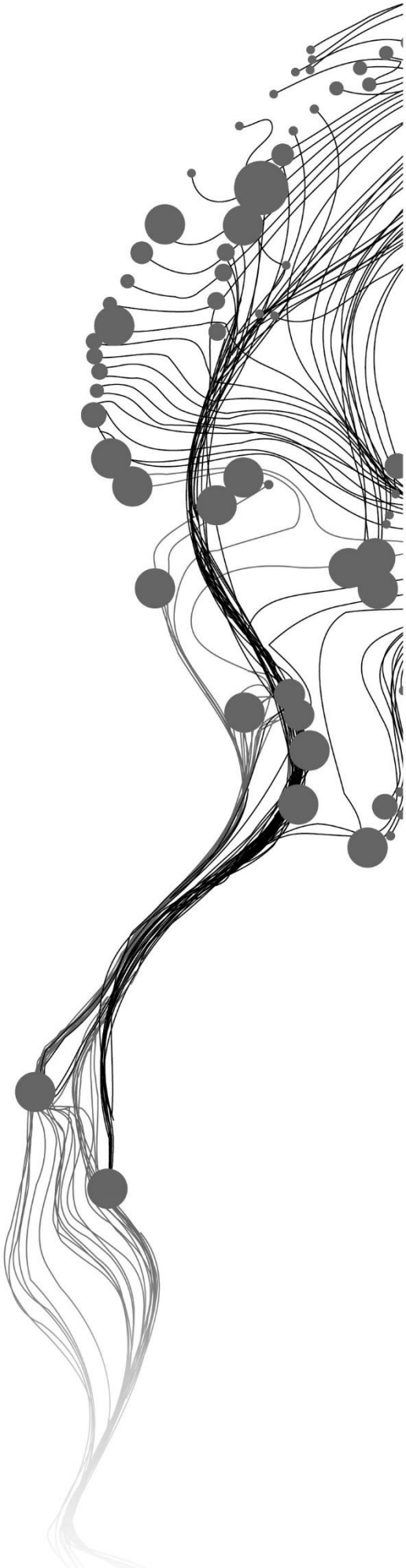
DEEP LEARNING ON 3D POINT CLOUDS FOR SAFETY-RELATED ASSET MANAGEMENT IN BUILDINGS

GEETHANJALI ANJANAPPA

July 2022

SUPERVISORS:
Dr. Ville. V. Lehtola
Dr. Ir. S.J. Oude Elberink

ADVISOR:
Robert L. Voûte (CGI Inc.)



DEEP LEARNING ON 3D POINT CLOUDS FOR SAFETY-RELATED ASSET MANAGEMENT IN BUILDINGS

GEETHANJALI ANJANAPPA
Enschede, The Netherlands, July 2022

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

Specialization: Geoinformatics

SUPERVISORS:
Dr. Ville. V. Lehtola
Dr. Ir. S.J. Oude Elberink

ADVISOR:
Robert L. Voûte (CGI Inc, Nederland)

THESIS ASSESSMENT BOARD:
Prof.dr.ir. M.G. Vosselman (Chair)
Dr. Matti Vaaja (External Examiner - Dept of Built Environment, Aalto University)
Drs. J.P.G Bakx
Dr. Ville. V. Lehtola
Dr. Ir. S.J. Oude Elberink

ITC CGI

ABSTRACT

Buildings are equipped with multiple safety-related assets depending on the need and functionality. For example, doors and windows for access; fire blankets, alarms, sprinklers, and extinguishers for fire suppression; exit and escape signs to navigate. It is vital to have up-to-date information on these assets for emergencies. As a part of infrastructure maintenance, asset management systems document assets within a building, maintain their records, and continuously monitor them to improve the asset's performance. In this regard, an asset management system can establish a centralized system for safety-related assets in buildings to (i) enable finding them quickly and effortlessly; (ii) keep up-to-date records for first responders; (iii) monitor the entire safety-related infrastructure; (iv) keep a check on the building's compliance with safety standards.

A crucial step in establishing an asset management system for safety-related assets is to identify these assets within the building. With the advancement of 3D sensing and Deep Learning (DL) technologies, it is possible to automate the identification of essential assets using 3D point clouds. In collaboration with CGI Inc., this research explored the scope of using 3D point clouds and the DL scene segmentation approach to identify safety-related assets within buildings. In this context, we adapted the Kernel Point-Fully Convolutional Network (KP–FCNN) to perform scene segmentation to identify safety-related assets. The research focused on common assets in most buildings like ceiling lights, exit signs, ventilation ducts, windows, doors, stairways, fire switches, and extinguishers. We used point cloud datasets acquired from three different 3D sensors to evaluate the designed method, namely, the depth camera (S3DIS), an MLS (HPS dataset), and a consumer-grade lidar sensor (iPhone dataset). In addition to standard evaluation metrics, asset identification rate (AIR) was used to evaluate the rate of correctly identified asset instances for S3DIS and HPS datasets. The iPhone dataset was only qualitatively assessed.

The results from various experiments showed that our workflow could successfully identify small-sized assets like fire switches and exit signs with a 100% AIR in some cases. The designed method proved invariant and robust by successfully performing scene segmentation for the three chosen datasets to identify safety-related assets. For S3DIS Area-6, the results obtained show that the method identified all the chosen assets with an AIR>75%. When assessed on new and unfamiliar buildings, the method generalized well and successfully identified small-sized assets like fire switches with AIR>80% (S3DIS Area-5 and HPS Scan 5). However, it failed to identify new-looking representations of fire extinguishers achieving a zero AIR for S3DIS Area-5 and HPS test scans. But through domain adaptation (transfer learning), we demonstrated that the method could effectively learn the new representations of fire extinguishers from the HPS scans when trained on them, later achieving 100% AIR. Additionally, based on qualitative analysis, we determined that it is possible to identify safety-related assets within a building using the point clouds obtained from simpler lidar devices, like iPhone.

Keywords: Asset management, 3D point clouds, Deep learning, Scene segmentation, Safety-related assets, KP–FCNN, Stanford 3D Indoor Scene Dataset (S3DIS), Mobile Lidar Scanner (MLS), iPhone lidar.

ACKNOWLEDGEMENTS

Throughout the development of this research and thesis, I received a lot of support and encouragement from various people.

First and foremost, I am grateful to my first supervisor, Dr. Ville. V. Lehtola, for offering such an exciting topic for research. Throughout the research period, his expertise, constant support, and constructive remarks helped and guided me in shaping this work. Our regular meetings with discussions, brainstorming ideas, and his thought-provoking questions were instrumental in developing and improving the quality of the research and methodology. At times when I would get anxious and worry about the direction of the work, he was always patient, understanding, and encouraging.

I am also grateful to my second supervisor Dr. Ir. S.J. Oude Elberink, who was also supportive and guided me through this research with his suggestions and quick responses to my questions. His feedback, insights on the datasets, and informative remarks helped me improve my work.

A special thanks to my internship supervisor and thesis advisor, Robert L. Voûte, vice president of CGI Nederland, for providing me with the opportunity to be a graduate intern throughout this research period. Our regular update meetings helped me keep in check and plan my work efficiently. Thank you for believing in me, always asking if I needed any help, encouraging me, and being curious about everything.

I also like to acknowledge the author of KP-FCNN, Thomas Hugues, for his timely response to all my queries regarding the network and for engaging in an informative discussion, which helped me make certain decisions for my work.

Further, I want to thank my family for their support, especially my beloved sister, Pallavi, for believing in me and encouraging me. I also want to thank my dear friend, Prathviraj, who was there to encourage me at the worst and celebrate with me at the best moments of this research. Thank you for constantly encouraging me, helping me stay sane, and always having the patience to listen to my stories throughout the research, even though they made little sense to you.

Finally, I am grateful to all my friends back home and at ITC, who made sure I was not lost at work and took timely breaks to have fun. I also extend my gratitude to ITC for supporting my education here by providing the Excellence Scholarship.

Happy Reading!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
AIR	Asset Identification Rate
ANN	Artificial Neural Network
BIM	Building Information Model
CNN	Convolutional Neural Network
CPUs	Central Processing Units
CUDA	Compute Unified Device Architecture
DGCNN	Dynamic Graph CNN
DL	Deep Learning
FN	False Negative
FP	False Positive
GIS	Geographic Information System
GPU	Graphics Processing Unit
HPS	Human POSEitioning System
IMU	Inertial Measurement Unit
IoU	Intersection over Union
KPConv	Kernel Point Convolution
KP-FCNN	Kernel Point - Fully Convolutional Neural Network
Lidar	Light detection and ranging
MLP	Multi-Layer Perceptron
MLS	Mobile Laser Scanner
MVPNet	Multi View PointNet
OA	Overall Accuracy
OGC	Open Geospatial Consortium
RANSAC	Random Sample Consensus
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
ROI	Region Of Interest
S3DIS	Stanford 3D Indoor Scene Dataset
SFM	Structure From Motion
SLAM	Simultaneous Localization and Mapping
SOR	Statistical Outlier Remover
SPG	Super Point Graphs
SSH	Secure Shell
TLS	Terrestrial Laser Scanner
TN	True Negative
TP	True Positive

1. INTRODUCTION

This chapter introduces the current research with a brief background in Section 1.1, describing the motivation and research identification in Section 1.2. Section 1.3 defines the research problem and innovation, formulating the objectives in Section 1.4. Section 1.5 outlines the overall thesis structure.

1.1. Background

Depending on the need and functionality, any building, for example, a university, office space, hospital, or shopping mall, is equipped with various assets like furniture, safety-related features[1], and electrical equipment. These assets are essential for building utilities and providing service values to them (Xie et al., 2020). In particular, for safety concerns and emergencies, it is vital to have up-to-date information on safety-related assets like doors and windows for access; fire blankets, alarms, sprinklers, and extinguishers for fire suppression; exit and escape signs to navigate (Hossain et al., 2021; Kostoeva et al., 2019). It is also necessary to regularly inspect and document the presence, location, and working conditions of these safety-related assets to check the building's compliance with safety standards over its lifespan. Additionally, precise locations of certain assets could play a vital role during an emergency or disaster response situation for first responders to aid the occupants of the building efficiently.

In this context, **Asset Management Systems** carry out monitoring and improvement mechanisms to fully and effectively realize the value and capability of an asset over its entire life cycle. (ISO, 2018; United Nations, 2021). Figure 1.1 illustrates the scope of asset management, where organizations provide the information of various assets associated with their business buildings to an asset management system. Later, this system uses the information to focus on (i) Asset position, condition, life extension, and interventions, (ii) Asset lifecycle activities – availability, performance, and reliability, and (iii) Asset databases and IT systems. These activities enable seamless building functionality, further improving the affiliated organization's sustainability and efficiency.

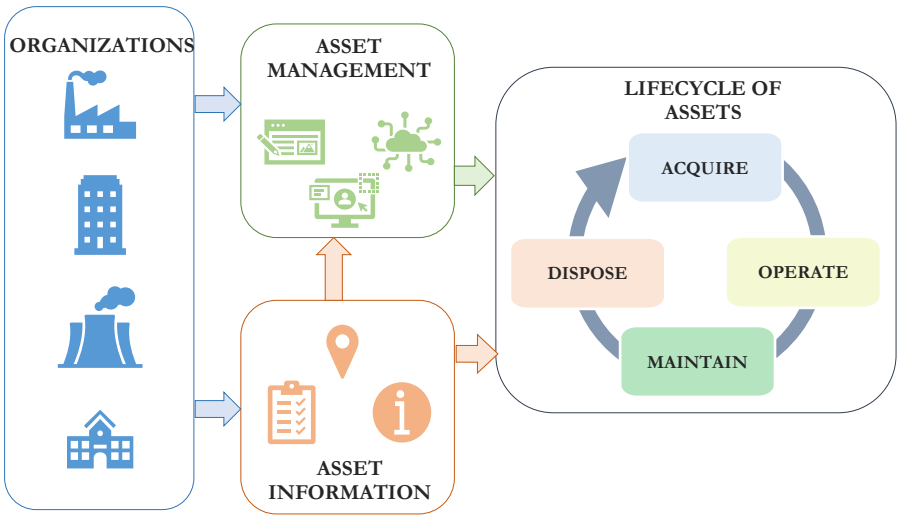


Figure 1.1: Scope of asset management system with organizations for their business buildings (Image source: author).

For safety-related assets, an asset management system could establish a centralized system to (i) enable finding them throughout the building effortlessly; (ii) keep up-to-date records for first responders and facility

[1] Equipment, devices, and systems to support the safety functionalities of a building.

managers; (iii) monitor the entire safety-related infrastructure (Esri, 2019). To establish such an asset management system for safety-related assets, the first and essential component is to obtain the asset information module, shown in Figure 1.1. It includes identifying the assets within the building by finding where the asset is (its spatial location) and classifying what asset it is (its categorical name).

1.2. Motivation and Research Identification

Traditional methods to get the assets information in a building are through manual surveying. These surveys have an operator physically going around the building to identify and note asset details. Later, Kostoeva et al. (2019) developed an interactive smartphone application based on images to identify assets, with the operator pinpointing assets while capturing images. Using the Red-Green-Blue (RGB) color data from images, a neural network predicted the labels that the operator confirmed. Further, with the increasing availability of multiple geospatial tools, Geographic Information Systems (GIS) are used for asset management in organizations. GIS systems analyze, manage, and visualize the asset details, generate indoor maps and explore spatial relationships using a building information model (BIM) (Teixeira et al., 2021). Such a GIS-based solution by Esri was, for example, implemented at the Raleigh Water Pump Station, United States. It utilized the ArcGIS Indoors tool to document information about assets in a building, generating a 3D representation of the pump station like Figure 1.2(b), with assets indicated using icons.

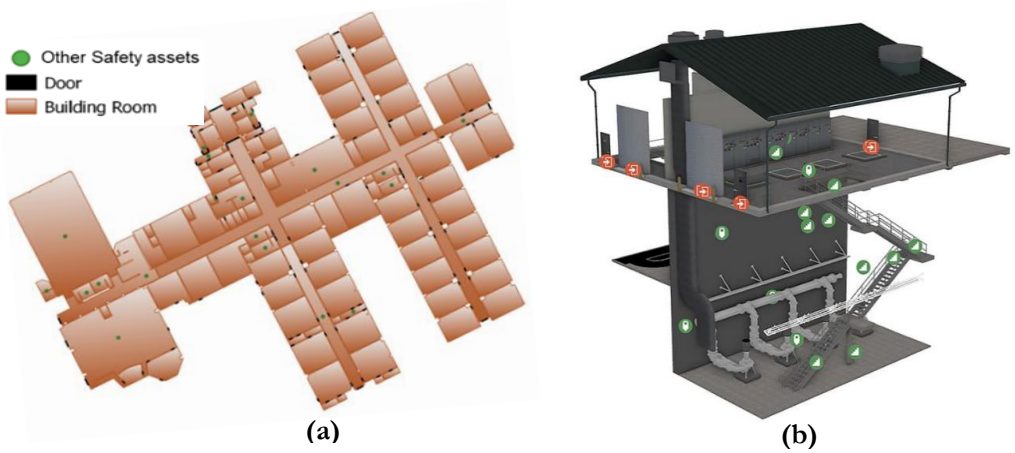


Figure 1.2: (a) 2D building layout with safety assets specified (Image source: Chen, 2019). (b) 3D representation of building with assets indicated using icons (Image source: Esri).

When dealing with large-scale buildings, surveys and capturing images are time-consuming, tedious, and operator-dependent to identify assets and obtain their information (Balamurugan and Zakhor, 2019). Additionally, these methods only provide asset locations in the form of two-dimensional (2D) buildings and floor plans like Figure 1.2(a) and do not provide the spatial three-dimensional (3D) location of assets in the real world (Warsop and Singh, 2010; Bello et al., 2020). Further, even though the GIS-based method by Esri provided asset information in 3D, they used operators to collect data for each asset and then manually generated the 3D location using the ArcGIS Indoors tool. Therefore, to efficiently obtain the asset information in 3D, we need an effective approach (i) to obtain 3D data representing the assets in buildings and (ii) to utilize such data to identify assets.

With the recent technological developments, 3D data as 3D point clouds[2] are more widely used data formats for real-world 3D representation and can provide the 3D location of objects within the building (Bello et

[2] A set of points in 3D metric or Euclidean space with x, y, and z coordinates distributed across the surfaces of the objects; may also include additional information like RGB color and surface normals (Bello et al., 2020).

al., 2020; Liu et al., 2019; Lehtola et al., 2017). One way of obtaining 3D point clouds of buildings is through the Structure from Motion (SFM) and photogrammetry techniques using multiple images. However, these generated point clouds have low resolution, are prone to noise, lack precision, and are computationally time-consuming for large-scale buildings (Thomas, 2019; Xu et al., 2019; Liu et al., 2019). Also, buildings are usually cluttered[3] with objects like furniture and electrical equipment, causing occlusions and constant interruptions during data acquisition using cameras (Lehtola et al., 2017; Soilán et al., 2019). Hence, these methods are time-consuming, require proper planning, and are not optimal when dealing with multistorey and large-scale buildings.

With the development of affordable 3D sensors, depth cameras, and laser scanners using light detection and ranging (lidar) can easily acquire 3D point clouds for large-scale buildings (Guo et al., 2021; Lehtola et al., 2017). These sensors provide 3D point clouds with rich geometry and color information without much effort, hassle, and planning (Lehtola et al., 2017; Thomas, 2019). For example, it is possible to scan large-scale and cluttered multistorey buildings in just one day using an indoor mobile laser scanner (MLS) in the form of trolleys or backpack systems, as shown in Figure 1.3. Therefore, point clouds using these 3D sensors can easily be obtained, which can be utilized to obtain up-to-date information on safety-related assets for asset management systems throughout the entire lifespan of the buildings.

Figure 1.3: Commercial indoor MLS devices. Left - trolley and right - backpack (Image Source: NavVis; Leica Geosystems).

Conventional methods to identify objects in buildings using 3D point clouds apply object detection[4] and semantic segmentation[5] techniques using algorithms like clustering and random sample consensus (RANSAC) (Vosselman and Maas, 2010, p. 63). For example, Mattausch et al. (2014) used planar patches generated from the point clouds and density-based spatial clustering to detect objects in an indoor environment. Similarly, Su et al. (2021) identified objects in the cluttered indoor scene using RANSAC and graph-clustering approaches. Later, as an open-science practice, Open Geospatial Consortium (OGC) set up an initiative to generate 3D indoor models with safety-related assets to support first responders using CityGML (Geography Markup Language) with 3D point clouds (Chen, 2019). However, these conventional and CityGML methods are step-wise procedures that use numerous workflows and are manually configured using the object's pre-defined geometric and structural features (Vosselman and Maas, 2010, p. 63). But, using deep learning, it is possible to automate the process of identifying objects (here, safety-related assets) in 3D point clouds, which is the main aim of this research.

Deep learning (DL) is a specific type of machine learning method which uses artificial neural networks (ANNs) inspired by the biological neurons of the human brain (Goodfellow et al., 2016, p. 96; Johnson and Khoshgoftaar, 2019; Bello et al., 2020). DL models use multiple ANN layers to process and learn discriminative feature representations of data to achieve various levels of abstraction (Liu et al., 2019; Bello et al., 2020). Recently, DL techniques with 3D point clouds for indoor spaces have succeeded in different fields like robotics, virtual reality, 3D building modeling, and indoor mapping (Goodfellow et al., 2016; Liu et al., 2019; Guo et al., 2021).

[3] Objects scattered or disorganised in a building.

[4] Object detection methods identify objects by creating 3D bounding boxes around each detected object and assigning labels to only the detected point sets (Bello et al., 2020; Guo et al., 2021; Liu et al., 2019).

[5] In semantic segmentation, each point in the 3D point cloud is labeled into a semantic category (Bello et al., 2020; Guo et al., 2021; Liu et al., 2019).

DL methods using object detection and semantic segmentation techniques automatically learn features from rich spatial and contextual information offered by 3D point clouds to identify and locate objects in a scene (Guo et al., 2021). In object detection, DL methods identify only the objects of interest by creating bounding boxes around them; in Figure 1.4(a), only tables and chairs are identified in an indoor scene. However, with semantic segmentation (also known as scene segmentation), DL methods identify all the semantic categories in the scene, which is spatially and conceptually more informative, as shown in Figure 1.4(b), with different colors representing different semantic categories.

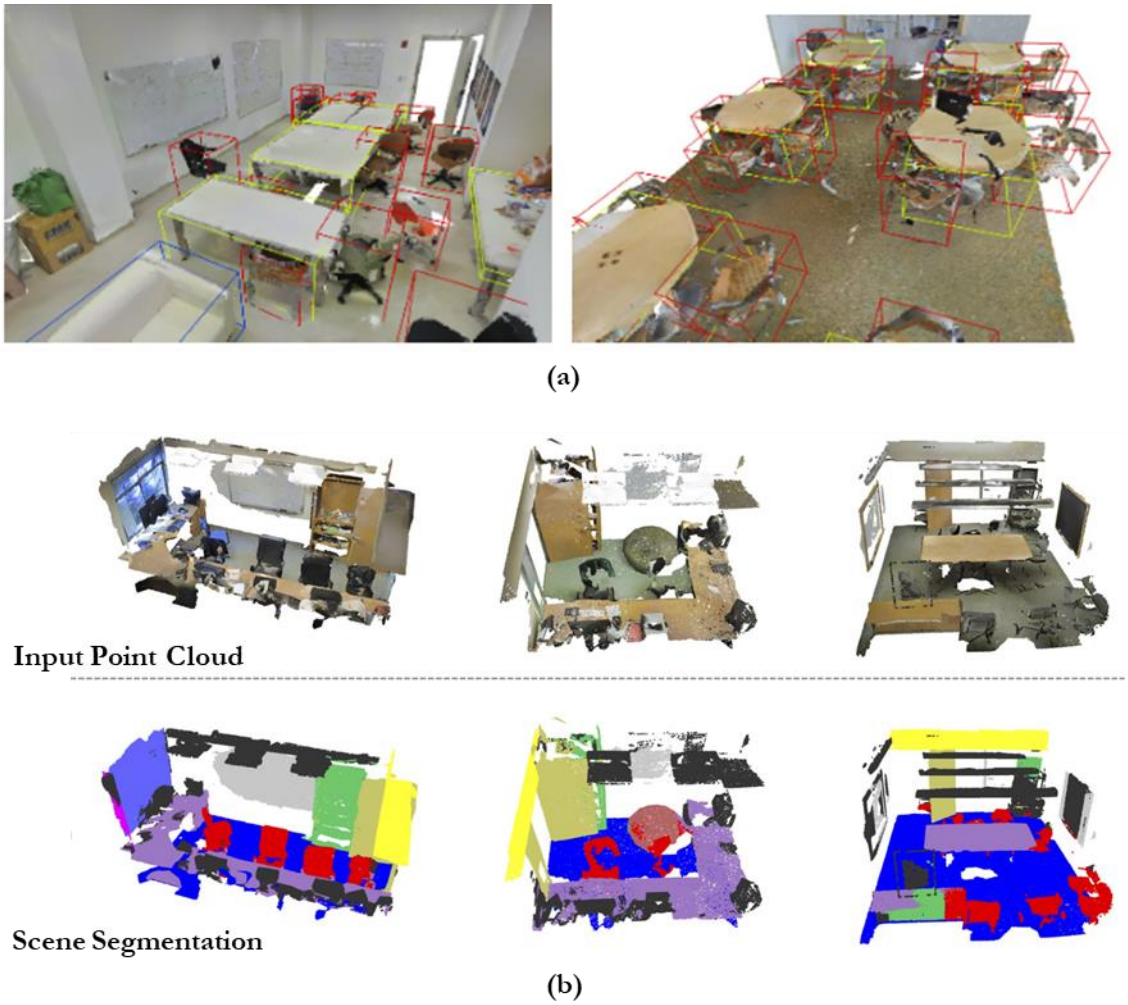


Figure 1.4: Deep learning methods to identify objects in indoor spaces with 3D point clouds: (a) Object detection - yellow bounding boxes for tables and red bounding boxes for chairs (b) Semantic segmentation - points belonging to different semantic categories represented by different colors. (Image source: Qi et al., 2017a; Rukhovich et al., 2021).

