364 31.02%	243568 6.44%	926587 15.54%
ceiling	123166 0.91%	547 0.00%	0 0.00%	84 0.00%	21942996 88.72%	228319 1.14%	15226 0.13%	23125 0.17%	65010 0.41%
floor	8306 0.00%	2227 0.02%	189 0.00%	0 0.00%	5292 0.00%	16892718 97.01%	105765 1.11%	113152 1.01%	114068 0.85%
stair	0 0.00%	4223 0.25%	52 0.00%	0 0.00%	0 0.00%	64487 0.00%	510710 42.61%	10225 0.00%	432 0.01%
wall	80743 1.02%	877952 12.57%	1831225 26.66%	26719 0.47%	637076 3.33%	162846 1.13%	58053 0.90%	3938376 48.10%	3527533 34.03%
window	311829 4.86%	1345852 24.63%	531294 9.93%	5761 0.14%	2598205 14.77%	108514 0.84%	227598 4.59%	696794 10.45%	2280241 25.77%

g. Networks trained on the Synthetic Point Clouds – 2c

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	6635016 79.64%	302852 4.74%	4593 0.10%	0 0.00%	569487 3.42%	33538 0.25%	5044 0.10%	44494 0.54%	1514161 16.43%
column	155 0.00%	569428 25.81%	0 0.00%	106 0.01%	0 0.00%	5495 0.06%	3355 0.39%	22284 0.56%	146866 2.92%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	11 0.00%	29915 1.00%	0 0.00%	669704 44.46%	0 0.00%	85457 0.85%	435169 26.15%	686027 14.33%	427538 7.34%
ceiling	270023 1.80%	9987 0.08%	0 0.00%	0 0.00%	21434144 92.10%	241175 1.20%	307 0.00%	17575 0.12%	425262 2.68%
floor	458 0.00%	5627 0.05%	0 0.00%	1 0.00%	83 0.00%	17035518 97.24%	30482 0.33%	96442 0.79%	73534 0.55%
stair	736 0.00%	12833 0.60%	0 0.00%	0 0.00%	0 0.00%	113338 0.00%	460243 58.10%	2420 0.00%	207 0.00%
wall	201627 2.16%	1084974 14.66%	259792 4.55%	5185 0.09%	428792 2.43%	160589 1.11%	7140 0.12%	4906627 53.38%	4085797 39.94%
window	445772 5.69%	1649067 28.02%	5543 0.13%	3597 0.08%	1716705 10.64%	119059 0.92%	52900 1.16%	1467752 19.12%	2645693 30.37%

h. Networks trained on the Synthetic Point Clouds – 2h

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	3586257 54.54%	15562 0.31%	1174755 13.56%	24328 0.47%	539630 3.35%	50042 0.40%	659151 11.86%	85715 1.29%	2973745 28.36%
column	180 0.01%	18727 2.32%	89801 2.00%	0 0.00%	0 0.00%	5502 0.07%	2944 0.21%	1772 0.07%	628769 9.97%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	2160 0.13%	1453818 27.56%	593294 33.64%	0 0.00%	63015 0.68%	162779 7.50%	44644 1.37%	14201 0.20%
ceiling	102092 0.77%	676 0.01%	66907 0.44%	9627 0.08%	21261304 93.39%	1966 0.01%	305646 2.50%	35652 0.27%	614603 3.59%
floor	0 0.00%	803 0.01%	11600 0.09%	3945 0.04%	88658 0.00%	15744150 94.24%	55863 0.58%	96320 0.90%	1240566 8.53%
stair	0 0.00%	155 0.02%	1651 0.04%	333 0.04%	0 0.00%	72774 0.00%	500337 38.54%	248 0.00%	14423 0.23%
wall	69279 0.91%	174646 2.91%	3085245 31.88%	289721 4.70%	294230 1.72%	125853 0.92%	35900 0.55%	3593070 46.92%	3472579 30.20%
window	283474 4.67%	655711 14.61%	2333310 28.59%	272237 5.85%	951085 6.09%	106426 0.88%	283621 5.61%	318809 5.19%	2901415 29.06%

i. Networks trained on the S3DIS dataset

Coonfusion and F1-Score Matrix

Ground Truth

Prediction	beam	column	door	railing	ceiling	floor	stair	wall	window
beam	3861498 54.75%	8139 0.17%	514488 10.47%	0 0.00%	1230176 7.12%	33123 0.26%	14422 0.30%	543962 7.63%	2903377 21.31%
column	0 0.00%	1875 0.33%	13154 1.80%	0 0.00%	47 0.00%	616 0.01%	318 0.05%	52378 1.77%	679309 7.19%
door	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
railing	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%
ceiling	160921 1.17%	44 0.00%	22781 0.20%	0 0.00%	21607542 90.31%	0 0.00%	8 0.00%	185729 1.35%	421448 2.08%
floor	0 0.00%	731 0.01%	1522 0.02%	0 0.00%	733111 0.00%	15911951 95.21%	101865 1.14%	218540 1.95%	273805 1.55%
stair	0 0.00%	0 0.00%	0 0.00%	0 0.00%	170 0.00%	85969 0.00%	476306 77.45%	26515 0.00%	943 0.01%
wall	226335 2.45%	72288 1.04%	150183 2.12%	0 0.00%	171052 0.88%	76943 0.52%	41625 0.59%	3552502 38.14%	9183902 58.11%
window	747196 11.41%	300643 7.08%	14350 0.33%	0 0.00%	1711906 10.20%	76575 0.63%	5599 0.13%	576709 8.70%	4673110 35.62%