For this research, semantic segmentation would identify multiple safety-related assets present within buildings along with details of other structures like the ceiling, wall, and floor. Thus, the obtained results would be contextually more beneficial for asset management systems to model the safety infrastructure and explore the spatial relationships between identified safety-related assets and the scene. Hence, we choose the scene segmentation method to identify assets for the asset management system. Also, with the recently emerging concept of open science, many DL networks for scene segmentation with state-of-the-art performance are available as open source. These methods can be adapted for customized applications and datasets. In this context, the current research explores the scope of using an existing DL scene segmentation network to identify safety-related assets in buildings using 3D point clouds.

1.3. Research Problem

As described in Section 1.1, the critical component of setting up an asset management system for safety-related assets is identifying them within a building. The 3D point clouds offer rich spatial and contextual information for DL methods to learn discriminative features of safety-related assets to identify them. However, it is not straightforward to use DL to identify these safety-related assets due to their inherent characteristics within a building that create various challenges, summarized in Figure 1.5.

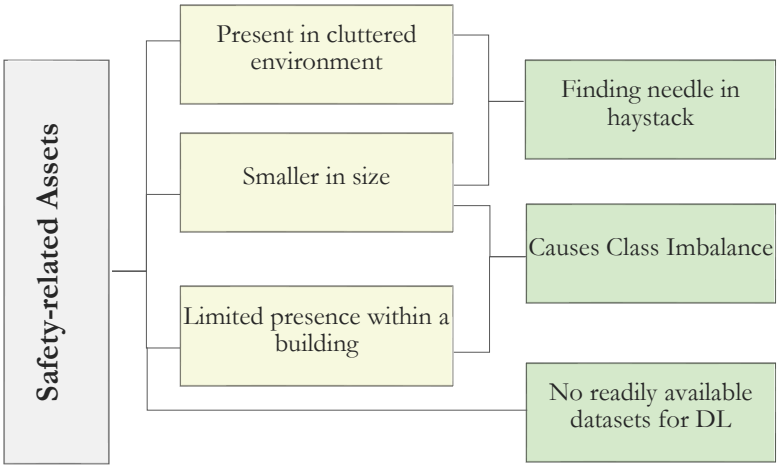


Figure 1.5: Features of safety-related assets (yellow blocks) and the corresponding challenges (green blocks) (Image source: author).

Indoor scenes of buildings consist of different structures like walls, floors, and ceilings and are cluttered with various objects like furniture and electrical equipment (Nikoohemat et al., 2020). Figure 1.6 shows the point cloud of a cluttered scene with various pieces of furniture. In Figure 1.6(a), it can be noticed that the fire switch in the yellow bounding box is smaller than other objects and structures present in the scene. Processing such a point cloud to find the safety-related asset (fire switch) would resemble "*finding a needle in the haystack.*" In the case of a building, the entire building's point cloud as the search space would be the haystack, and some small-sized safety-related assets to be identified as the needles, making it difficult to segregate and identify such assets.

Figure 1.6: Point cloud of a room (a) Front view - fire switch (small safety asset) in yellow bounding box; (b) Top view (ceiling removed) – Cluttered scene with furniture and other pieces of equipment (Image source: Armeni et al., 2016).

Ren and Sudderth (2018) reduced the search space to find smaller objects by first identifying big objects like tables, which function as supporting surfaces, to smaller objects like lamps as they are placed above the big objects. However, this logic cannot be directly applied to identify small-sized safety-related assets, as they

are independent and standalone objects (Chen, 2019; Hossain et al., 2021). However, most safety assets have a fixed placement in a building. For example, the exit or escape signs are closer to the roof, and the ventilation systems are present in the ceiling. Therefore, we can utilize this prior information on safety-related asset placement locations to reduce search space to identify small-sized assets efficiently.

Another setback is caused due to the limited occurrences of safety-related assets compared to other objects or structures in the building. Let us consider the room in Figure 1.6 again. In the displayed point cloud, the points representing walls and furniture exist in much higher numbers than the small fire switch in the yellow bounding box. The overall number of points representing the fire switch here is small due to (i) the smaller physical size and (ii) the limited number of instances present in the room. This problem applies to other safety-related assets like exit signs, fire extinguishers, fire sprinklers, and smoke detectors whose total counts in a building are limited. If we consider the point cloud of the entire building, the obtained data would then be "*class-imbalanced"* with the majority and minority classes (Johnson and Khoshgoftaar, 2019). In our case, classes like permanent structures (ceiling, floor, and wall) and furniture would fall under the majority class, and the safety-related assets would be the minority class. From the point distribution of an indoor point cloud dataset by Stanford University in Table 1.1, it is noticeable that the classes in red cells have a lesser number of points than other semantic classes, making them the minority class.

Table 1.1: Point distribution based on semantic classes in Stanford's indoor point cloud dataset (created by author).

Semantic Class	Total Points	Semantic Class	Total Points	Semantic Class	Total Points
Ceiling	51680391	Exit Sign	75683	Fire Switch	45127
Floor	45207796	Stairs	598622	Fire Extinguisher	164147
Wall	89069836	Door	13072333	Ventilation	1682344
Furniture	40557078	Clutter	18933074	Light	5308726
Window	6891880				

When dealing with class-imbalance data, Anand et al. (1993) explain that for DL, the majority set in the class-imbalanced data dominates the process of updating the model weights during the learning or training phase. In such cases, the error of the majority class is reduced rapidly during the initial learning stages compared to the error of the minority class, resulting in slow convergence of the DL network (Anand et al., 1993; Johnson and Khoshgoftaar, 2019). In our case, any DL method would take longer to learn to identify the safety-related assets (minority classes). Also, it would exhibit imbalanced efficiency among classes as the data to train for minority classes are not as abundantly available as majority classes. Hence, developing an efficient methodology to manage class-imbalanced data is necessary for identifying safety-related assets.

1.3.1. Innovation

Many DL methods for scene segmentation with state-of-the-art performance are open-source and can be adapted for customized 3D point cloud datasets. However, existing studies for indoor scene segmentation using readily available DL networks like PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b) focus on permanent structures and big objects. Recently, Hossain et al. (2021) experimented with the PointNet++ to identify safety-related assets in buildings using point clouds. But their experiment failed to correctly identify small assets like fire sprinklers and fire alarms, forcing them to use images first to identify assets and then transfer the results to point clouds. So, it is of interest to revisit the idea of utilizing DL scene segmentation with a new starting point and with another novel method to identify safety-related assets using 3D point clouds.

Additionally, DL methods are data-driven and require extensive labeled datasets for abstraction and generalization (Guo et al., 2021; Song et al., 2015). However, DL on 3D point clouds faces the challenge of having limited labeled datasets (Guo et al., 2021). In addition, even though the existing labeled datasets like ShapeNet Part (Yi et al., 2016) and Stanford indoor dataset (Armeni et al., 2016) contain a variety of indoor objects, they do not include most safety-related assets as readily available labeled classes. Therefore, these datasets are not immediately helpful in training a DL network to identify safety-related assets (Hossain et al., 2021). Hence, we must prepare datasets for safety-related assets by processing and labeling them to train and evaluate the DL methods.

1.4. Research Objectives And Questions

1.4.1. Main Objective

The research aims to explore the scope of using the DL scene segmentation approach with 3D point clouds to identify safety-related assets for asset management systems. The indoor 3D point clouds used as input are huge with high point density, while some safety-related assets are small. Hence, the problem resembles *"find-the-needle-in-a-haystack,"* where a 3D point cloud (haystack) must be analyzed to find various small-sized safety-related assets (needles). In this regard, an existing DL network and relevant open-source point cloud datasets are adapted to identify the safety-related assets of interest by performing scene segmentation. The resulting outputs are the semantic layout of buildings with safety-related assets categorized into semantic classes.

1.4.2. Sub-Objectives with Research Questions

- **To adapt a DL network for identifying safety-related assets in an indoor scene.**
  1) How should the chosen DL network be modified for identifying safety-related assets in an indoor scene?
  2) What is a suitable strategy to handle the class imbalance problem for safety-related assets and to effectively assist their feature learning for DL?
  3) How can prior knowledge of safety-related assets be used to support their identification?

- **To assess the acceptability of the designed methodology for safety-related asset management.**
  4) How accurately does the designed methodology identify the safety-related assets?
  5) How well does the method generalize on unfamiliar point cloud datasets?
  6) Which evaluation metrics are most suitable for asset management with the obtained results?
  7) What is the acceptable point cloud resolution to identify safety-related assets with the designed methodology?
  8) Is the designed methodology robust to data generated from different 3D sensor systems?

1.5. Thesis Structure

This thesis consists of seven chapters. We introduce the research by elaborating its background, relevance, and objectives in Chapter 1. In Chapter 2, we review the concepts and earlier DL works relevant to this research from the literature. The datasets with software and hardware tools used to conduct this research are presented in Chapter 3. We explain the designed methodology to achieve the research objectives in detail in Chapter 4. In Chapters 5 and 6, we present the research findings in the form of results obtained, discussions, and critical analysis. Finally, we conclude the research with answers to the research questions and future recommendations in Chapter 7.

2. LITERATURE REVIEW

We briefly review existing DL methods and open-source 3D point cloud datasets for indoor scene segmentation in Section 2.1 and Section 2.2, respectively. In Section 2.3, we describe the DL network used in this research, Kernel Point - Fully Convolutional Neural Network (KP-FCNN).

2.1. Deep Learning For Indoor Scene Segmentation

Scene segmentation is a process where each point in a point cloud is assigned to the corresponding semantic category in a scene (Guo et al., 2021; Liu et al., 2019). Point clouds are inherently irregular, unordered, and unstructured (Bello et al., 2020; Guo et al., 2021). Existing DL scene segmentation methods can broadly be grouped into indirect and direct DL methods based on how DL networks consume and process them. Figure 2.1 summarizes the categories of DL networks for scene segmentation. All the DL works discussed and reviewed in this section are based on their implementation for indoor scene segmentation.

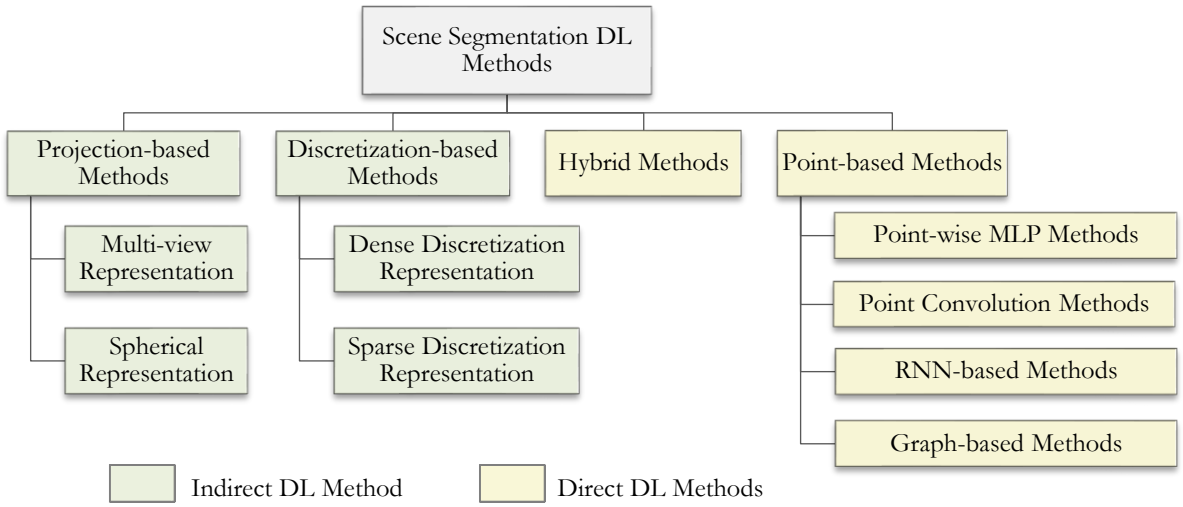


Figure 2.1: Taxonomy of DL methods for 3D scene segmentation; Here, MLP stands for Multi-Layer Perceptron, and RNN is Recurrent Neural Networks (Image source: author, based on Guo et al. (2021)).

2.1.1. Indirect DL Methods

Indirect methods transform a point cloud into structured data by projecting into 2D images or 3D volumetric representations using voxels before processing them (Bello et al., 2020).

Projection-based Methods: These DL methods project the 3D point clouds into multi-view and spherical 2D images to perform scene segmentation. They use multi-viewpoints of the camera to project the point cloud onto a 2D plane to generate synthetic images for different views (Bello et al., 2020). Later, pixel-wise predictions on these images for each view are combined to assign a semantic category per point. Few methods use spherical representations to retain more information in the projected data than the single view method.

Discretization-based Methods: These DL methods convert 3D point clouds into dense or sparse discrete representations such as volumetric voxels (Figure 2.2) or permutohedral lattices. SEGCloud (Tchapmi et al., 2017) is an end-to-end DL network that uses a voxelized point cloud with 3D fully convolutional neural

network (CNN). A CNN is a specialized feedforward neural network that processes multi-dimensional and grid-like data (Goodfellow et al., 2016, p. 326; Johnson and Khoshgoftaar, 2019). SEGCloud uses CNN to produce coarse labels on voxelized point clouds, later mapped to the original point cloud using an interpolation layer. Finally, fully connected Conditional Random Fields produce semantic labels.

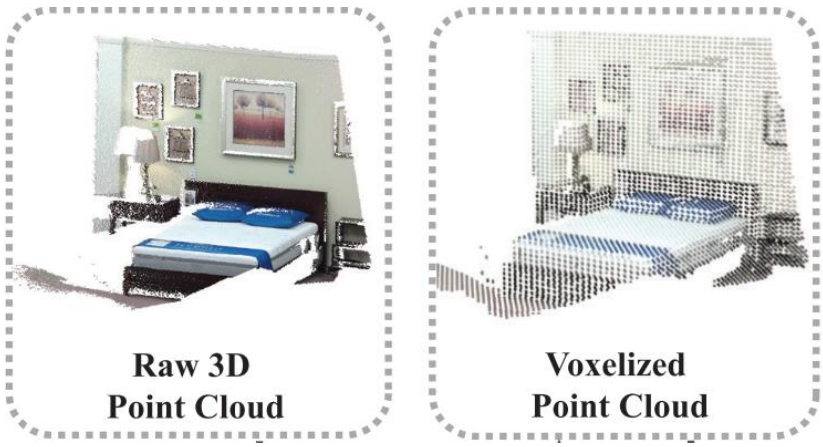


Figure 2.2: An indoor point cloud converted into voxel representation (Image source: Tchapmi et al., 2017).

The performance of projection-based methods for indoor scene segmentation is sensitive to viewpoint selection, occlusions in the scene caused by clutter, and density variations (Guo et al., 2021; Thomas et al., 2019). For discretization-based methods, deciding the appropriate grid size for a voxel is tricky, as low-resolution data loses minute details, and high-resolution data demands high memory and computational efficiency. In addition, both these methods experience data loss due to projections and volumetric transformations, and using intermediate representations causes discretization errors (Guo et al., 2021).

2.1.2. Direct DL Methods

Direct DL methods include hybrid and point-based methods that process point clouds without data conversions like indirect methods, hence no spatial information loss. Recently, there has been more focus on these methods for scene segmentation, especially when dealing with cluttered indoor environments. Figure 2.3 shows a chronological outline of some relevant hybrid and point-based DL methods implemented for indoor scene segmentation.

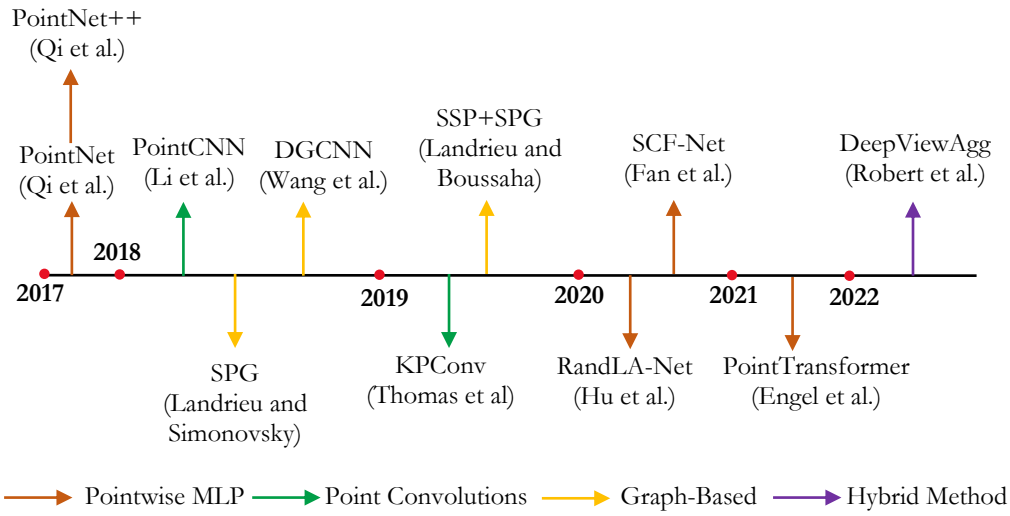


Figure 2.3: Chronological outline of some relevant direct DL methods for indoor scene segmentation (Image source: author).

Hybrid Methods: These methods learn multi-modal features to leverage all the available information from the 3D scans to perform scene segmentation. Multi-view PointNet (MVPNet) by Jaritz et al. (2019) and DeepViewAgg (Robert et al., 2022) methods integrate and utilize information from 2D images and 3D point clouds as a multi-modal approach to perform scene segmentation.

Point-based Methods: These methods work directly on 3D point clouds.

i. **Point-wise Multi-Layer Perceptron (MLP) Methods:** MLPs, also known as feedforward neural networks, are the standard DL model implementation (Goodfellow et al., 2016, p. 164). As illustrated in Figure 2.4, MLPs use single or multiple hidden layers to process and learn data features to achieve various levels of abstraction (Liu et al., 2019; Bello et al., 2020).

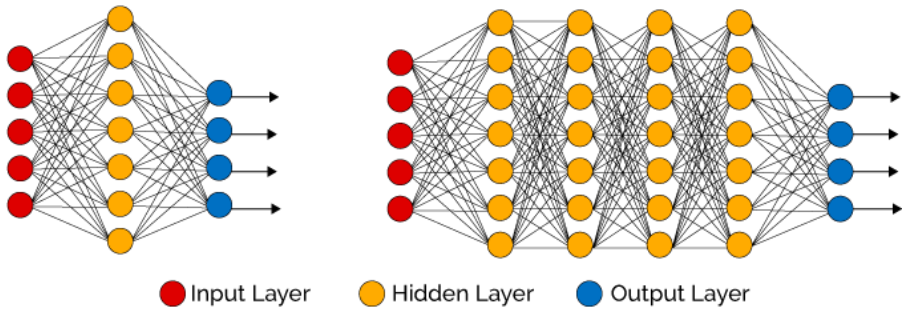


Figure 2.4: Illustration of shallow MLP (left) and deep MLP (right), based on the number of hidden layers (Image Source: Vázquez, 2017).

Pointwise MLP-based DL methods use a shared MLP as a basic unit to extract point-wise features and pooling functions to learn global features. PointNet (Qi et al., 2017a) learns point-wise features using shared MLPs directly and independently from the point cloud and later uses a symmetric pooling function to achieve global features and permutation invariance. However, shared MLP-based feature learning fails to capture local geometry and the context of a point with its neighbors (Qi et al., 2017a).

Later, methods based on different pooling mechanisms were proposed to capture and learn rich local structures. PointNet++ (Qi et al., 2017b) uses neighbor feature pooling to group the points progressively and hierarchically learn features. Some networks like Point Transformer (Engel et al., 2021) and Spatial Contextual Features Network (SCF-Net) by Fan et al. (2021) use attention-mechanism (Vaswani et al., 2017) based aggregation method. RandLA-Net (Hu et al., 2020) captures complex local structures through a local feature aggregation unit.

ii. **Point Convolution Methods:** These methods use unique and efficient convolution operators for point clouds rather than the regular grid convolution. PointCNN (Li et al., 2018) is a DL network that uses PointNet-like MLPs with k-neighborhood points and $\mathcal{X}$-transformation where $\mathcal{X} = MLP(point\ 1, point\ 2, \dots, point\ k)$ to model the local structures and then a normal convolution is applied on these $\mathcal{X}$-transformed features. Later, Thomas et al. (2019) proposed Kernel Point Convolution (KPConv), a point-wise convolution operator with a variable number of kernel points, supporting deformable convolution. Kernel Point Fully Convolutional Network (KP-FCNN) is a DL network based on the KPConv block for scene segmentation.

iii. **Recurrent Neural Networks (RNN) based Methods:** These DL methods use RNNs to capture contextual information from 3D point clouds to perform scene segmentation. RNNs are an extended version of MLPs with feedback connections (Goodfellow et al., 2016, p. 164). Liu et al. (2017) proposed an RNN-based DL network with a CNN and Deep Q-network (DQN) for

semantically segmenting large-scale 3D point clouds. Here, the CNN performs feature learning through color and spatial distribution of the points as input features, DQN localizes the objects as per the semantic classes, and RNN obtains segmentation results using the final feature vector (Guo et al., 2019).

iv. **Graph-based Methods:** These methods capture local geometric structures and shapes of objects in 3D point clouds by constructing a local neighborhood graph with every point as a node, as seen in Figure 2.5. Wang et al. (2019) proposed Dynamic Graph CNN (DGCNN), which utilizes local neighborhood information to learn shapes and multi-layer feature spaces to capture semantic properties from dynamically computed graphs. Some other scene segmentation networks proposed were based on Superpoint Graphs (SPG) like SPG by Landrieu and Simonovsky (2018) and Super Point-SPG (SSP+SPG) by Landrieu and Boussaha (2019). Here, a point cloud is first divided into geometrically similar elements (Superpoints), which are used to construct SPGs and then processed with graph convolutions to be segmented into semantic classes.

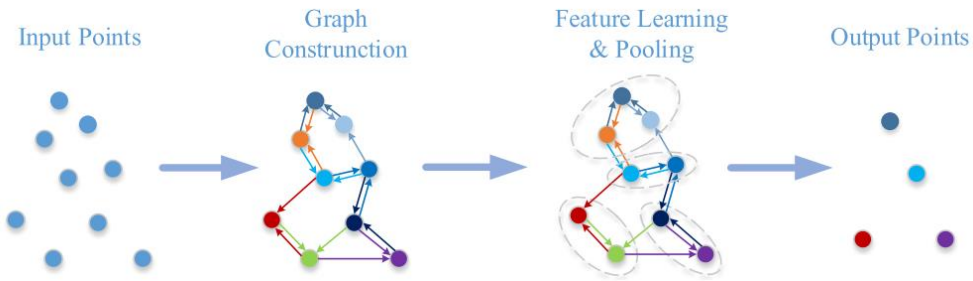


Figure 2.5: Illustration of workflow for graph-based DL network (Image source: Guo et al., 2021).

2.2. Open-Source 3D Point Cloud Datasets

In recent years, universities and industries have published many 3D indoor point-cloud datasets to compare the findings of different approaches for various applications. These open-source datasets are of both virtual and real scenes and provide vast ground truth data and labels, which help train the DL networks. Table 2.1 lists some open-source point cloud datasets available for 3D indoor scene segmentation research, based on the review by Bello et al. (2020) and Guo et al. (2021).

Table 2.1: Open-source 3D indoor point cloud datasets for scene segmentation (created by author).

Name	Developed By	Acquisition technique	Semantic Labels	Coverage Area
SUN3D	Princeton University (Xiao et al., 2013)	SFM	Yes	254 spaces in 41 buildings
Stanford 3D Large-Scale Indoor Spaces (S3DIS)	Stanford University (Armeni et al., 2016)	Depth camera	Yes	271 rooms in 6 areas
MultiSensor Indoor Mapping and Positioning Dataset	Xiamen University (Wang et al., 2018)	Backpack MLS	Yes	4 scenes
Human POSEitioning System (HPS) 3D scenes	Tübingen University (Guzov et al., 2021a)	Trolley MLS	No	6 Scenes

Among the datasets presented in Table 2.1, S3DIS is a benchmark and one of the most popular open-source datasets used to evaluate DL methods developed in indoor scene segmentation. Figure 2.6 shows the performances of some of the direct DL methods discussed in Section 2.1.2 with S3DIS.

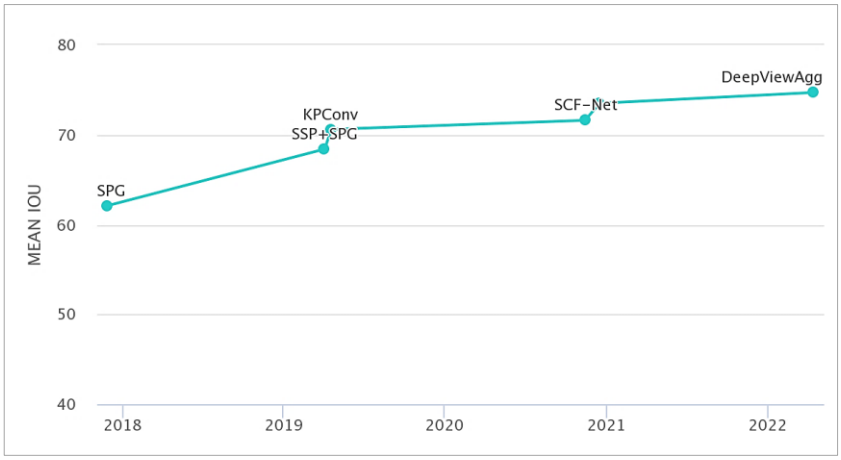


Figure 2.6: Benchmark performances on S3DIS for direct DL methods developed in 2018-2022 for indoor scene segmentation with mean Intersection over Union (IoU) in % (Image source: Papers With Code, 2022).

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

One of the requirements for this research is adapting an open-source DL method suitable for handling various research problems explained in Section 1.3. We chose the KP-FCNN by Thomas et al. (2019), a state-of-the-art DL method for indoor scene segmentation. It is a point-based method using an effective point convolution operator, KPConv, illustrated in Figure 2.7.

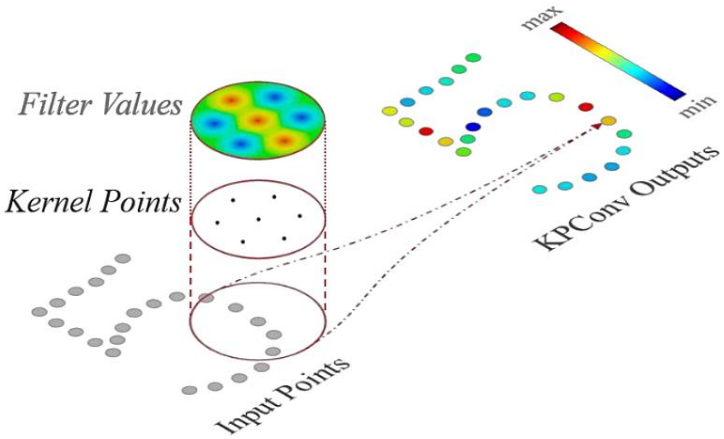


Figure 2.7: Illustration of KPConv. Input points (shown in grey) are convolved through kernel points (in black) with filter weights on each point where the area of influence of these weights is defined by a linear correlation function (Image source: Thomas et al. (2019).

Point clouds are inherently irregular, with sparse and dense regions, which is one of the challenges of using point clouds directly with DL methods (Bello et al., 2020; Guo et al., 2021; Liu et al., 2019). Additionally, the overall densities of point clouds change with the varying acquisition techniques and sensors. For example, the SFM-generated point clouds have low and sparse point densities, but Terrestrial Laser Scanner (TLS) and MLS systems provide a high-resolution point cloud (Lehtola et al., 2017; Liu et al., 2019). DL networks learn data representations using features of objects, and for 3D point clouds, such learning is sensitive to variation in point density (Thomas et al., 2019). Therefore, we need a network to efficiently manage this

varying point density within point clouds irrelevant to the acquisition methods. KP-FCNN can manage the problem of varying point densities by using the grid subsampling and radius neighborhoods strategy, making the convolution more robust and reducing the computational cost (Thomas et al., 2019).

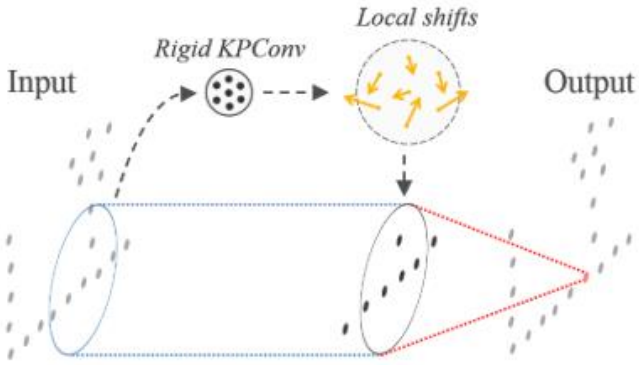


Figure 2.8: Illustration of deformable KPConv showing local shifts on the kernel points (Image source: Thomas et al. (2019).

Furthermore, deformable KPConv, as illustrated in Figure 2.8, allows the kernel points to adapt their shape based on the local geometry, suitable for the unstructured point clouds. Thomas et al. (2019) also noticed that deformable KPConv performs well for large and varied indoor datasets like S3DIS and improved network adaptability to the geometry of objects in the scene. Additionally, KPConv offers more flexibility by not restricting the number of kernel points, unlike other convolution methods with a fixed number of kernel points. Using the deformable KPConv, KP-FCNN with S3DIS achieved a mean IoU of 67.1% and 70.6% for Area-5 and K-fold tests, respectively. It also achieved a mean class recall of 72.8% for Area-5 and 79.1% for K-fold tests with S3DIS.

As explained in Section 1.3, class imbalance in the input point clouds for safety-related assets is one of the research problems. It affects feature learning in DL methods, making it a significant challenge to perform indoor scene segmentation for smaller safety-related assets directly. By design, KP-FCNN allows picking the training datasets as small spherical input clouds across the scenes based on either random or regular picking strategies. The **random picking** method arbitrarily chooses several sphere centers balanced by the semantic classes in the dataset, picking the same number of spheres centered on an object for each class. However, the **regular picking** method chooses input spheres consistently across the dataset, ensuring spatial regularity. Therefore, the random picking strategy offered by KP-FCNN partially tackles our research problem of class-imbalanced data. This strategy allows the network to see the minority classes more often, inherently improving the feature learning of minority classes, in our case, safety-related assets.

The above-discussed network features motivated us to choose this DL network for the current research. Additionally, the code implemented for this network by Thomas Hugues (the author) is an open-source and well-maintained repository with sufficient description for customizing its usage. As the network achieves state-of-the-art performance, it is also widely adapted for various other tasks by other users, with active community participation, providing an added advantage to adapt the network for the current research.

3. DATA AND TOOLS

This chapter includes a brief analysis of the 3D point cloud datasets and the tools used in this research under Sections 3.1 and 3.2, respectively.

3.1. Data

3D scanning technologies for indoor environments have been expanded into devices like depth cameras, MLS devices, and handheld smartphones with lidar (Díaz-Vilariño et al., 2022). For this research, we use colored 3D point cloud datasets of buildings obtained from these different 3D sensors, as listed in Table 3.1. We choose S3DIS and HPS datasets from the open-source 3D point cloud datasets listed in Table 2.1 and create a new dataset, iPhone data, using smartphone lidar. They contain various safety-related assets like ceiling lights, exit signs, doors and windows, ventilation air ducts, temperature controllers, smoke detectors, fire alarms, sprinklers, and extinguishers. Semantic labels for the HPS dataset for these assets are not available; however, the S3DIS dataset was already partially labeled for some safety-related assets.

Table 3.1: Details of point cloud datasets used in the research.

Name	Sensors	Device Used	Developed By	Total Scan Areas	Points (In millions)	Labels For Safety-related Assets
S3DIS	Depth Cameras	Matterport Camera	Stanford University (Armeni et al., 2016)	6	273M	Partially available; Remaining assets labeled manually
HPS	lidar + Camera	NavVis M6 Mobile Laser Scanner	Tübingen University (Guzov et al., 2021a)	6	500M	All assets manually labeled
iPhone Data	lidar + camera	Smart Phone	CGI Inc (Author, 2022)	4	2.41M	Not labeled

3.1.1. Stanford 3D Indoor Scene Dataset (S3DIS)

S3DIS (Armeni et al., 2016) is a benchmark dataset from three distinct educational and office buildings covering six large-scale indoor areas, namely Areas 1-6. Architecturally, Areas 1, 3, and 6 and Areas 2 and 4 are similar, whereas Area-5 is captured from a different building than other areas. The dataset provides geometry and color attributes along with semantic labels from the categories: ceiling, floor, wall, beam, column, stairs, window, door, table, sofa, board, bookcase, chair, and clutter.

Initially, 3D textured meshes of the scanned area were reconstructed using RGB and depth images captured from the Matterport depth camera. These reconstructed 3D meshes were densely and uniformly sampled and were assigned the corresponding colors to generate the colored 3D point clouds (Armeni et al., 2017). The point cloud density in this dataset is consistent across the scenes. However, the point clouds contain noise and are less accurate than those acquired with laser scanners (Lehtola et al., 2017; Thomas, 2019). Nevertheless, the data is adequate for scene understanding and 3D indoor modeling tasks through semantic segmentation (Nikoohemat et al., 2018). Figure 3.1 shows three indoor scenes from the S3DIS dataset.

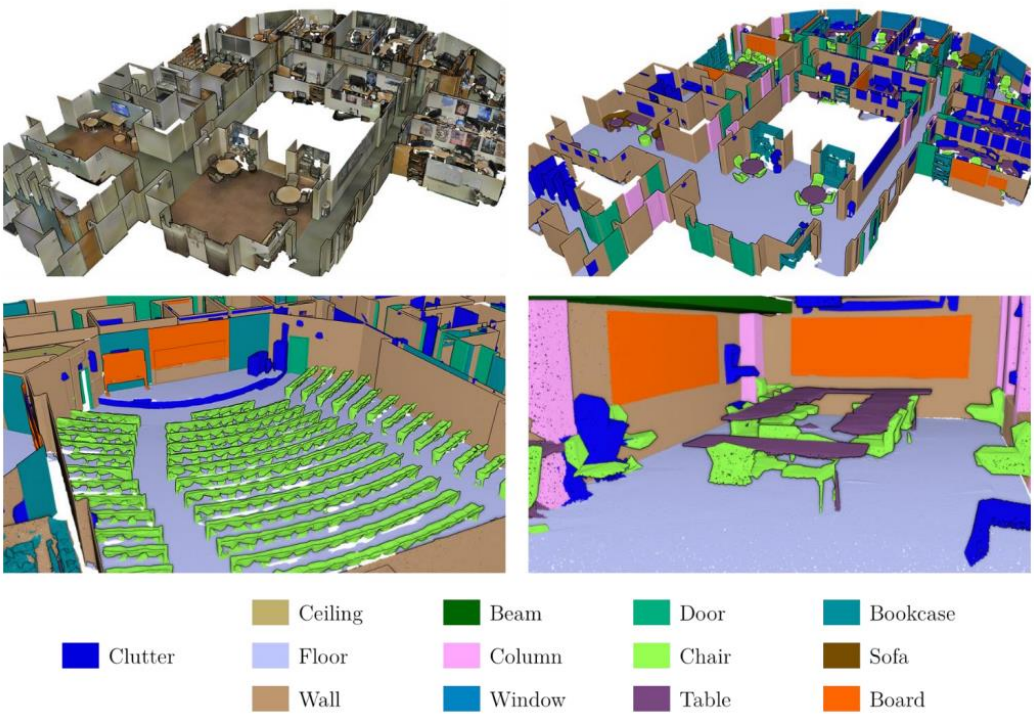


Figure 3.1: Screenshots of indoor scenes from the S3DIS dataset (Image source: Thomas, 2019).

3.1.2. Human POSEitioning System (HPS) Dataset

HPS dataset was compiled by Guzov et al. (2021) for other 3D indoor environment-related research purposes. It is a collection of multiple datasets containing 3D scene scans, videos from head-mounted cameras, and Inertial Measurement Unit (IMU) poses. The 3D scenes in this dataset were acquired using a commercial trolley MLS device, NavVis M6 (NavVis, 2022), shown in Figure 1.3. It uses four lidar sensors and six RGB cameras to reconstruct a scene using the Simultaneous Localization and Mapping (SLAM) algorithm (Guzov et al., 2021b). The point clouds obtained are in Polygon File Format (PLY) with geometry, colors, surface normals, curvature, and camera-specific attributes. The dataset contains eight large-scale 3D scans, with two outdoor and six indoor scans. All the indoor scans cover large working spaces cluttered with objects like furniture and other equipment, as seen in Figure 3.2.

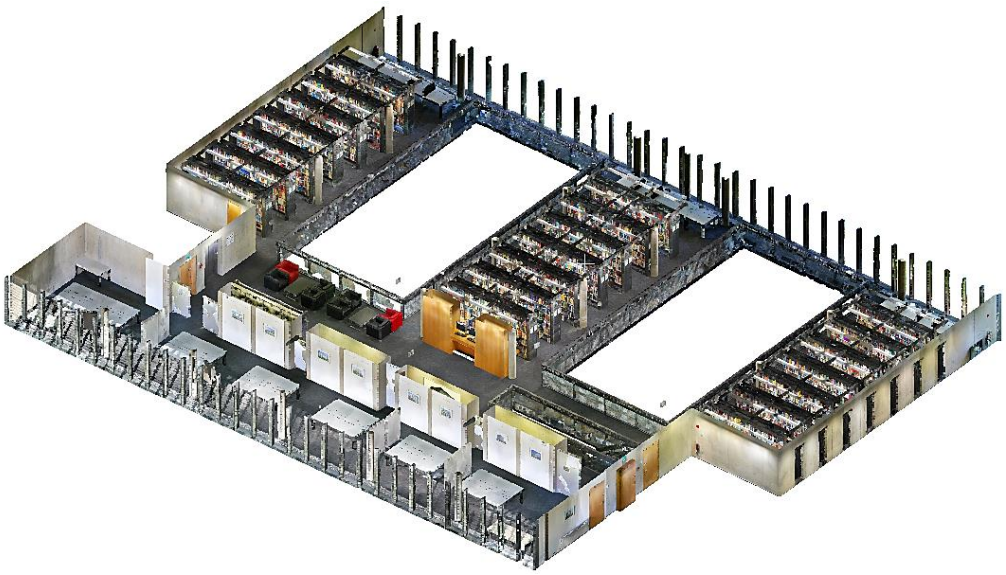


Figure 3.2: An indoor scene from the HPS dataset (ceiling is removed to show the layout).

We use the six indoor scans, namely, MPI_BIBLIO_UG, MPI_BIBLIO_EG, MPI_Etage6, MPI_KINO, MPI_BIBLIO_OG, and MPI_EG; from now on, referred to as HPS scans numbered 1-6. Out of all the attributes, we use geometry and color information for the current research. Since the 3D indoor scans in this dataset were not prepared for scene segmentation or other point-wise operations, the point clouds are not cleaned or labeled into semantic categories. Figure 3.3 shows an example of a raw scan available in the dataset, showing partially captured structures above a building due to occlusion during data acquisition. Therefore, for this research, we need to process these scans to remove outliers, define a scene's extent, and label them.

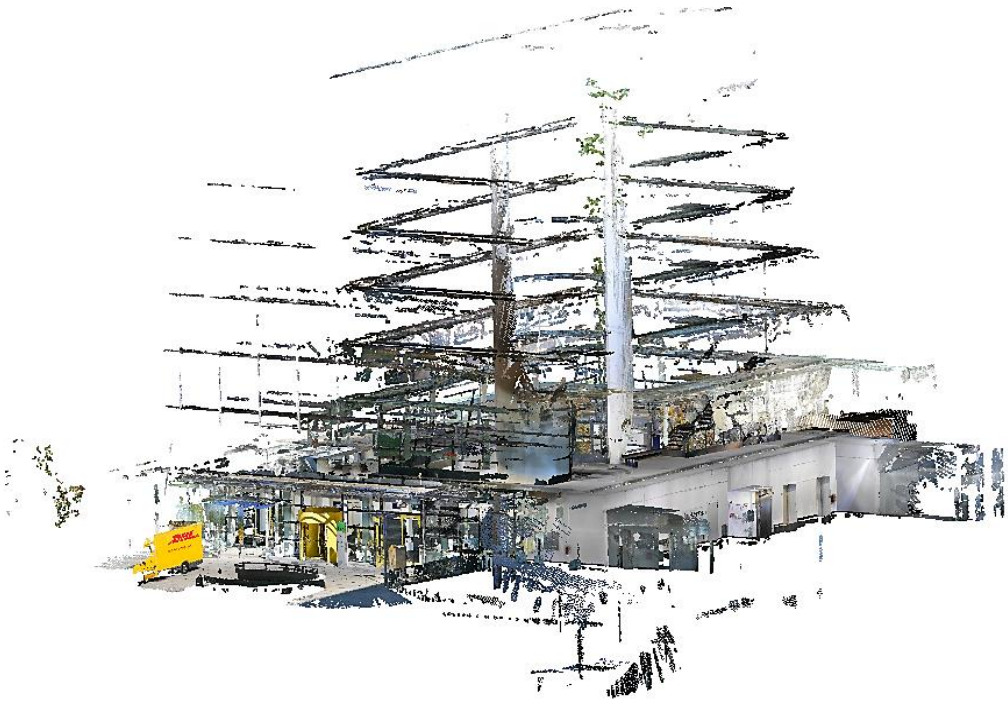


Figure 3.3: Raw point cloud of a building from the HPS dataset.

3.1.3. iPhone Data

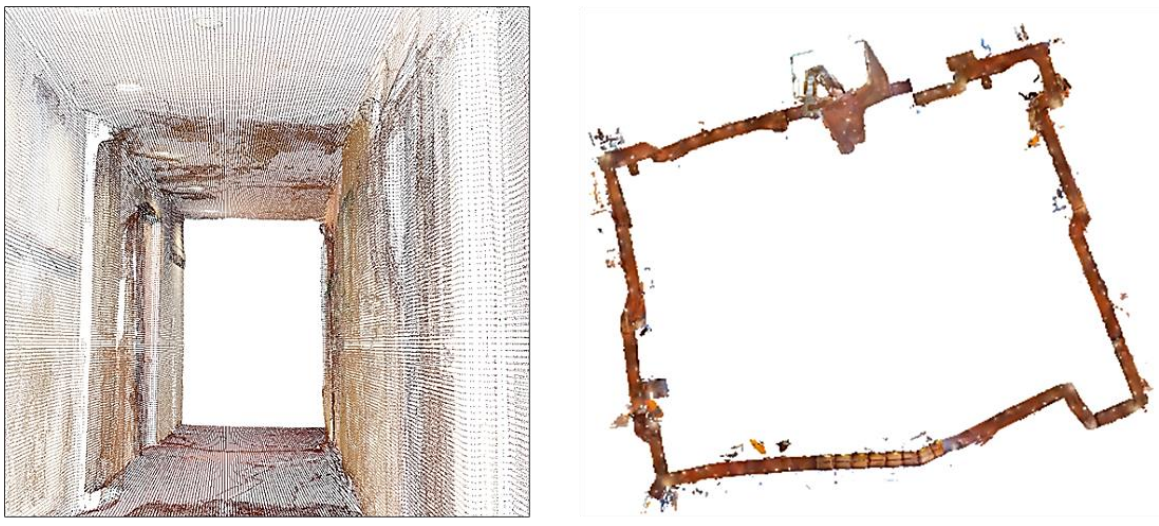
In collaboration with CGI Inc., we established a new dataset for this research acquired with an iPhone 12 Pro (hereafter iPhone) embedded lidar as a measurement device. iPhone, a smartphone with a lidar scanner, was introduced by Apple Inc. (Apple, 2020), presenting a hand-held consumer-grade laser scanning option. The device has a 6.1inch display with a 12MegaPixel camera and a lidar scanner with a maximum range of up to 5 meters. Applications like 3D Scanner, Polycam, Scaniverse, and EveryPoint can quickly scan objects and rooms in buildings in 3D and export them as point clouds.

We performed various scans using the Scaniverse mobile application by walking with a hand-held iPhone at the Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, The Netherlands. The scans were curated by carefully covering every scene angle, especially the safety-related assets. Scaniverse combines distance measurements from lidar and colors from the images captured from the camera to generate a high-resolution colored 3D mesh. The resulting data was then exported into 3D point clouds of PLY file format. The built-in crop feature in the application enabled dynamically adjusting the extent of the scan along all three axes. The effective range of the lidar scan was set to 5 meters. We obtained seven scans using the iPhone covering the room, hallway, and lobby areas. For this research, we use one room, two hallways, and one lobby scans to identify safety-related assets. Figure 3.4 shows scans of a room and hallway from the ITC building.

Figure 3.4: Room with ceiling removed (left) and hallway (right) scan of ITC building using iPhone.

The lidar sensor in iPhone uses a 24x24 grid of infrared dots projected into the scene and measures the flight time to convert them into distances to produce a depth map (Luetzenburg et al., 2021). Therefore, the points in this dataset are placed regularly in a grid, as shown in Figure 3.5(a), with an average of 2-centimeter point spacing. Also, the scans here are of lower quality and resolution than the scans of the S3DIS and HPS datasets.

Based on firmware code property, iPhone lidar scans limit the maximum number of points per scan, so the point density decreases with an increase in the extent of the scan area. We verified this principle during the collection of the current dataset. For example, a small room scan in Figure 3.4 had 711,251 points, but the point density was poor when a whole floor, as shown in Figure 3.5(b), was scanned at once, consisting of 491,759 points. The reduced point cloud density also implies that the number of points representing small safety-related assets like exit signs and fire alarms also reduces drastically. Additionally, Díaz-Vilariño et al. (2022) verified through their experiments that iPhone lidar failed to generate good quality point clouds when captured for more than two rooms, suggesting smaller scans. Hence, we keep the scans short, covering smaller areas like rooms and limited sections of hallways.



(a) (b)

Figure 3.5: iPhone scan: (a) Point cloud showing a regular grid pattern; (b) Entire floor in the building.

3.2. Tools and Technologies

3.2.1. Hardware

All the processes were implemented on the remote high computing Linux server offered by the Faculty of ITC, University of Twente. The server consists of a processor with 64-bit architecture and 256 Giga Bytes (GiB) memory. Since the training and testing of the DL network require high computational memory, a Graphics Processing Unit (GPU) and Compute Unified Device Architecture (CUDA) for GPU acceleration were used. The server specifications with the Central Processing Units (CPUs) and GPU are in Table 3.2.

Table 3.2: Hardware details

Hardware	Count	Details
CPU	2	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
GPU	1	NVIDIA A40

3.2.2. Software

The current research is implemented in Python programming language, using the Python3.8 version. Open-source Python implementation for the chosen KP-FCNN DL network is available on GitHub with Tensorflow and PyTorch libraries. The present research uses the PyTorch implementation (Hugues, 2020). A Conda virtual environment is used for this research implementation, specifically with PyTorch 1.10.2, CUDA 11.3, and cuDNN 8.2.0. To connect remotely from the local computer to the server and run commands, we used the open-source Windows application MobaXterm (personal edition). It provides essential remote network tools like Secure Shell (SSH) protocol to provide a secure way to access the remotely connected computer. This application has a user-friendly interface allowing connections to multiple servers, uploading and downloading data from the local computer to the server and vice versa, and securely transferring files between servers.

Table 3.3 lists all the other essential libraries used during the research. Additionally, libraries like os, sys, glob, and shutil were used for managing the file paths along with the files, argparse for command-line interfaces, and tqdm for following the progress of loops. Visual Studio (VS) Code with Jupyter, an open-source code editor of version 1.67.1 on Windows, was used to edit the source code, calculate performance metrics, and generate confusion matrices for the results. Lastly, Cloud Compare (version 2.12), an open-source 3D point cloud processing software for Windows, is used to label, subsample, and crop the point clouds, calculate the geometric features, and visualize the scene segmentation results.

Table 3.3: Packages and libraries used for the research.

Library	Version	Application
Numpy	1.21.2	Numerical Operations
Scikit-learn	0.23.2	To interoperate with NumPy and SciPy Python libraries
PyYAML	5.3.1	Data interpreter, parser, and emitter for Python
Pandas	1.4.2	Data handling
Seaborn	0.11.2	Visualization
Matplotlib	3.3.1	

4. METHODOLOGY

In this chapter, we elaborate on the designed methodology illustrated in Figure 4.1 to identify safety-related assets in buildings using DL on 3D point clouds. We define the semantic classes used for this research in Section 4.1. Next, we describe the data preparation steps, including pre-processing and train-test data split for various experiments in Section 4.2. In Section 4.3, we present the implementation of One-shot and Stage-wise Methods for scene segmentation, referred to in Figure 4.1. Further, in Sub-sections of 4.3, we elaborate on the KP-FCNN network architecture, chosen parameters, data augmentation strategies, and adjustments to overcome class-imbalance problems. Lastly, we present the metrics used to evaluate the performance of the designed methodology for asset management in Section 4.4.

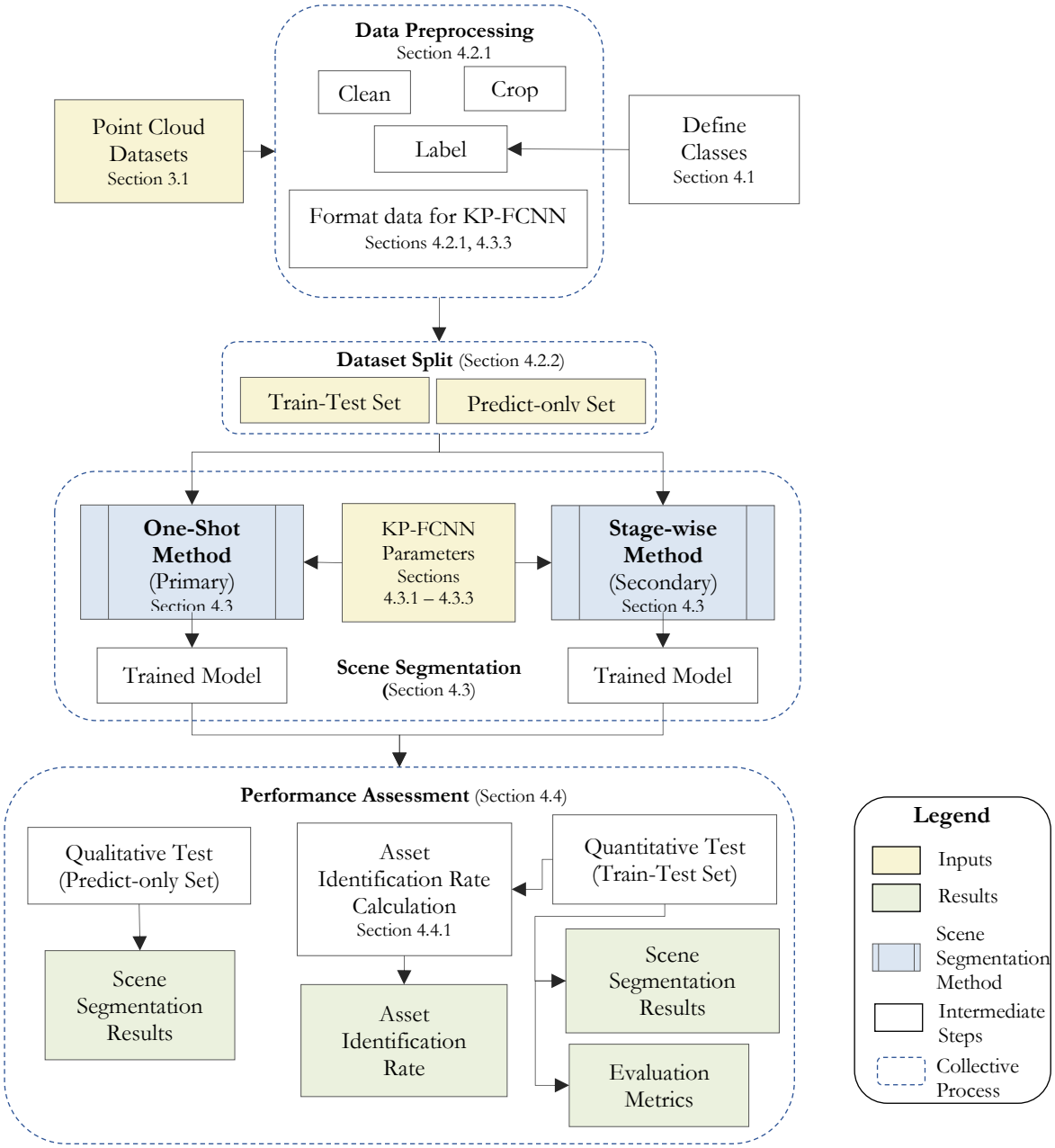


Figure 4.1: Overall workflow of the methodology. Here the One-shot method is the primary method used throughout the research. (Image source: author).

4.1. Semantic Classes

A building consists of multiple safety-related assets depending on the need and functionality, as listed in Table 4.1. However, every building may not have all these safety-related assets. Therefore, we identify a list of assets commonly found in most buildings, as mentioned in Table 4.1 (column 3). Hence, we choose eight asset classes for the current research: doors, windows, stairways, fire switches, fire extinguishers, exit signs, ventilation ducts, and ceiling lights. Since we also intend to provide complete scene semantics, we include other semantic classes like ceiling, floor, wall, furniture, and clutter. Here, the clutter class includes any object in the indoor scene that does not belong to the other chosen semantic classes. Therefore, we have 13 semantic classes, eight safety-related and five scene-semantics classes. Figure 4.2 shows some chosen safety-related assets present in all the datasets described in Section 3.1.

Table 4.1: List of safety assets based on functionality and chosen asset classes (Hossain et al., 2021; NAPSG, 2020).

Functionality	Safety-related assets	Chosen assets
Access	Doors, Stairway, Elevators, Escalators, Building Entrance-exit, Fire Escape Access, Roof Access, Windows, Fire Door	Doors, Windows, Stairways
Fire Suppression Features	Fire Hydrant, Fire Alarm, Firewall, Sprinkler, Fire Alarm Switch, Extinguisher, Smoke Detector	Fire Alarm Switch, Fire Extinguisher
Utility Shutoff	Electric, Gas, Water	-
Hazmat	Oxygen Cylinders	-
Other	Exit Sign, Stop Sign, Escape Route Sign, First Aid Kit, Ventilation Air Duct, Ceiling Lighting, Emergency Lighting	Exit Sign, Ventilation Duct, Ceiling Lighting

Figure 4.2: Screenshots of a few safety-related assets in the datasets used for this research: Area-wise S3DIS and iPhone dataset, namely (left to right), exit sign; fire switch; lights and ventilation ducts on the ceiling; wall-embedded fire extinguisher; HPS Scans, namely (left to right), ceiling lights, hand-useable fire extinguisher, fire alarm switch, exit sign, and ventilation duct (Image source: Armeni et al., 2017; Guzov et al., 2021a).

4.2. Data Preparation

We divide the datasets described in Section 3.1, as illustrated in Figure 4.3, based on their usage. We use the train-test set to assess the designed methodology quantitatively; hence they require labels. In contrast, the predict-only set is an unlabeled dataset used for qualitative assessment.

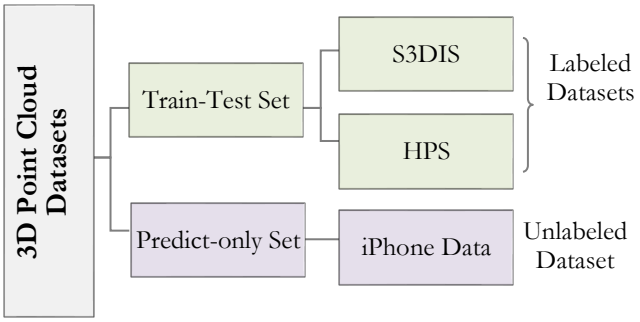


Figure 4.3: Datasets used for this research where green blocks represent labeled datasets, and purple blocks represent unlabeled datasets (Image source: author).

4.2.1. Pre-Processing

S3DIS is a pre-processed dataset and, thus, does not require additional data cleaning procedures. The point clouds in this dataset had labels with semantic categories: ceiling, floor, wall, beam, column, stairs, window, door, table, sofa, board, bookcase, chair, and clutter. Among these, doors, windows, and stairs are safety-related assets that we can readily use for this research. However, other safety-related classes listed in Section 4.1, like fire switches, fire extinguishers, exit signs, ventilation air ducts, and ceiling lights, require labeling. We manually label these assets using CloudCompare. First, we segment out the object of interest and save it as a text file with X, Y, Z, R, G, and B fields with the class label as the file name. Later these class labels are appended to the data as a new attribute in the pre-processing steps within the DL network.

Some semantic classes in S3DIS included a few safety-related assets. For example, the ceiling class had ventilation ducts and a few lights; the remaining lights and all exit signs were labeled clutter; fire extinguishers and switches were within the wall class. We regroup and label the original semantic classes into appropriate safety-related classes. We use a Python script to reformat the existing table, sofa, board, bookcase, and chair classes into one furniture class and beam and column classes into the wall class. Figure 4.4 summarizes the regrouping of S3DIS original classes for the safety-related asset classes chosen in Section 4.1.

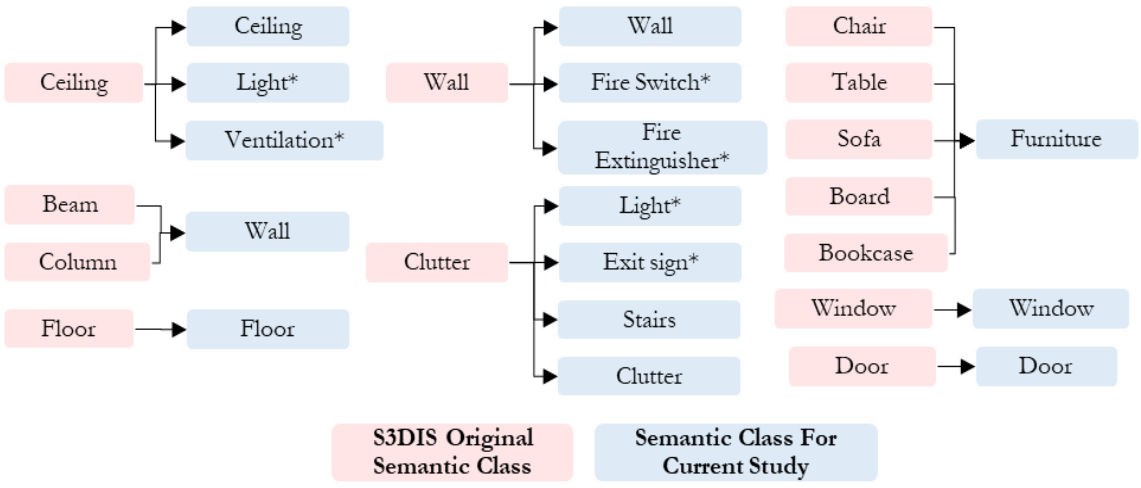


Figure 4.4: Regrouping original semantic classes of the S3DIS dataset (in pink) into classes for the current research (in blue). Here, we added the classes with *, and their point cloud data are manually labeled.

In contrast to the S3DIS dataset, the HPS dataset is characterized by large-scale raw point clouds with outliers and back reflections from the scene and is not labeled, as discussed in Section 3.1.2. Hence the point clouds from this dataset require pre-processing before we use them to perform scene segmentation. We partially remove reflections and outliers from 3D point clouds using CloudCompare's Statistical Outlier Remover (SOR) filter tool. It uses the average neighborhood point distances to reject the points farther than the average distance plus n-times the standard deviation. Table 4.2 shows the parameters used for the SOR filter. Later the segment tool is used to clean further and crop the filtered point cloud interactively to define the extent of a scene. Finally, we label all the scans manually using CloudCompare, as we did for S3DIS.

Table 4.2: Parameter values for pre-processing HPS dataset.

Operation	Parameter	Value
SOR Filter	Number of neighbors	20
	Standard deviation multiplier	1

The data from iPhone is cleaner and requires no rigorous pre-processing. CloudCompare's segment tool is used to crop the point cloud interactively to define the extent of a scan. Additionally, it is a predict-only dataset assessed qualitatively and does not require labeling.

4.2.2. Train-Test Data Split

We use the data division in Figure 4.3 to design data splits to perform various experiments in this research. We split the datasets into familiar and unfamiliar train-test categories, as shown in Table 4.3. Here, familiarity means the train-test areas for the network share some similarities with feature representations with architecture and objects (geometry and color). In comparison, the unfamiliarity split evaluates the network's generalization ability, i.e., its capability to perform on new and unfamiliar datasets (Goodfellow et al., 2016, p. 108).

Table 4.3: Description and train-test data split for all the experiments.

Experiment	Train Data: S3DIS Areas	Train Data: HPS Scans	Test Data	Description	Train-Test Split
1	1, 2, 3, 4, 6	-	S3DIS Area-5	Generalization within the same dataset	**Unfamiliar (Model generalization):** train-test buildings with no similarities in feature representations
2	1, 2, 3, 4, 5, 6	-	**a)** HPS Scans: 5, 6 **b)** iPhone data	Generalization with a dataset from different sensors: a) MLS lidar b) Smartphone lidar	
3	1, 2, 3, 4, 5	-	S3DIS Area-6	Areas 1, 3, and 6: Similarly-looking buildings.	**Familiar:** train-test buildings with similarities in feature representations
4	1, 2, 3, 4, 5, 6	1, 2, 3, 4	HPS Scans: 5, 6	**Domain adaptation:** Improve the performance of the DL network for the lidar dataset	

4.2.2.1. Unfamiliar Train-Test Split

For model generalization, we design Experiments 1 and 2 in Table 4.3. Here, generalization means the network's capability to perform on new and unfamiliar building datasets (Goodfellow et al., 2016, p. 108). By design, the S3DIS dataset has areas representing parts of buildings that look alike with some common architectural features and objects. However, Area-5 was captured from a different building than all other areas in the S3DIS dataset (Armeni et al., 2017). Based on this information, for Experiment-1, we follow a standard train-test split defined by Armeni et al. (2017) to test the designed methodology's generalization ability. The data split in Experiment-1 is non-overlapping, ensuring that no similarly looking buildings appear in the train and test set. However, the train and test areas belong to the same dataset.

Though the HPS and iPhone datasets are also set in office and educational spaces like S3DIS, they are structurally and architecturally different with variations in object representations. Hence, we use these differences to assess the ability of the S3DIS-only trained model to generalize on different datasets obtained from different lidar sensors (Experiments 2a and 2b in Table 4.3). These test datasets also consist of distinct types of safety-related asset representations than S3DIS. Here, Experiment-2(b) is a qualitative test on the iPhone data, and the results are evaluated visually.

4.2.2.2. Familiar Train-Test Split

We design Experiments 3 and 4 in Table 4.3 with a familiar data split. Here, familiarity indicates that the train-test areas share similar feature representations. However, they are not entirely alike, and the test areas are still unseen by the network.

In S3DIS, Areas 1, 3, and 6 and Areas 2 and 4 have commonalities among safety-related assets, as shown in Figure 4.2. Using this knowledge, we design Experiment-3 in Table 4.3 to evaluate the model's performance after training it with features familiar to the test area. Though train and test areas share few similarities in safety-related assets and architectural features, they are not entirely alike, ensuring that the model is not biased. Additionally, we repeat Experiment-3 with subsampled point clouds to estimate the smallest possible point cloud resolution feasible with the designed methodology to perform scene segmentation and thus, identify safety-related assets. We use the **spatial subsample** method in CloudCompare to generate the subsampled point clouds. Here the spacing between two points is used to select the points from the original point cloud. The higher the point spacing value, the fewer points retained from the original point clouds.

For domain adaptation, a type of transfer learning approach, we design Experiment-4. **Domain adaptation** is a process where learning from one setting is utilized to improve generalization in another setting for a similar task but with a slightly different input data distribution (Goodfellow et al., 2016, p. 534). Here, we retrain the S3DIS-only trained model from Experiment-2 on a different dataset acquired from the lidar sensor (HPS dataset) to improve the DL model's learning and overall abstraction ability. Among the six indoor scans in the HPS dataset, scans 1, 2, and 5 and scans 3, 4, and 6 have similarities in appearance concerning the interiors and objects in the scene. Hence, based on visual examination, we divide the train and test scans as shown in Table 4.3 for Experiment-4.

Here, Experiment-4 includes multiple representations of each safety-related asset type among the chosen training areas. For example, the network trains on different representations of fire extinguishers in the S3DIS area and HPS scans, namely wall-mounted and cylindrical fire extinguishers, as shown in Figure 4.2. However, the test area has just the cylindrical type of fire extinguisher. In the real-world such scenarios are plausible. Therefore, we evaluate the designed methodology's performance in such cases using Experiment-4. Finally, we use the performance from Experiments 2(a) and 4 to compare the model's ability to adapt and improve on datasets from new buildings and with new representations of safety-related assets.

4.3. Scene Segmentation: One-shot and Stage-wise Methods

We designed, One-shot method and Stage-wise method for performing scene segmentation to identify safety-related assets, as illustrated in Figure 4.5.

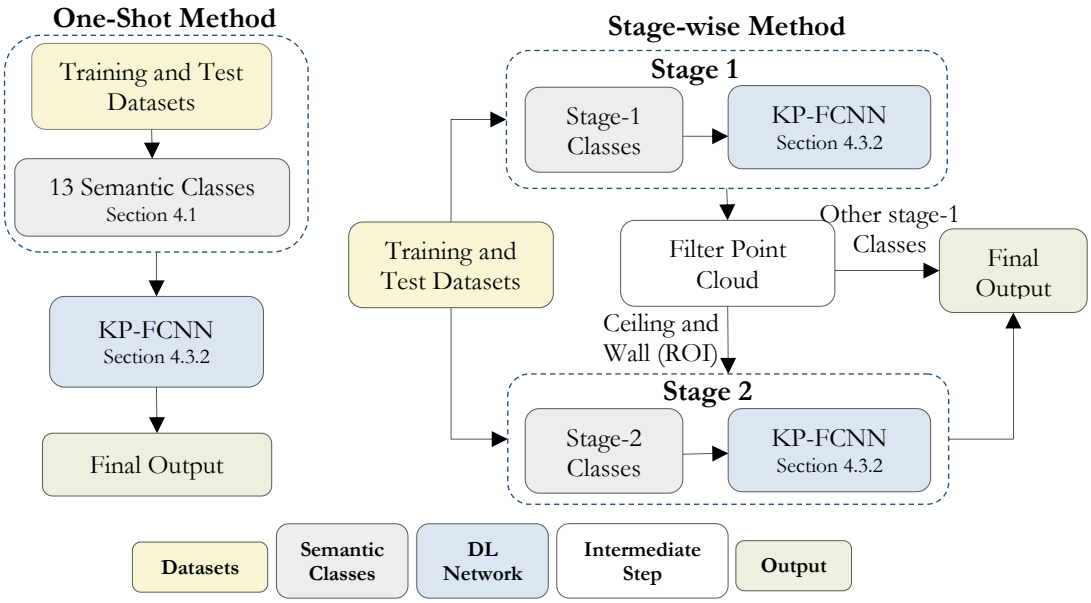


Figure 4.5: Scene segmentation approaches in the designed methodology. The One-shot method processes the whole point cloud at once; the Stage-wise method first identifies a region of interest (ROI) like the ceiling and wall where the safety-related assets are placed (Image source: author).

In the **One-shot method,** we perform scene segmentation for all the 13 semantic classes listed in Section 4.1 at once, using the entire indoor scene data. However, the indoor datasets contain irrelevant classes like furniture and clutter, which form a noticeable part of the datasets. For example, from Table 1.1, classes like furniture and clutter add up to 25% of the S3DIS dataset. Additionally, it is a known fact that most chosen safety-related assets have a defined placement in a building. For example, the exit signs are closer to the roof and primarily in the hallway, or the ventilation ducts are in the ceiling. Hence, we use this prior knowledge and propose the **Stage-wise method** to filter out the irrelevant information and segregate ceiling and wall sections as a region of interest (ROI) to look for safety-related assets, as shown in Figure 4.5.

In the stage-wise method, the 13 semantic classes defined in Section 4.1 are initially modified into **Stage-1 classes:** ceiling, wall, floor, furniture, clutter, door, and window. The lights, ventilation ducts, and exit signs are merged into ceiling class, while the fire switch and fire extinguisher are merged into wall class. Then we perform scene segmentation and using Stage-1 results, the predicted ceiling and wall points are filtered out as ROI. This ROI is again segmented for **Stage-2 classes**: ceiling, wall, light, ventilation duct, exit sign, fire switch, and fire extinguisher. Figure 4.6 summarizes each stage's regrouping for the ceiling and wall classes.

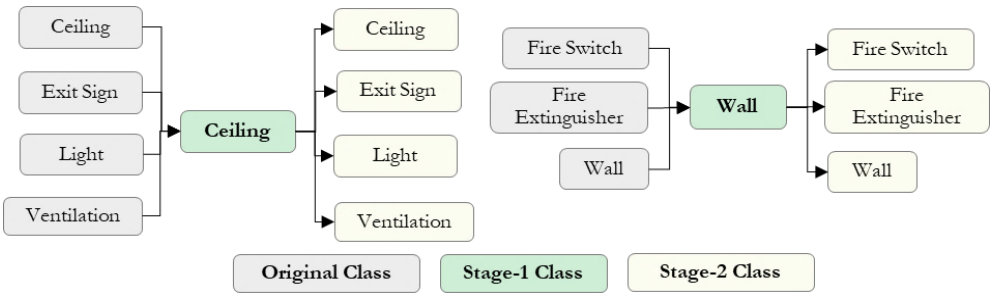


Figure 4.6: Class labels regrouped for ceiling and wall at each stage in the Stage-wise method (Image source: author).

Further, in Stage-2, we reduce the S3DIS training dataset to focus on areas like hallways, auditoriums, and conference rooms where the safety-related assets are more likely to be present to provide the network with more relevant and helpful information than the entire vast dataset. Finally, the Stage 1 and 2 results are combined to generate a complete indoor scene semantic layout for all the 13 semantic classes.

Since the main objective of this research is to explore the scope of identifying safety-related assets with DL, we primarily use the One-shot method for all the experiments designed in Table 4.3. In contrast, the Stage-wise Method is an alternate approach assessed only through Experiment-1. This alternate approach is a proposed improvement to enhance asset identification by reducing data volume before searching for safety-related assets, especially small-sized assets.

4.3.1. KP-FCNN – Network Architecture

KP-FCNN is the DL network used for scene segmentation tasks to identify safety-related assets using both methods described in Section 4.3. It is a CNN with an encoder and decoder with skip links between them, as illustrated in Figure 4.7. Each encoder layer comprises two convolutional blocks with a regular KPConv layer used for pooling features (strided KPConv) as the first block except for the first layer. Each convolutional block in the encoder consists of a KPConv layer for convolution, batch normalization, and a leaky ReLu activation. The decoder derives the point-wise features using the nearest upsampling method. The features between the intermediate layers of the encoder and decoder are passed using the skip links as shown in Figure 4.7, which are concatenated to the upsampled features. These features are further processed using a unary convolution equivalent to a 1 x 1 image convolution.

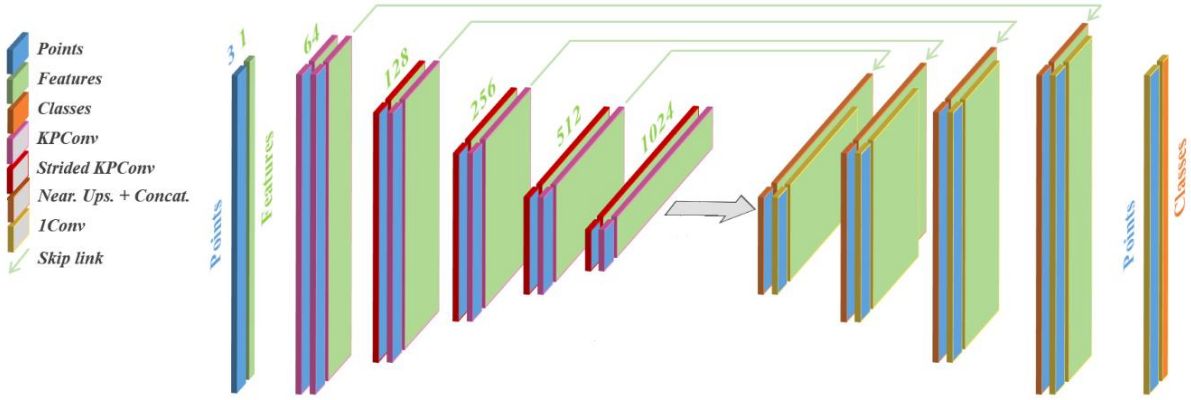


Figure 4.7: Illustration of KP-FCNN network with encoder and decoder blocks (Image source: Thomas et al., 2019).

A Kernel Point Convolution (KPConv) layer takes points $\mathcal{P} \in \mathbb{R}^{\mathrm{N} \times 3}$ and their corresponding features $\mathcal{F} \in \mathbb{R}^{\mathrm{N} \times D_{in}}$ and the neighborhood indices matrix $\mathfrak{N} \in [\![1, \mathrm{N}]\!]^{\mathrm{N}' \times n_{max}}$ with N' computed neighborhood locations and n_{max} as the size of the biggest neighborhood. For each layer j, the network parameters are inferred from layer-wise cell size dl_j which depends on the first subsampling cell size dl_0. For K kernel points, the influence distance (σ_j) is set as $\sigma_j = \Sigma \times dl_j$ from which the kernel point layout distance d_{center} and convolution radius r are inferred. For deformable kernels, the convolution radius is set to $r_j = \rho \times dl_j$. At every pooling layer, the cell size is doubled $dl_{j+1} = 2 \times dl_j$ along with other related parameters to increase the receptive field and progressively reduce the number of points (Thomas et al., 2019). A grid subsampling method is applied to ensure spatial consistency of the point sampling locations and further control the input point density at each network layer (Thomas et al., 2019). This subsampling strategy projects the point cloud into a grid with a chosen voxel size (here dl_0) and samples one point per voxel, which can be any point within the voxel (Thomas, 2019). To preserve the original shape of the input, KP-FCNN keeps the barycentre of the points in the voxel. The parameter Σ controls the point density for the convolution, as illustrated in Figure 4.8.

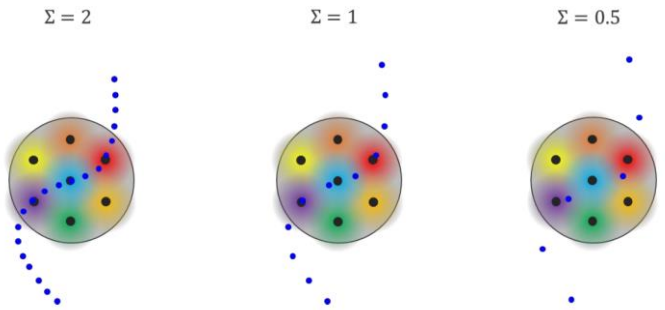


Figure 4.8: Illustration of input point density controlled by parameter Σ for KPConv, where different colors indicate each kernel point's influence area (Image source: Thomas, 2019).

4.3.2. KP-FCNN – Network Parameters

For the current research, we use the PyTorch KP-FCNN implementation from GitHub (Hugues, 2020) and customize it for the datasets in Section 3.1. The network takes the area-wise input point clouds as text files with geometry and color fields with the semantic class labels as the file names. As a pre-processing step in the network, the class labels are extracted and added to every point, generating area-wise PLY files with seven properties - X, Y, Z, R, G, B, and label. Deciding the input features for the network is a crucial step. Using X and Y properties for safety-related assets is not particularly useful as they do not carry helpful information. For example, the XY position of an exit sign or a fire switch is irrelevant, as the objects could be anywhere in a building. However, the Z values give information on the height above ground which is relevant for understanding the spatial context. For example, an exit sign would be far from the ground and closer to the ceiling, with the ground at Z = 0 and the ceiling higher. Hence, we choose four input features (R, G, B, and Z) along with the standard constant feature encoding the geometry of the input points, making a total of five input features $(D_{in} = 5)$ for the network.

Since the point cloud dataset is too big to process as a whole, KP-FCNN uses spherical sub clouds of the scene with a radius (R), which are smaller than the whole input dataset. Based on several experiments, Thomas et al. (2019) suggests that R should be chosen such that it is proportional to $50 \times dl_0$. The R and dl_0 values vary depending on the dataset, the size of the objects of interest and the level of details needed for the designated task. Since we focus on small objects like fire switches and exit signs in the current study, lower dl_0 values for subsampling would provide more details for feature learning of such small objects. Based on discussions with the original author of KP-FCNN on GitHub (Anjanappa, 2022), we use $dl_0 = 1cm$. Based on this chosen dl_0 value, with trial and error, we choose R values (proportional to $50 \times dl_0$) for both one-shot and stage-wise methods. Other network parameters are kept the same as suggested in the original network implementation with $K = 15,\ \Sigma = 1.2$ and $\rho = 5.0$.

As explained in Section 2.3, KP-FCNN offers two different point sampling methods to pick sub clouds. In the training phase of One-shot and stage-2 of Stage-wise methods, we use the random picking strategy to ensure that the network sees the minority classes (safety-related assets) more often, especially the small-sized assets. However, we use regular picking in stage-1 of the Stage-wise scene segmentation method, as the semantic classes in this stage are not class imbalanced. For testing, we pick the spheres regularly in both methods to ensure that each point is evaluated multiple times at different sphere locations.

Further, a batch size of 6 is used with a learning rate of 10^{-2} at which the training parameters and network weights are updated. A momentum gradient descent optimizer with a momentum of 0.98 is used to minimize the point-wise cross-entropy loss. By design, the network requires a minimum of 400 epochs to converge, so we train the network for 500 epochs for all the experiments, with each epoch as a 500-optimizer step, defining the number of input spheres. Here it is equivalent to 5000 spheres to be seen by the network.

Table 4.4: Chosen KP-FCNN training parameters for the proposed methods.

Network Parameters	Description	One-shot Method Values	Stage-wise Method Values	
			Stage 1	Stage 2
D_{in}	Input feature dimension	5	5	5
R	Radius of spherical subclouds in meter (m)	0.7	1.0	0.7
dl_0	First subsampling grid size in meter (m)	0.01	0.01	0.01
Sampling strategy	Input sphere sampling strategy for training True: Regular picking; False: Random picking	False	True	False
Class weights	Weights for cross-entropy loss	Yes	No	Yes

In addition to the network parameters, the KP-FCNN implementation includes various data augmentation strategies like scaling, rotation, flipping, noise addition, and color annealing choices to increase the input data variability, which would improve the network's robustness. Table 4.5 presents the various augmentation strategies and parameters used with the designed methods. The most relevant strategy for this research is color annealing, which enables the network to learn features occasionally using only geometry without colors.

Table 4.5: Data augmentation approaches in KP-FCNN and parameters used for both the proposed methods.

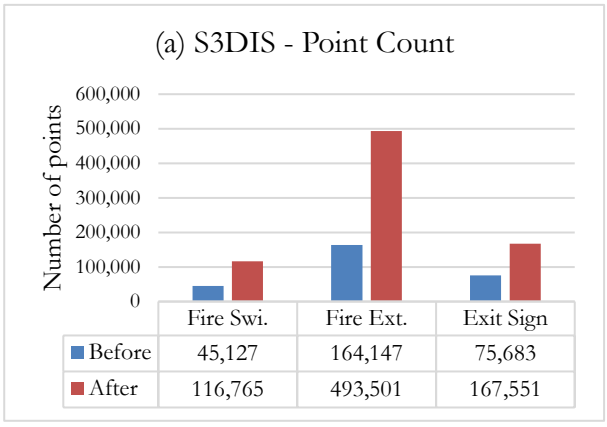
Property	Description	Value
Scaling	Input point clouds scaled independently in each dimension	Minimum scale = 0.9 Maximum scale = 1.1
Rotation	Rotate the point clouds around the vertical axis	Random angle in $[0, 2\pi]$
Noise	Gaussian noise is added to point coordinates for perturbing the point positions	0.001 meter
Color	Color annealing probability to erase color features of some input clouds	0.8

4.3.3. Strategy for Class-imbalanced Data

S3DIS and HPS datasets discussed in Section 3.1 offer large point clouds with a high point density of around 273million and 500million points, respectively. However, as explained in Section 1.3, these datasets are class-imbalanced for safety-related asset classes. To manage the class imbalance problem for DL algorithms, Johnson and Khoshgoftaar (2019) summarized three existing methods:

i. **Data-level methods**: Modifying the training data to decrease imbalance.
ii. **Algorithm-level methods:** Boosting sensitivity towards the minority class by adapting the network's learning or decision method.
iii. **Hybrid methods:** Both (i) and (ii).

We use a hybrid method with data and algorithm level solutions to address the class imbalance problem. The data-level method is implemented for the following classes: exit signs, fire switches, and extinguishers. We use a synthetic data generation strategy, a type of data augmentation, to add copies of already existing assets to the training dataset. In CloudCompare, the existing instances of these assets were first cloned and then placed in multiple locations using the interactive transformation tool. Figure 4.9 shows the change in the point counts for the assets with the data-level solution.



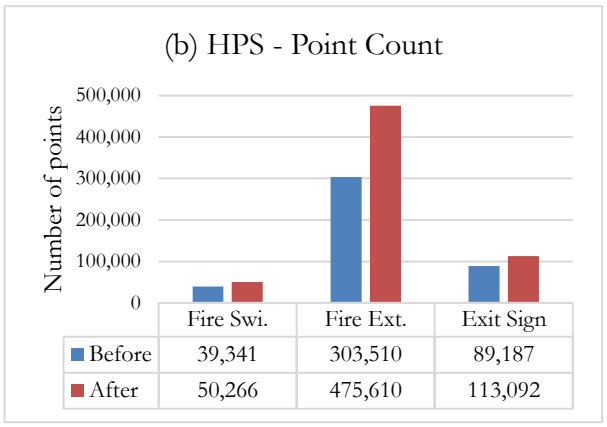


Figure 4.9: Point counts before and after data-level solution for (a) S3DIS and (b) HPS datasets.

By design, the random picking strategy in KP-FCNN picks spherical sub clouds of the scene with a chosen radius based on class labels for training the network. We use this information to manually place copies of assets in the areas like hallways where assets like exit signs, fire switches, and extinguishers are likely to be present and in proximity to each other. That way, when KP-FCNN picks a sphere based on one safety-related asset class, it would also include information on other asset classes as neighborhood points within the chosen radius, providing information to learn about them simultaneously to the network. We ensure that the point counts for these assets do not vary at a high rate, as seen in Figure 4.9, so no bias is created.

As an algorithm-level method, we added class weights during the training phase to reshape the cross-entropy segmentation loss of the network according to class balance to boost the sensitivity to the minority classes (Johnson and Khoshgoftaar, 2019). The class weights used are dynamically calculated during training in proportion to the point counts for chosen batches of data and based on the semantic classes. We use the KP-FCNN author's suggested formula to calculate these weights.

$$Class\ Weights = \sqrt{\frac{100}{P}}$$

Equation (4.1)

where, P is the total point proportions for all the classes in the chosen training data batches.

4.4. Evaluation Metrics

The quantitative performance of the designed algorithm can be derived from the detailed class-wise confusion matrix obtained from the test point cloud results of the DL network. The generated confusion matrix provides a detailed breakdown of each class's correct and incorrect classifications. Table 4.6 summarizes a simple confusion matrix for binary classification with green cells as correct (positive) and red cells as incorrect (negative) classifications.

Table 4.6: Confusion matrix with green cells as correct and red cells as incorrect classifications.

		Actual Class	
		Positive	**Negative**
Predicted Class	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Overall Accuracy (OA) is the most used metric to evaluate classification results (Equation 4.2). However, when working with class-imbalanced data, accuracy values are misleading as it is dominated by the majority classes (Johnson and Khoshgoftaar, 2019). One way to overcome this problem is to use average Per class Accuracy (avPA). However, other metrics like Precision, Recall, and F1-score are better evaluation metrics when dealing with class-imbalanced data.

$$Overall\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation (4.2)

Precision (Goodfellow et al., 2016, p. 418) measures the percentage of positive predictions reported by the model that are actually correct or true positives (Equation 4.3). Though it is sensitive to the class imbalances in the data as it considers the number of points incorrectly labeled as positive predictions (FP), it does not provide insight into the number of points mislabeled in the positive group (FN). On the other hand, **Recall** (Equation 4.4) measures the correct positive predictions over all the points that should have been predicted as positive (Johnson and Khoshgoftaar, 2019). Since recall is only dependent on the positive group, it is not affected by the data imbalance. However, it does not consider the number of positive samples wrongly classified as positive (FP). The **F1 Score** is used to overcome the trade-offs between the precision and recall, combining them using a harmonic mean (Equation 4.5).

$$Precision = \frac{TP}{TP + FP}$$

Equation (4.3)

$$Recall = \frac{TP}{TP + FN}$$

Equation (4.4)

$$F1\ Score = \frac{(1 + \beta^2) \times Recall \times Precision}{\beta^2 \times (Recall + Precision)}$$

where, β decides the relative importance of precision versus recall.

Equation (4.5)

In particular, for scene segmentation tasks, **mean Intersection over Union (mIoU)** is a widely used evaluation metric (Guo et al., 2021; Liu et al., 2019). IoU is the proportion of overlap of the area between ground truth and predicted results (TP) over the area of their union (TP + FP + FN), as shown in Equation 4.6 and Figure 4.10. mIoU is the mean of the IoUs of all the classes.

$$IoU = \frac{TP}{TP + FP + FN}$$

Equation (4.6)

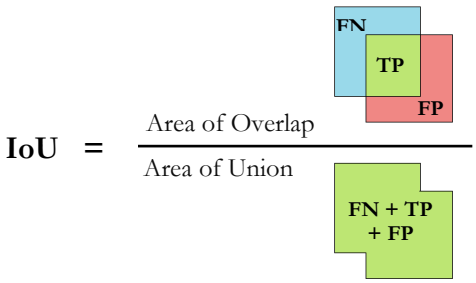


Figure 4.10: Illustration of IoU - the area of overlap as TP; the area of union as TP+FP+FN.

4.4.1. Asset Identification Rate

Given a method, there always exists a level of noise that causes errors in the results. However, determining a suitable metric to assess the performance would help understand if the chosen methodology is suitable for the desired task and guide the subsequent actions to improve the method (Goodfellow et al., 2016, p. 417). In the case of scene segmentation of a point cloud, the confusion matrix is the standard assessment approach derived from the point-wise labels of the predictions made by the DL network. The evaluation metrics described in Section 4.4 are computed using these point-wise counts, which would be optimal to assess per point metrics for a scene. However, the final stakeholder or user using this research as a real-world product or service for asset management would be interested in finding which assets are located where within the building. More specifically, (i) total asset counts per area, (ii) the presence or absence of an asset instance at a particular location; (iii) Reduced false asset locations for reliability.

In this regard, we define an evaluation process as binary classification by segregating the correct and incorrect asset instances identified by the designed methodology. We propose **Asset Identification Rate (AIR)** to evaluate the rate of correctly identified asset instances over the total count for a particular safety-related asset class in each area. Inspired by the detection mechanism used in PointNet (Qi et al., 2017a), we design a workflow presented in Figure 4.11 for this research.

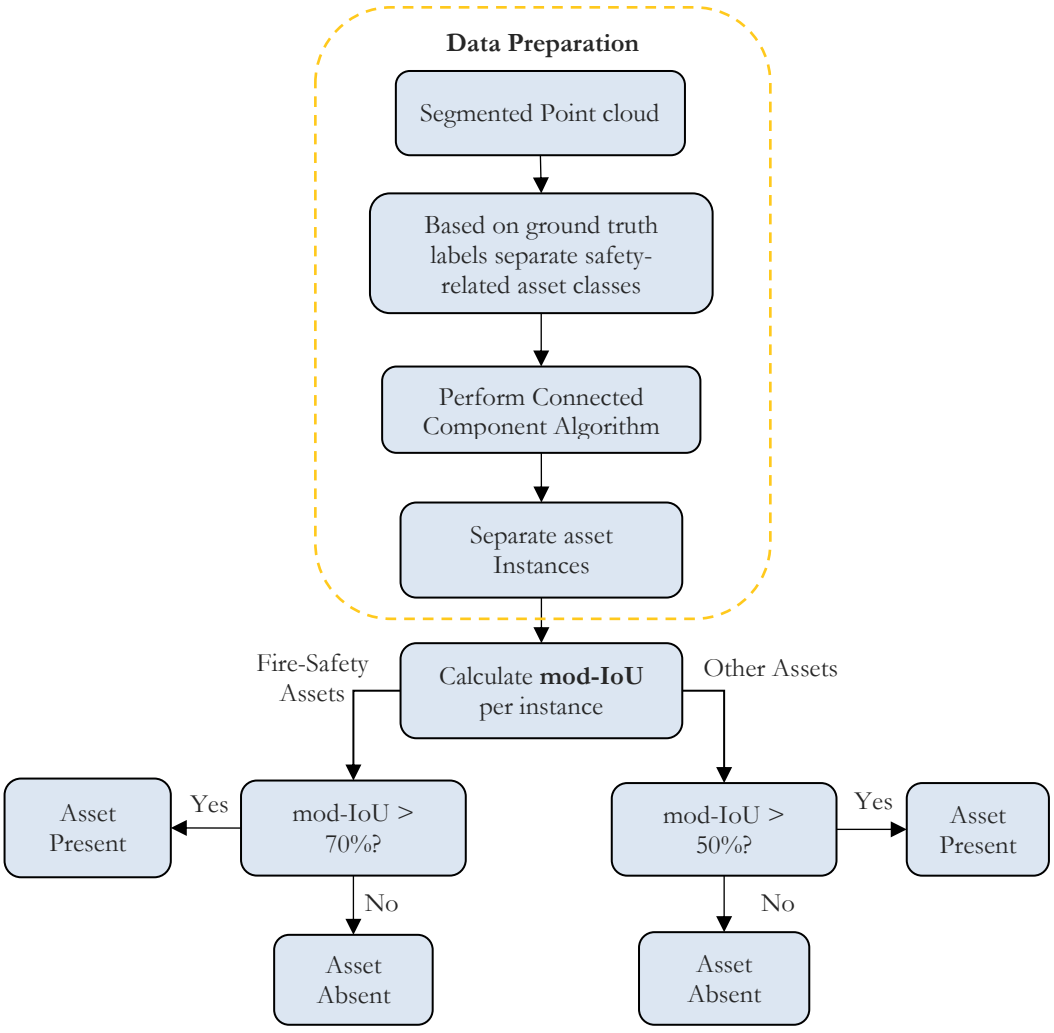


Figure 4.11: Workflow to calculate asset identification rate for safety-related assets (Image source: author).

The area-wise results from KP-FCNN contain the ground truth and the predicted labels for each point in the point cloud. We use this area-wise point cloud to generate a sub-cloud based on ground truth labels for each safety-related asset class. Using the obtained safety-related asset sub-clouds, we separate asset instances using the Connected Component Algorithm in CloudCompare, where a sub-cloud is segmented into smaller parts, i.e., individual assets. Figure 4.12 illustrates these three steps of data preparation for door class.

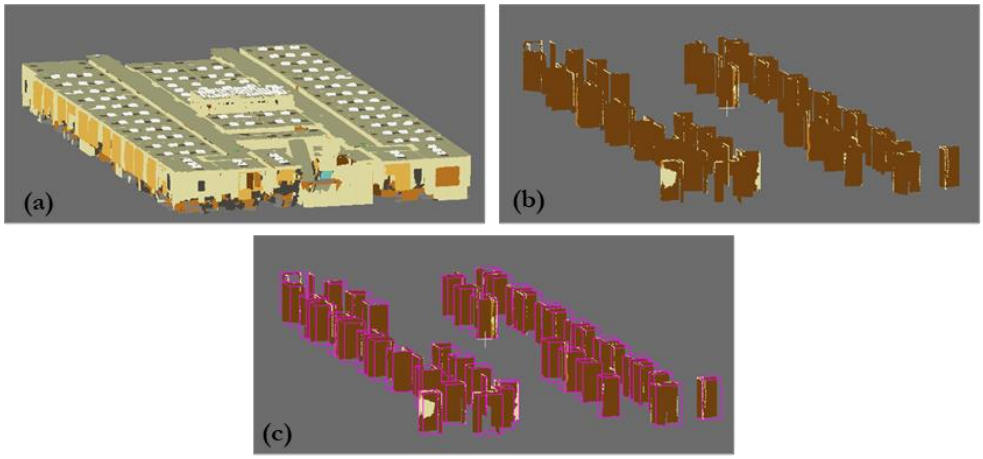


Figure 4.12: Data preparation steps to calculate asset identification rate: (a) Area-wise results from KP-FCNN; (b) Using ground truth labels, a sub-cloud for door safety-related asset class is generated; (c) Asset instances are separated using the Connected Component Algorithm, where each instance is shown with a pink bounding box.

We use a slightly modified version of IoU (Equation (4.6), namely mod-IoU, using Equation 4.7 as a deciding criterion to find the AIR. Here for the given ground truth points of a safety-related asset instance, TP are correctly labeled points from KP-FCNN, and FN are wrongly labeled points, as illustrated with an example in Figure 4.13. In Figure 4.13, the ground truth for a door instance shows TP in brown and FN in yellow.

$$mod\text{-}IoU = \frac{TP}{TP + FN}$$

Equation (4.7)

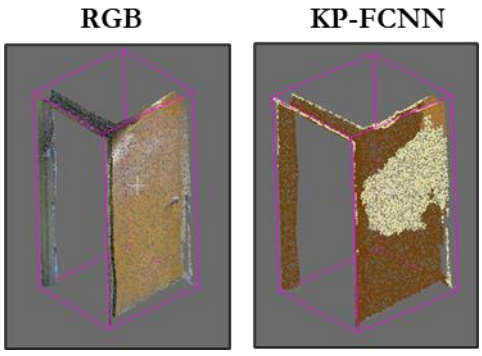


Figure 4.13: The ground truth for a door instance in RGB (left) and predictions from KP-FCNN with TP in brown and FN in yellow (right).

Since the fire safety-related asset classes like fire switches and extinguishers must be identified with the highest reliability and accuracy, we consider a mod-IoU>70%. It means that if 70% of points in the ground truth for a fire switch are identified correctly, it is counted as a correctly identified asset. Additionally, as stairs are not instance-wise structures, calculation of instance-wise identification rate is not feasible. Therefore, we calculate AIR only for light, fire switch, fire extinguisher, ventilation duct, the exit sign, door, and window safety-related asset classes using the workflow in Figure 4.11.

5. RESULTS

In this chapter, we present the research findings through Experiments 1-4 in Table 4.3, using One-shot and Stage-wise methods designed in Section 4.3. Here, the **One-shot Method** is the primary method used for all the experiments, which performs scene segmentation for all the 13 semantic classes in Figure 5.1 at once, using the entire scene data. In contrast, the **Stage-wise Method** is an alternate approach assessed only through Experiment-1, which reduces the data volume of the scene to separate a region of interest (ceiling and wall structures) and then identifies the safety-related assets present in those regions. Figure 5.1 shows the 13 semantic classes and the colors used to represent each class in the results.

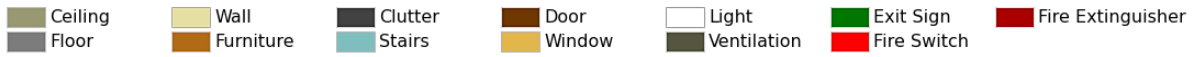


Figure 5.1: Semantic classes and colors used to represent them in results

We present the One-shot Method's generalization ability on new and unfamiliar datasets (Experiments 1 and 2) in Section 5.1 and the performance with familiar datasets (Experiments 3 and 4) in Section 5.2. Then, in Section 5.3, we compare the results of One-shot and Stage-wise Methods using Experiment-1. All the experiments are described in Section 4.2.2; however, they are again listed here for an easy and clear understanding for the reader. In this chapter, the abbreviations **OA** is overall accuracy, **mIoU** is mean Intersection over Union, and **AIR** is Asset Identification Rate. Also, only the evaluation metrics calculated for safety-related assets are presented here. IoU and confusion matrices for all 13 classes are added in Appendices 1 and 2.

5.1. Model Generalization – New Building Datasets (One-shot Method)

To evaluate the network's generalization ability, we performed two experiments in Table 5.1 using the One-shot Method and its corresponding network parameters in Table 4.4. Here, generalization means the network's capability to perform on new and unfamiliar building datasets (Goodfellow et al., 2016, p. 108). For both experiments, the train-test areas were non-overlapping in terms of feature representations.

Table 5.1: Experiments performed for model generalization.

Exp.	S3DIS Train Areas	Test Data	Description (Model generalization)
1	1, 2, 3, 4, 6	S3DIS Area-5	Generalization within the same dataset (S3DIS)
2	1, 2, 3, 4, 5, 6	a) HPS scans: 5, 6	Generalization with a new building dataset from a different sensor - MLS lidar
		b) iPhone data	Generalization with a new building dataset from a different sensor - smartphone lidar data

5.1.1. Experiment – 1

In this experiment, the train-test areas belonged to the same dataset (S3DIS), but Area-5 used for testing is from a different indoor space than the areas used for training the network (Armeni et al., 2017). We examined the data and noticed that Area-5 had no stairs class. Though the network was trained for all 13 semantic classes, the mIoU calculated in Table 5.2 is for 12 semantic classes, excluding stairs. Table 5.2 shows that the method achieves an **OA** of 88.3% and **mIoU** of 56.6%. Based on evaluation metrics in Table 5.2, ventilation ducts are the most well-segmented safety-related asset, and the fire-extinguisher is the poorly

segmented asset with a zero-recall rate. These results are consistent with the **AIR** in Table 5.3, with fire extinguishers showing a zero AIR and exit signs with a poor AIR. However, assets like ventilation ducts, fire switches, doors, and windows are generalized well and achieve an AIR>75%

Table 5.2: OA and mIoU for S3DIS Area-5 with evaluation metrics for safety-related assets (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Stairs	Door	Wind.
S3DIS Area - 5	IoU (%)	40.1	52	**0**	**74**	7.6	-	56.9	60.2
	Precision (%)	44.3	61	**0**	88.7	88.6	-	74	**89.8**
	Recall (%)	81	77.9	**0**	**81.7**	7.7	-	71	64.6
	F1-score (%)	57.3	68.4	**0**	**85.1**	14.1	-	72.5	75.2
	Overall Accuracy = 88.3%				**mIoU** = 56.6% (Classes = 12)				

Table 5.3: Asset identification rate for S3DIS Area-5 (highest and lowest scores in green and red).

Semantic Class	Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
Total Assets	222	32	6	124	14	70	49
Correctly Identified	158	26	0	117	1	57	39
AIR (%)	71.2	81.3	**0**	**94.4**	7.1	81.4	79.6

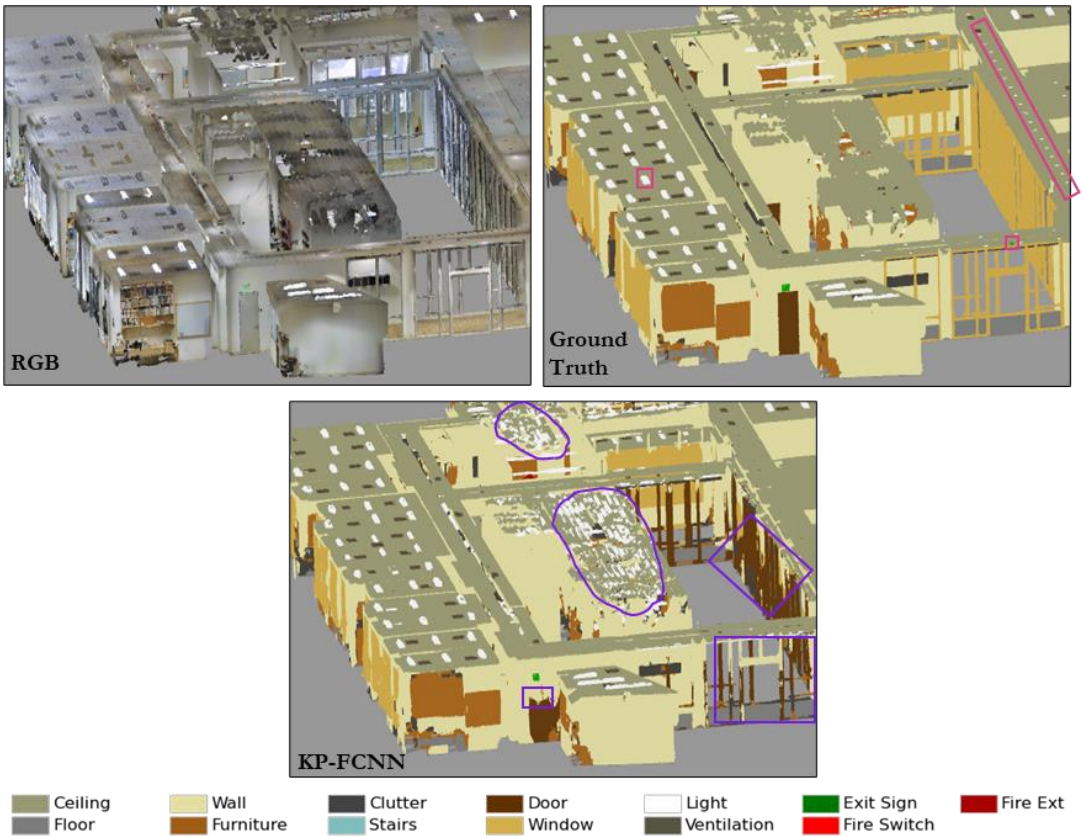


Figure 5.2: S3DIS Area-5 scene segmentation results with bounding boxes in pink for safety-related assets unidentified by the network and misclassifications in purple.

Figure 5.2 shows the RGB image, ground truth, and KP-FCNN results (one-shot method) for a section of the S3DIS Area-5. We have added bounding boxes for safety-related assets unidentified by the network in the ground truth image (pink) and misclassifications among classes in the KP-FCNN image (purple). The permanent structures like walls, ceilings, and floors are well-segmented. However, there are misclassifications for the lights and window safety-related assets in Figure 5.2. Firstly, the window class is wrongly segmented as the door and wall, and secondly, the ceiling is misclassified as the lights. On visual inspection, Figure 5.2 shows that round-shaped lights in this area are primarily unidentified (in the pink bounding box in the ground truth image).

5.1.2. Experiment – 2(a)

In this experiment, the train-test areas belonged to different datasets. We evaluated the S3DIS-only trained network's generalization ability on the MLS lidar HPS dataset. Among the HPS scans chosen for testing, scan 5 had no stairs, and scan 6 had no ventilation duct classes. Therefore, mIoU is calculated for the other 12 semantic classes corresponding to the scans, but the method was trained for all 13 semantic classes.

Table 5.4: OA and mIoU for HPS scan 5 and 6 with evaluation metrics for safety-related assets (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Stairs	Door	Wind.
HPS scan 5	mIoU (%)	56.8	**69.8**	**0**	8.1	5	-	7.3	30.9
	Precision (%)	78.1	**78.3**	**0**	24.7	5.4	-	7.8	44.3
	Recall (%)	67.5	**86.6**	**0**	10.8	41.6	-	52.2	50.5
	F1-score (%)	72.4	**82.2**	**0**	15	9.5	-	13.6	47.2
	Overall Accuracy = 82.5%				**mIoU** = 43.1% (Classes = 12)				
HPS scan 6	mIoU (%)	27	34.2	6.7	-	**40.9**	**0**	25.9	16.9
	Precision (%)	38.5	39.4	28.6	-	**58.5**	**0**	46.7	22.1
	Recall (%)	47.5	**72.2**	8	-	57.7	**0**	36.7	41.7
	F1-score (%)	42.5	51	12.5	-	**58.1**	**0**	41.1	28.9
	Overall Accuracy = 86%				**mIoU** = 41.6% (Classes = 12)				

Table 5.5: Asset identification rate for HPS scans 5 and 6 (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
HPS scan 5	Total Assets	150	8	5	41	2	6	2
	Correctly Identified	68	7	0	5	1	3	1
	AIR (%)	45.3	**87.5**	**0**	12.2	50	50	50
HPS scan 6	Total Assets	105	6	5	-	18	23	6
	Correctly Identified	33	4	0	-	14	5	2
	AIR (%)	31.4	66.7	**0**	-	**77.8**	21.7	33.3

Table 5.4 show that the method achieves an **OA** of 82.5% and a **mIoU** of 43.1% for the HPS **scan 5**. Based on the calculated metrics in Table 5.4, the fire switch class is segmented well for this scan. For the HPS **scan 6**, the method achieves an **OA** of 86% and a **mIoU** of 41.6%, with the exit signs well-segmented and stairs poorly segmented. The **AIR** in Table 5.5 shows that 87.5% of fire switches in HPS scan 5 and 77.8% of exit signs in HPS scan 6 were successfully identified. However, the method fails to identify the cylindrical fire extinguishers in both the scans, achieving zero AIR scores, as the S3DIS-trained network is unfamiliar with the cylindrical type of fire extinguishers.

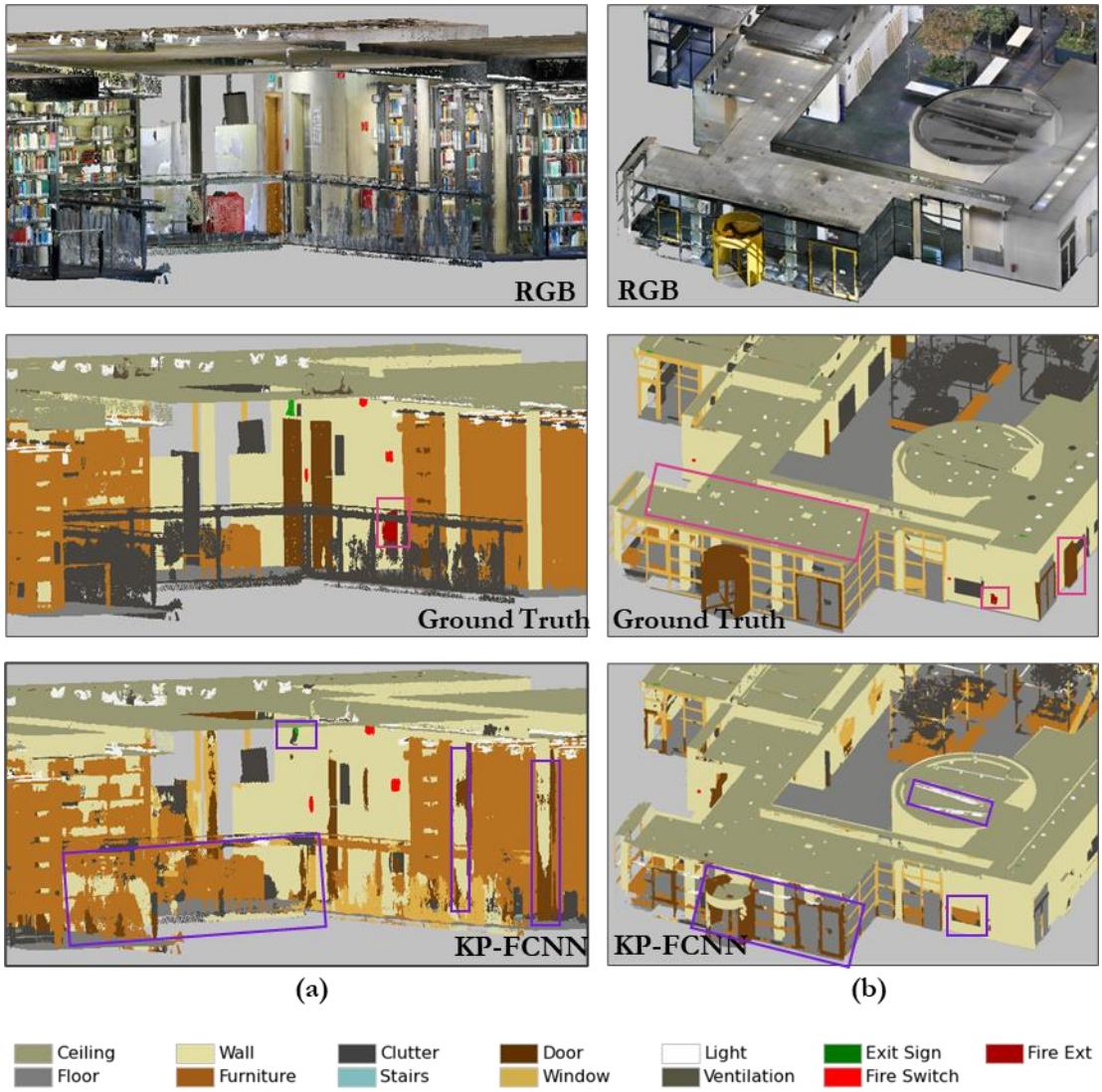


Figure 5.3: Scene segmentation results for (a) Scan 5 and (b) Scan 6, with bounding boxes in pink for safety-related assets unidentified by the network and misclassifications in purple.

Figure 5.3 shows the RGB image, ground truth, and KP-FCNN results (one-shot method) for a section of HPS scans 5 and 6. We have added bounding boxes for safety-related assets unidentified by the network in the ground truth image (pink) and misclassifications in the KP-FCNN image (purple). The permanent structures like walls, ceilings, and floors are well-segmented for both scans. But Figure 5.3(a) shows that for scan 5, the clutter class is wrongly segmented as furniture, and the pillars (here wall class) in the scene are partially mixed up with the door. From Figure 5.3(b), for test scan 6, the windows are wrongly segmented as the door and wall classes (purple bounding box in (b) KP-FCNN image), and a few ceiling lights are unidentified (pink bounding box in (b) ground truth image). As per Figure 5.3, the commonly unidentified asset in both scans is the fire extinguisher, marked pink bounding boxes in the ground truth images.

5.1.3. Experiment – 2(b)

The areas for training (S3DIS) and testing (iPhone scans) in this experiment belonged to different datasets. We used four scans from iPhone data, one room, two hallways, and one lobby for this experiment. Since this experiment uses a predict-only dataset, the results are qualitatively assessed, and no evaluation metrics are calculated. Figures 5.4 (a)-(d) present the predictions for all the iPhone scans using the S3DIS-only trained network. Figure 5.4 show that the permanent structures like walls, ceilings, and floors are well-segmented. Safety-related assets like lights, fire switches, and exit signs are identified in (a)-(c) scans, shown in the dark blue bounding boxes. But the doors in Figures 5.4 (b) and (c) and the stairs in Figure 5.4(d) are partially identified. In addition, the method fails to identify the cylindrical fire extinguishers with the yellow bounding box in Figure 5.4(d), as the network is unfamiliar with the cylindrical type of fire extinguishers.

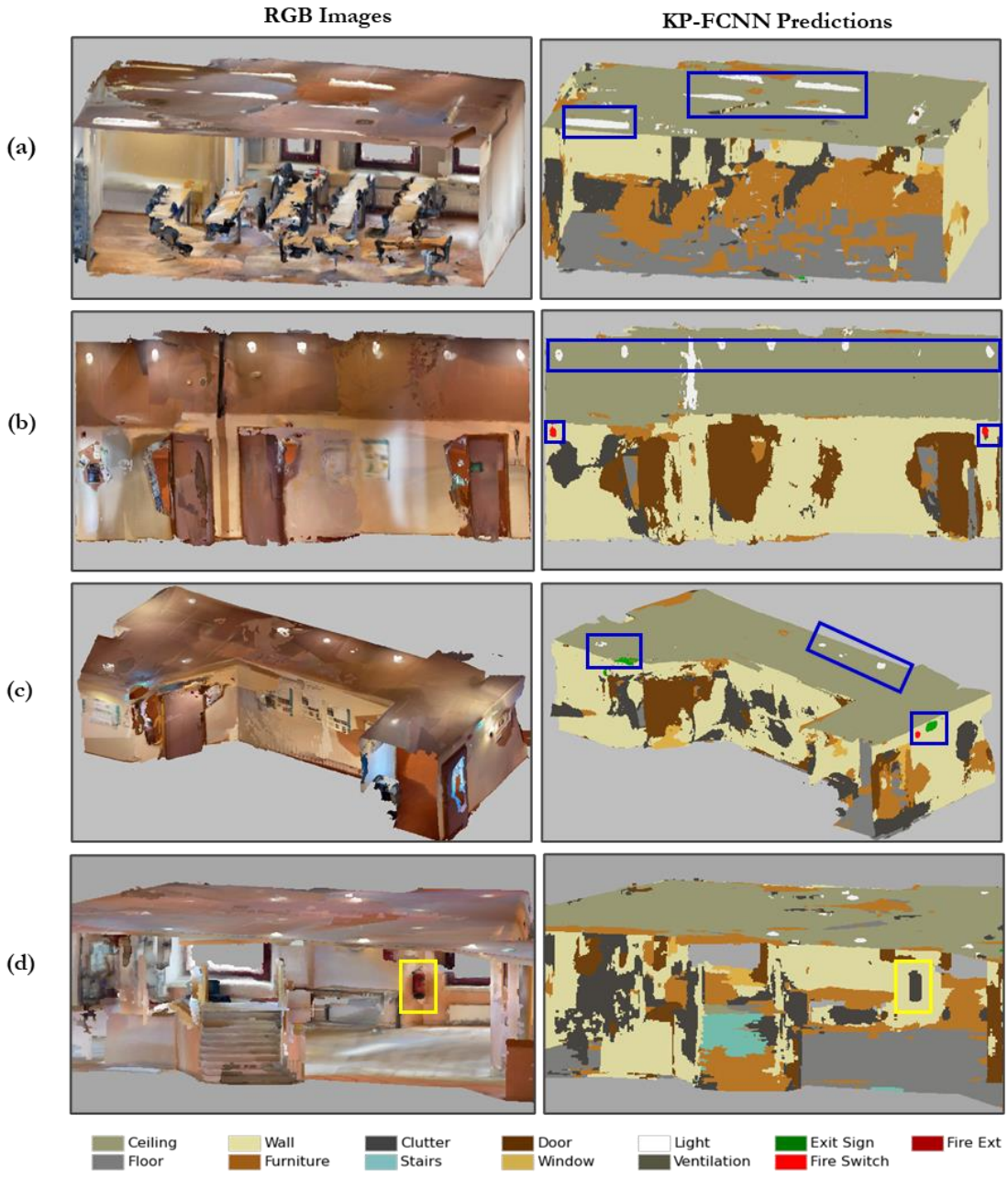


Figure 5.4: Experiment-2(b) scene segmentation results for iPhone data, with dark blue bounding boxes for safety-related assets correctly identified and yellow bounding boxes for unidentified assets in the scene.

5.2. Familiar Building Datasets (One-shot Method)

We performed Experiments 3 and 4 with the train-test data in Table 5.6 to evaluate the model's performance on a familiar dataset using the One-shot Method and its corresponding network parameters in Table 4.4. Here, familiarity means that the network has been trained on feature representations of the architecture and objects (geometry and color) similar to that present in the test area. Though train and test areas share few similarities in safety-related assets and architectural features, they are not entirely alike, and the test areas are still unseen by the network.

Table 5.6: Experiment performed for familiar building datasets.

Experiment	Train Data		Test Data	Description
	S3DIS Areas	HPS Scans		
3	1, 2, 3, 4, 5	-	S3DIS Area-6	Areas 1, 3, and 6: Similarly-looking
4	1, 2, 3, 4, 5, 6	1, 2, 3, 4	HPS scans 5, 6	Domain Adaptation: Improve the network's performance for the HPS dataset

5.2.1. Experiment – 3

Among the five training areas used in this experiment, S3DIS Areas 1 and 3 shared few similarities with the S3DIS test Area-6, and the remaining areas were different. Table 5.7 shows that the method achieves an **OA** of 94.7% and **mIoU** of 83.4% for 13 semantic classes. Based on the evaluation metrics in Table 5.7, doors are the most well-segmented safety-related asset, and fire switches are the least well-segmented assets. Furthermore, exit signs achieve the highest precision score of 95.2%, which means that among all the points predicted as an exit sign, 95.2% of them correctly match the ground truth. Table 5.8 shows that all assets are successfully identified and achieve **AIR**>75%. Additionally, assets like fire extinguishers, exit signs, and doors achieve a 100% identification rate.

Table 5.7: OA and mIoU for S3DIS Area-6 with evaluation metrics for safety-related assets (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Stairs	Door	Wind.
S3DIS Area - 6	IoU (%)	81.3	62.1	80	84.8	89.2	76.6	89.5	85.8
	Precision (%)	90.9	68.4	86.4	89.9	95.2	82.6	91.6	90.6
	Recall (%)	88.6	87.1	91.5	93.7	93.4	91.3	97.5	94.2
	F1-score (%)	89.7	76.6	88.9	91.7	94.3	86.7	94.5	92.4
	Overall Accuracy = 94.7%			**mIoU** = 83.4% (Classes = 13)					

Table 5.8: Asset identification rate for S3DIS Area-6 (highest and lowest scores in green and red).

Semantic Class	Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
Total Assets	169	11	5	116	5	49	31
Correctly Identified	133	10	5	112	5	49	29
AIR (%)	78.7	90.9	100	96.6	100	100	93.6

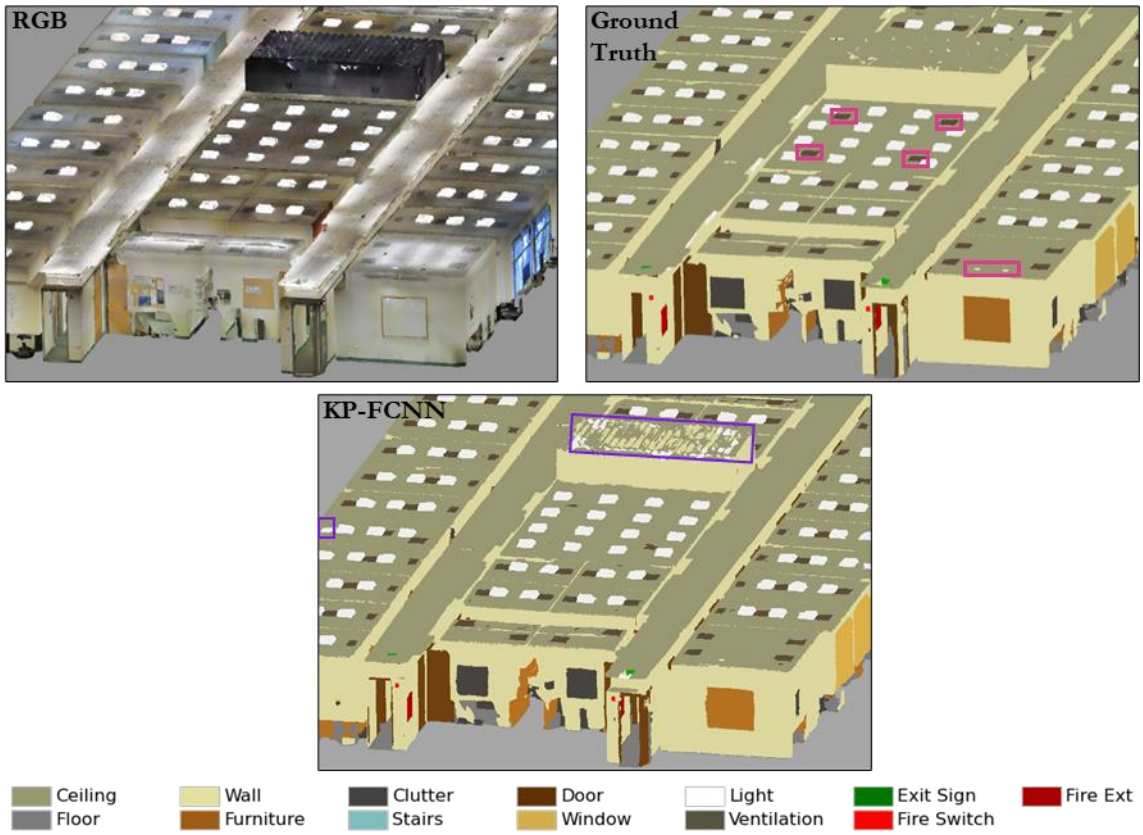


Figure 5.5: S3DIS area-6 scene segmentation results with bounding boxes in pink for safety-related assets unidentified by the network and misclassifications in purple.

Figure 5.5 shows a part of S3DIS Area-6, with an RGB image, ground truth, and KP-FCNN results (one-shot method). We have added bounding boxes for safety-related assets unidentified by the network in the ground truth image (pink) and misclassifications in the KP-FCNN image (purple). The permanent structures like walls, ceilings, and floors are well-segmented for this area. From Figure 5.5, safety-related assets are well-segmented and identified, except for a few ventilation ducts missing shown in pink bounding boxes in the ground truth image. Also, no significant misclassifications are noticeable except for the ceiling wrongly segmented, shown in Figure 5.5 with a purple bounding box.

Additionally, we repeated this experiment for subsampled S3DIS point clouds to estimate the acceptable range of point cloud resolution suitable for the designed methods to perform scene segmentation to identify safety-related assets. We used the **spatial subsample** method with point spacing values of 0.01m, 0.02m, and 0.03m to subsample the point clouds with CloudCompare. For each subsampled point cloud set, we performed Experiment-3 with the network parameters in Table 4.4 for One-shot Method. Table 5.9 shows the status of scene segmentation with each set of subsampled point clouds. As additional results, we have listed the IoU for all thirteen classes for use cases 1 and 2 in Appendix 1 (Table 2).

Table 5.9: Scene segmentation and safety-related asset identification with subsampled point clouds.

Case	Point Spacing (in meter)	Scene Segmentation	Safety-related Assets
1	0.01	Successful	Identified
2	0.02	Successful	Identified
3	0.03	Failed	Not feasible

By design, KP-FCNN expects 100000 points within the sub clouds chosen per batch for training (Thomas, 2019). The chosen network parameter *R* in Table 4.4 defines the radius of sub clouds. For case 3, the subsampled point clouds provided as input with a point spacing of 0.03m did not have sufficient points within the sub clouds to reach the expected point range per batch chosen for training. Hence, the network failed to converge while computing neighborhood points during training and could not perform scene segmentation. Hence, identifying safety-related assets was not feasible with this set of subsampled point clouds. Therefore, we conclude that point clouds up to point spacing of 0.02m are suitable in our case for performing scene segmentation using the One-shot method and its designed network parameters used in this research. However, using point clouds with much lower resolution for scene segmentation might be possible if the network parameters are altered accordingly, which needs to be evaluated.

5.2.2. Experiment – 4 (Domain Adaptation)

This experiment was performed for **domain adaptation** (a type of transfer learning) to check the S3DIS-only trained model's ability to adapt and improve on datasets from new buildings obtained from lidar sensors. We re-trained the S3DIS-only network from Experiment-2a with the HPS scans-1, 2, 3, and 4. The resulting network was assessed on HPS test scans 5 and 6. The network parameters specified for the One-shot Method in Table 4.4 were used for this experiment.

As discussed in Section 5.1.2, scan 5 has no stairs, and scan 6 has no ventilation ducts classes. Hence, Table 5.10 shows mIoU for scene segmentation of the other 12 semantic classes corresponding to the scans. Compared to the results of Experiment-2(a) in Table 5.4 and Experiment-4 in Table 5.10, the model's performance improved in Experiment-4. The **mIoU** for HPS scan 5 increased from 43.1% in Experiment-2a to 75.5% in Experiment 4. Similarly, for HPS scan 6, the **mIoU** increased from 41.6% in Experiment-2a to 58.7% in Experiment 4. Based on the metrics in Table 5.10, for scan 5, the window class is well segmented, and fire extinguishers are well-segmented for scan 6. Furthermore, for scan 5, the ventilation class is poorly segmented, and for scan 6, the lights are poorly segmented.

Table 5.10: OA and mIoU for HPS scan 5 and 6 with evaluation metrics for safety-related assets (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Stairs	Door	Wind.
HPS scan 5	mIoU (%)	68.9	73.8	72.1	**36.6**	69.9	-	59.2	**83.1**
	Precision (%)	70.3	76.7	77.2	92.4	86	-	**63.1**	**96.2**
	Recall (%)	**97.2**	95.3	91.6	**37.8**	78.9	-	90.4	85.9
	F1-score (%)	81.6	85	83.6	**53.6**	82.3	-	74.3	**90.8**
	Overall Accuracy = 94.9%				**mIoU =** 75.5% (Classes = 12)				
HPS scan 6	mIoU (%)	**28.3**	42.9	**84.9**	-	53	44.4	51	34.7
	Precision (%)	71.9	**44.9**	**94.1**	-	69	58.7	65.5	70.5
	Recall (%)	**31.8**	**90.5**	89.7	-	69.6	64.6	69.8	40.6
	F1-score (%)	**44.1**	60	**91.9**	-	69.3	61.5	67.6	51.5
	Overall Accuracy = 88%				**mIoU** = 58.7% (Classes = 12)				

From **AIR** scores in Table 5.11, six out of seven safety-related assets in scan 5 achieve an AIR>75%, with four assets achieving a 100% AIR. For scan 6, four assets achieve AIR>75%, with fire switches and extinguishers achieving the highest identification rate of 100%.

Table 5.11: Asset identification rate for HPS scans 5 and 6 (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
HPS scan 5	Total Assets	150	8	5	41	2	6	2
	Correctly Identified	115	7	5	23	2	6	2
	AIR (%)	76.7	97.5	**100**	**56.1**	**100**	**100**	**100**
HPS scan 6	Total Assets	105	6	5	-	18	23	6
	Correctly Identified	54	6	5	-	14	19	2
	AIR (%)	51.4	**100**	**100**	-	77.8	82.6	**33.3**

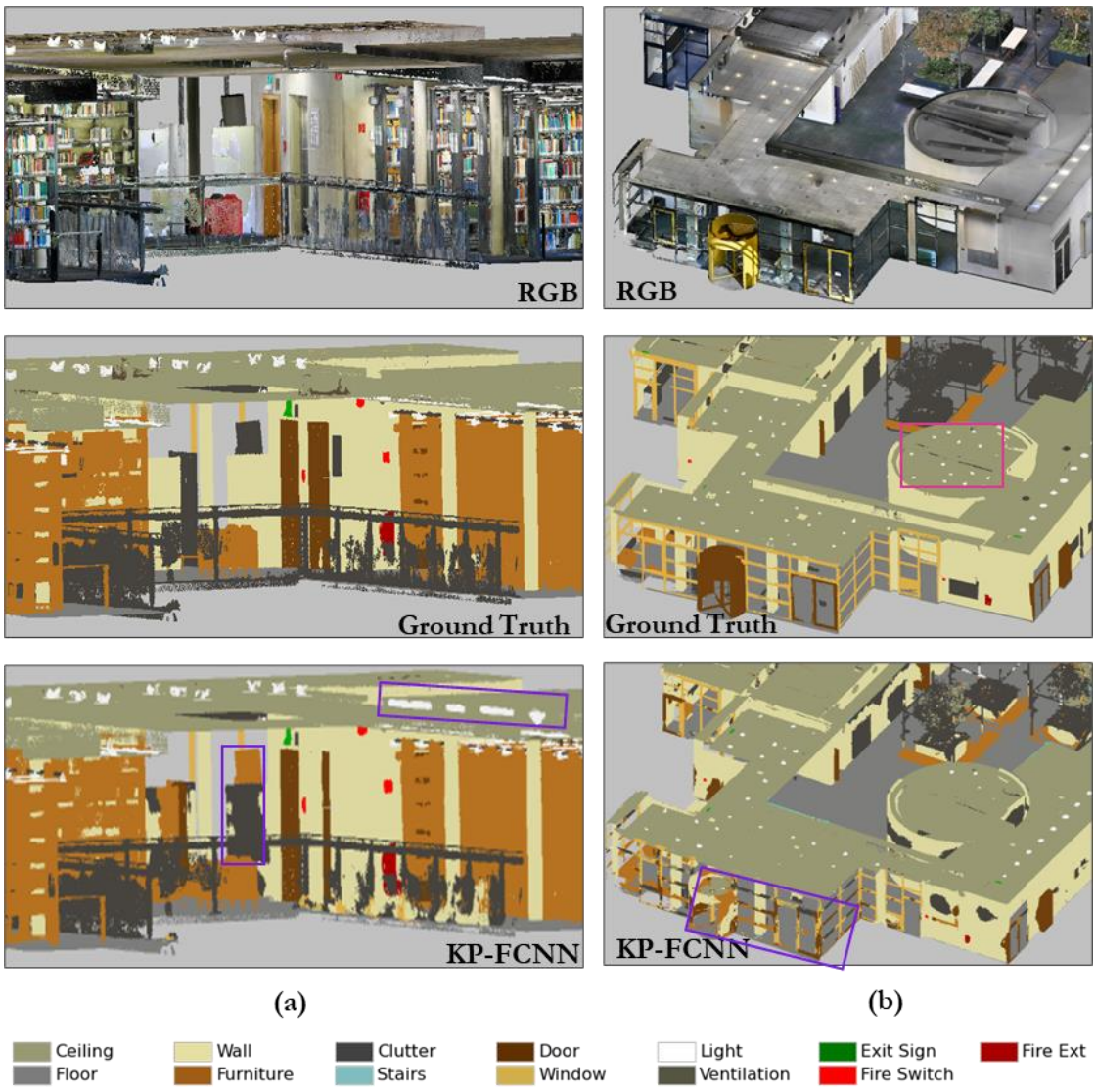


Figure 5.6: Scene segmentation results for (a) scan 5 and (b) scan 6, with bounding boxes in pink for safety-related assets unidentified by the network and misclassifications in purple.

Figure 5.6 shows a part of the scene segmentation results for HPS test scans from Experiment-4, with an RGB image, ground truth, and KP-FCNN results (one-shot method). We have added bounding boxes for safety-related assets unidentified by the network in the ground truth image (pink) and misclassifications in the KP-FCNN image (purple). The permanent structures like walls, ceilings, and floors are well-segmented for both scans. Safety-related assets like exit signs, fire switches, doors, and lights are identified well. Comparing the results of Experiments-2a and 4 from Figure 5.3 and 5.6, Experiment-4 shows fewer misclassifications between classes, showing an improvement in the network's learning. However, for test scan 6, the window class still achieves a poor identification rate in Experiment-4. But the misclassifications of windows into doors are reduced in Figure 5.6 (Experiment-4) compared to Figure 5.3b (Experiment-2a).

5.3. Comparison of Scene Segmentation Methods

We repeated Experiment-1 with the Stage-wise method to use prior knowledge of safety-related assets to segregate an ROI to identify the assets. The train-test split for Experiment 1 is listed in Table 5.12, and the corresponding stage-wise network parameters used are listed in Table 4.4. The Stage-wise method's implementation is described in Section 4.3. In Stage-1, the ceiling, wall, floor, furniture, clutter, door, and window classes were semantically segmented. Then the Stage-1 results were processed to segregate ceiling and wall points based on the prediction labels using the *Pandas* data handling library. In Stage-2, the ceiling, wall, light, ventilation duct, exit sign, fire switch, and fire extinguisher classes were semantically segmented. We modified the network code to carry the original semantic class labels with the Stage-1 prediction results, which were used to modify the safety-related class labels for the Stage-2 input, as shown in Figure 4.6.

Table 5.12: Experiment performed using Stage-wise Method.

Experiment	Train Data	Test Data	Description
1	1, 2, 3, 4, 6	S3DIS Area-5	No similarly-looking buildings in train and test areas

Table 5.13: OA and mIoU for S3DIS Area-5 using Stage-wise method with evaluation metrics for safety-related assets (highest and lowest scores in green and red).

Semantic Class		Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Stairs	Door	Wind.
S3DIS Area - 5	mIoU (%)	41.1	42.5	**1.1**	**77.6**	13.1	-	68.9	56.1
	Precision (%)	44.6	46.5	**2.4**	88.5	40.9	-	82.5	**96.8**
	Recall (%)	83.9	82.5	**2.2**	**86.3**	16.1	-	80	57.2
	F1-score (%)	58.2	59.7	**2.3**	**87.4**	23.1	-	81.6	71.9
		Overall Accuracy = 90.1%			**mIoU** = 58.2% (Classes = 12)				

Table 5.14: Asset identification rate for S3DIS Area-5 with Stage-wise Method (highest and lowest scores in green and red).

Semantic Class	Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
Total Assets	222	32	6	124	14	69	49
Correctly Identified	163	28	0	119	2	69	41
AIR (%)	73.1	87.5	**0**	96	14.3	**100**	83.7

Table 5.13 show that the Stage-wise Method achieves an **OA** of 90.1% and **mIoU** of 58.2%. Compared to the metrics using the One-shot Method for Experiment-1 in Table 5.2, the mIoU increased from 56.6% to 58.2% for the Stage-wise method. Furthermore, compared to the One-shot Method's **AIR** in Table 5.3, the identification rates for all the safety-related assets (except the fire extinguisher) improved with Stage-wise Method, as seen in Table 5.14. For example, the doors had an AIR of 81.4% using the One-shot Method but achieved 100% AIR with Stage-wise Method.

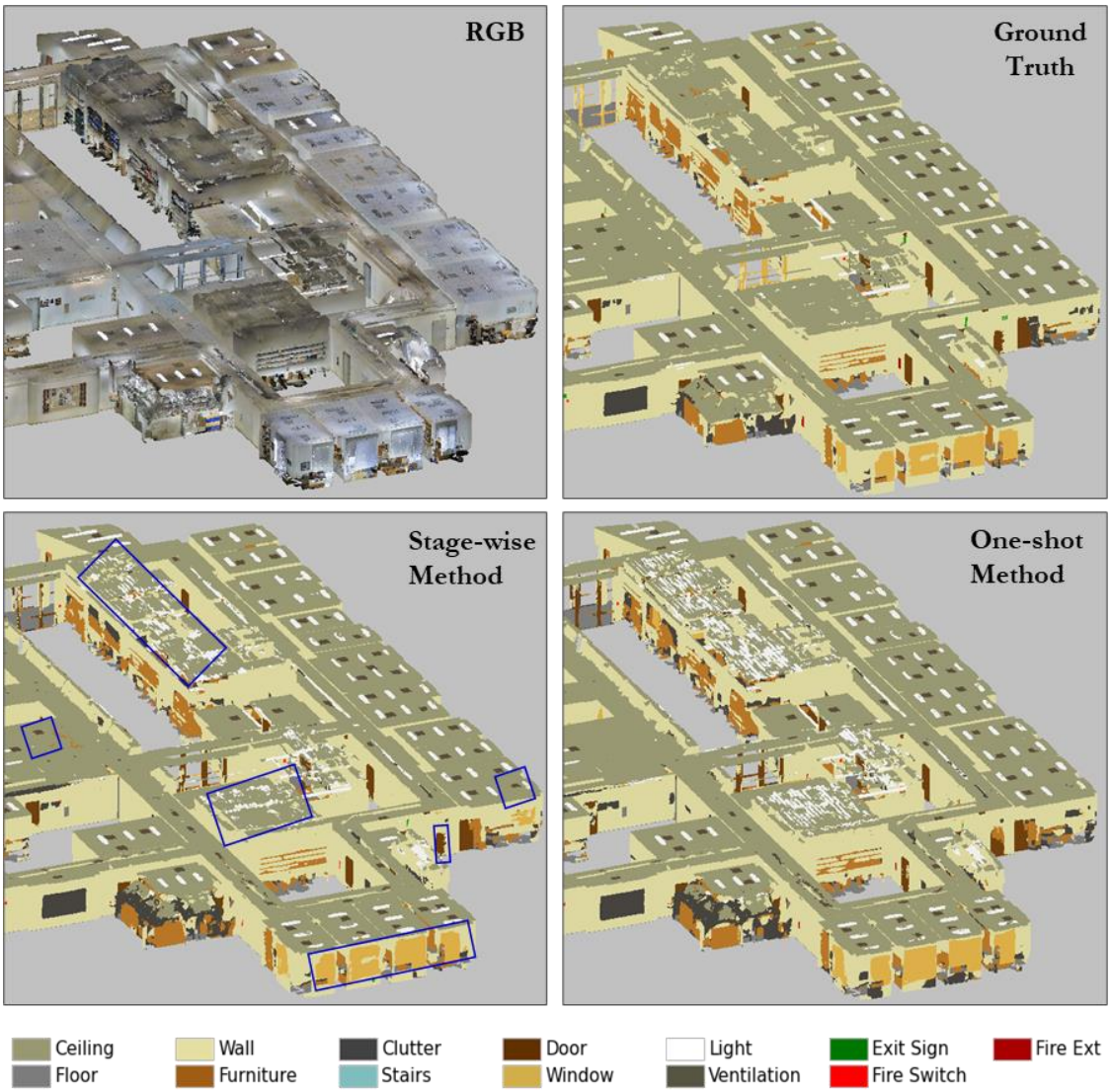


Figure 5.7: Comparing scene segmentation results for S3DIS Area-5 using One-shot and Stage-wise Methods through Experiment-1. Areas with improved segmentation results are shown using dark blue bounding boxes.

Figure 5.7 shows a part of the S3DIS Area-5 with RGB image, ground truth, and KP-FCNN results for both One-shot and Stage-wise Methods. Some improved segmentation results using the Stage-wise Method are shown using dark blue bounding boxes. It can be noticed that there are visibly reduced class mixups between the ceiling and light classes compared to the One-shot Method results. The windows highlighted with the bounding boxes in Stage-wise Method results show that they are precisely identified in terms of the shape of windows with ground truth. Table 5.2 and 5.13 also show that the precision rate for windows increased from 89.8% for the One-shot Method to 96.8% for the Stage-wise Method. Furthermore, Table 1 in Appendix 1 shows an improvement in IoU for eight semantic classes with Stage-wise Method.

6. DISCUSSION

In this chapter, we discuss the research results obtained for safety-related asset identification in Chapter 5. Section 6.1 analyzes the designed methodology's overall performance to identify safety-related assets through Experiments 1-4 in Table 6.1. We elaborate on misclassifications found in the results under Section 6.2 and present the limitations of the designed methodology in Section 6.3.

6.1. Overall Identification of Assets

We have summarized all the experiments performed and their **Asset Identification Rates** (AIR) for safety-related assets from One-shot Method (Experiments 1-4) and Stage-wise Method (Experiment-1*) in Table 6.1. Here, the **One-shot Method** performs scene segmentation for all 13 semantic classes at once, using the entire scene data. **Stage-wise Method** reduces the data volume by separating regions of interest (ceiling and wall structures) and then identifies the assets in those regions.

Table 6.1: Summary of experiments and their AIR in % for safety-related assets. Experiments 1-4 use the One-shot Method; Experiment-1* uses the Stage-wise method.

Dataset Split	Exp.	Train Areas	Test Area	Light	Fire Swi.	Fire Ext.	Vent.	Exit Sign	Door	Wind.
Model generalization: train-test buildings with different feature representations	**1**	S3DIS areas 1-4, 6	S3DIS area 5	71.2	81.3	**0**	**94.4**	7.1	81.4	79.6
	1*	S3DIS areas 1-4, 6	S3DIS area 5	73.1	87.5	**0**	96	14.3	**100**	83.7
	2(a)	S3DIS areas 1-6	HPS scan 5	45.3	**87.5**	**0**	12.2	50	50	50
			HPS scan 6	31.4	66.7	**0**	-	**77.8**	21.7	33.3
Familiar: train-test buildings with similarities in feature representations	**3**	S3DIS areas 1-5	S3DIS area 6	**78.7**	90.9	**100**	96.6	**100**	**100**	93.6
	4	S3DIS areas 1-6 HPS scans 1-4	HPS scan 5	76.7	97.5	**100**	**56.1**	**100**	**100**	**100**
			HPS scan 6	51.4	**100**	**100**	-	77.8	82.6	**33.3**

Figure 6.1: Representations of some safety-related assets in the datasets used in this research: Area-wise S3DIS, iPhone dataset, and HPS Scans (Image source: Armeni et al., 2017; Guzov et al., 2021a).

Figure 6.1 shows some images of safety-related assets in the S3DIS, HPS, and iPhone datasets used for this research. In the following sections, we refer to experiments and AIR from Table 6.1 and asset representations from Figure 6.1 to discuss the overall asset identification performance of the designed methods.

6.1.1. Model Generalization

In this section, we discuss how well the method generalizes on safety-related assets when assessed on new and unfamiliar buildings. Here, unfamiliarity means that the train and test areas for the network are non-overlapping in terms of feature representations. We use Experiment 1 with a test area within the same dataset as train areas (S3DIS area 5) and Experiment 2 with a test area from a different dataset (HPS and iPhone scans), as described in Table 6.1.

- **Experiment-1 (S3DIS Area-5):** Table 6.1 shows that the model generalizes well for safety-related assets, with four out of seven assets achieving AIR>75% with the **One-shot Method**. Further, it effectively identifies assets like fire switches and lights with AIR>70%, which look slightly different than those present in the areas used to train the network. For example, as seen in Figure 6.1, the lights for S3DIS Area-5 are different in geometry and color from other areas of S3DIS used for training. Yet, 71% of lights in Area-5 were correctly identified. Here the robustness of the model can be attributed to two reasons. Firstly, the color annealing property used during training enables the network to learn features occasionally using only geometry without colors (Thomas, 2019). Secondly, the Z information of objects as an input feature to the network encodes the height above ground (with the floor at Z=0), creating a spatial context for representation learning of assets.

  Further, when we repeated Experiment-1* using **Stage-wise Method**, Table 6.1 shows an improvement in AIR for all assets compared to the One-shot Method (except fire extinguishers). With the reduced number of classes in Stage-2 of the Stage-wise Method, the network gets the opportunity to learn more efficiently to abstract the safety-related classes. For this research, KP-FCNN chooses 5000 input point clouds (N) to train 13 semantic classes (C). Then the random picking strategy discussed in Section 2.3 chooses N/C point clouds for each class, here approximately 384 clouds per class (Thomas, 2019). Therefore, as the number of classes decreases in stages 1 and 2 of the Stage-wise method, the number of input spheres per class increases, improving the learning per class.

- **Experiment-2:** We evaluated the **One-shot Method** for generalization with train (S3DIS) and test areas (HPS and iPhone datasets) acquired from different sensors. Based on the dataset properties discussed in Section 3.1, S3DIS and iPhone datasets contain generated point clouds with a consistent density across the scenes (Armeni et al., 2017). In contrast, the scans in the HPS dataset are MLS lidar acquired point clouds with irregular density (Lehtola et al., 2017; Thomas, 2019). Additionally, from Table 3.1, the HPS dataset is vast, and the iPhone dataset is small compared to the S3DIS dataset.

  Despite the varying point densities and sizes of train-test data in Experiment-2a, the model proved compatible and generalized reasonably well by identifying assets like fire switches and exit signs with an average AIR>63% for HPS scans (Tables 6.1). The grid subsampling strategy in KP-FCNN, as described in Section 2.3, enables spatial consistency among train-test datasets making the method invariant to point cloud size and density (Thomas et al., 2019). Comparatively, the One-shot method's performance for Experiment-2a in generalizing and identifying assets is lower than in Experiment-1. Further, we demonstrate from Experiment-2b that the designed method can successfully identify assets using point clouds from a smartphone lidar sensor, which is low-grade in design than the high-grade 3D sensors like depth cameras and MLS lidar. Additionally, from Figure 6.1, we notice that the safety-related assets in iPhone and S3DIS datasets used for training are different. Yet, the designed method

generalizes assets like light, fire switches, and exit signs well, as shown in Figure 5.4. However, a setback is that the iPhone scans do not cover large areas, limiting its usability directly on large buildings.

- **Unidentified Assets:** The designed methods fail to identify some assets when the network is entirely unfamiliar with the representations of assets. In Experiment-1, both the One-shot and Stage-wise methods achieve zero AIR for the fire extinguisher and AIR<15% for the exit sign. From Figure 6.1, S3DIS Area-5 has a visually different fire extinguisher and exit sign compared to the assets in the training area, making them unfamiliar to the network. Similarly, Figure 6.1 show that S3DIS, HPS, and iPhone datasets have different representations of all the safety-related assets. Especially the geometry of fire extinguishers, with the S3DIS having a wall-mounted type and the HPS and iPhone datasets with cylindrical ones (Figure 6.1). In Experiment-2a, the S3DIS-only trained model from the One-shot method is unaware of the cylindrical fire extinguisher in HPS test scans, failing to identify them (AIR=0%). Similarly, for the iPhone scan in Experiment-2b, the fire extinguisher is unidentified, as shown in Figure 5.4(d).

  In some cases, assets are partially identified. For example, in Experiment-1, lights achieve 71% AIR. But most of the unidentified lights for S3DIS Area-5 in Figure 5.2 are round-shaped, and these types of lights are not prominently found in the training areas. As previously explained, the random picking method affects the probability of picking such rarely found representations for the light class, affecting the model's generalization ability on test data. In Experiment-2a, the network is unfamiliar with the lidar point clouds and their feature representations, as the train and test areas belong to different datasets. The assets with high differences in representations like ventilation ducts, doors, and lights result in poor AIR and are only partially identified as the network is unfamiliar with such representations.

6.1.2. Familiar Datasets

In this section, we discuss the **One-shot Method's** performance in segmenting safety-related assets when the network is trained with features familiar to the test area. Here, familiarity means the train and test areas for the network share similarities with feature representations of the assets (geometry and color). However, the test areas are still unseen by the network. Experiments 3 and 4, summarized in Table 6.1, correspond to this scenario. From Table 6.1, it can be noted that the method achieves AIR>75% for most of the assets in these two experiments and performs well than model generalization experiments.

- **Experiment-3 (S3DIS Area-6):** All the safety-related assets in this experiment achieve AIR>75%, with fire extinguishers, exit signs, and doors achieving a 100% AIR. Since training Areas 1 and 3 share similarities with the test Area-6, the network is aware of the asset representations resulting in excellent identification rates. However, lights achieve the least AIR=78% for this experiment, with the hanging type of lights being unidentified. These types of lights are unique to Area-6.

Figure 6.2: Unidentified lights in yellow bounding box for S3DIS Area-6 in Experiment 3; left – RGB image, right – KP-FCNN results.

- **Experiment-4 (HPS scans 5 and 6):** As a part of domain adaptation (a type of transfer learning), we train the network with all S3DIS areas and four HPS scans and assess it on the remaining scans of the HPS dataset. Six assets in test scan 5 and four assets in test scan 6 achieve AIR>75%. Compared to results in Experiment-2a, the AIR scores increased in Experiment-4, shown in Table 6.1. In

Experiment-4, even though the network learns on multiple different representations of each asset type among the chosen train areas (S3DIS+HPS), it abstracts well on both HPS test scans. For example, the network trains on the wall-mounted (three distinct types) and cylindrical (one type) fire extinguishers, as shown in Figure 6.1 from the S3DIS areas and HPS scans. However, the test area has only the cylindrical type of fire extinguisher. Despite the variations within the class, Experiment-4 achieves a 100% AIR score for the fire extinguisher class. These results demonstrate that the network can efficiently learn features from multiple representations of one asset and generalize well.

6.2. Misclassifications

One significant misclassification in S3DIS Area-5 (Experiment-1) is between ceiling and light classes, as shown in Figure 5.7. The ceiling in the yellow bounding box in Figure 6.3 is one of the misclassified regions.

Figure 6.3: The ceiling region misclassified in S3DIS Area-5 in the yellow bounding box.

The highlighted ceiling region is slightly ridged and geometrically different from the planar ceiling prominent throughout the S3DIS dataset. Due to the random picking strategy for training KP-FCNN explained in Section 2.3, planar ceiling feature representations are highly probable to dominate the feature learning for the ceiling class (Thomas, 2019). Hence, the resulting network fails to assign such features to the correct semantic class during testing. Instead, it confuses them with other classes with the closest feature resemblance, causing misclassifications. In our case, the ceiling was misclassified as lights. However, from Figure 5.7, we notice that the ceiling and light class misclassification is considerably reduced with the 2-stage method. Stage-wise separation of classes allows the network to learn and abstract safety-related and non-safety-related classes more effectively, as previously explained in Section 6.1.1.

(a) (b) (c)

Figure 6.4: Examples of windows: a) Experiment-2 train areas; b) S3DIS area 5; c) HPS scan 6 (contains door in yellow).

Another significant misclassification is windows wrongly classified as doors. In **Experiment-1** (S3DIS test Area-5), window types shown in Figure 6.4(b) are misclassified as doors. Figure 6.4(a) shows the commonly found windows in the training area for this experiment, which differ from the misclassified ones shown in Figure 6.4(b) for Area-5. The geometry of windows in Area-5, especially the height, closely resembles the geometry of doors rather than the windows, confusing the network and causing misclassifications. In **Experiments 2(a) and 4**, for the HPS scan-6, window and door classes are misclassified, as shown in Figure 5.3(b) and Figure 5.6(b). The misclassification in Experiment-2a is due to the unfamiliarity with the different window representations in the test scan. However, in Experiment 4, the window class continues to be misclassified as door even though the HPS scans used for training consisted of these types of window representations. The main reason for misclassifications here is inter-class similarities; in our case, the similarity between the window and door classes, as shown in Figure 6.4(c). During the network's training phase, these inter-class similarities produce similar feature representations for both these classes (Bengio et al., 2013; Venkataramanan et al., 2021). The resulting model struggles to differentiate and abstract such classes during testing, resulting in misclassifications.

6.3. Limitations

The AIR evaluation metric designed in Section 4.4.1 for this research proved informative compared to the standard evaluation metrics described in Section 4.4. AIR uses the per-instance mod-IoU metric, which depends on point-wise labels. On evaluating the asset instances using the workflow in Figure 4.11, a few smaller assets were considered incorrect even though identified correctly. For example, the assets displayed in Figure 6.5 are some assets that were erroneously categorized as incorrect but are segmented well; however, they failed to fulfill the mod-IoU criteria defined to calculate AIR. Since this workflow uses the ground truth labels to segregate the assets, errors while defining the asset's ground truth are one reason that causes such erroneous situations. For example, the exit sign in Figure 6.5 has some part of the ceiling as the ground truth of the exit sign. Though the predicted results correctly separate the ceiling and exit sign classes, the workflow for AIR in Figure 4.11 does not recognize such cases.

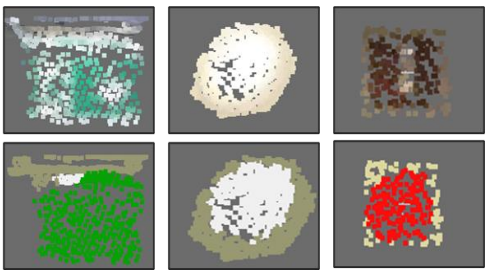


Figure 6.5: Small assets mistakenly categorized as incorrect using AIR workflow. Top – RGB image; Bottom – KP-FCNN results.

Figure 6.6: Objects (2nd and 3rd column) with resemblances with safety-related assets (1st column) in an indoor scene from HPS dataset (a) exit sign (b) fire switch.

Additionally, given a method and dataset, there always exists a level of noise that causes errors in the results. Since safety-related assets play a vital role in emergencies, it is a necessity that the assets are identified accurately. However, indoor scenes from buildings contain various objects, and some of the objects have resemblances to safety-related assets, as shown in Figure 6.6. Due to the resemblances in representations, these objects can be wrongly identified as safety-related assets. The evaluation metrics like precision, recall, and F1 scores were calculated to provide an insight into false positives. However, the current research does not emphasize on falsely identified assets. Therefore, the obtained results require post-processing procedures to recognize and segregate the wrongly identified safety-related assets. In future works to post-process the results, statistical approaches can be utilized to estimate an acceptable point count range for assets to categorize them as correct and incorrect instances. Also, the prior safety-related locations utilized in the Stage-wise method can be employed here to remove incorrect instances.

Another setback occurs when reusing the designed methods for a building with entirely new-looking safety-related assets. Section 6.1.1 discusses how our methods fail to identify new-looking assets like fire extinguishers when the network is unfamiliar with those asset types. New buildings to be evaluated could inevitably have different representations of safety-related assets. For example, the ceiling lights can be square, rectangular, round, or hanging. The fire extinguishers can be squared or cylindrical and wall-mounted or placed on the floor. Such variations within safety-related asset classes from train and test data lead to poor performance and identification rates. One solution for using our method for a building with new-looking assets is that the user can train the network with a part of the dataset and use the resulting network on the remaining dataset to identify the safety-related assets accurately (based on results of domain adaptation).

7. CONCLUSION AND RECOMMENDATIONS

7.1. Conclusion

This research began with the aim of exploring the scope of using the DL scene segmentation approach on 3D point clouds to identify safety-related assets in buildings for asset management. In this context, we designed a workflow to (i) process relevant open-source point cloud datasets for DL, (ii) adapt KP-FCNN with One-shot and Stage-wise methods for scene segmentation, and (iii) estimate asset identification rate (AIR) to evaluate the rate of correctly identified assets. The research focused on commonly found safety-related assets like ceiling lights, exit signs, ventilation ducts, windows, doors, stairs, fire switches, and extinguishers. Since the main objective of this research was to explore the scope of identifying safety-related assets with DL, we primarily used the One-shot method throughout the research. To evaluate the method's adaptability and robustness with data from different 3D sensors, we used three datasets: (i) the benchmark S3DIS dataset generated using a depth camera; (ii) the HPS dataset acquired using an MLS lidar sensor; (iii) a new dataset created using the iPhone lidar sensor in collaboration with CGI Inc. Though these datasets were of different sizes, resolutions, and accuracy, based on quantitative and qualitative results, the method proved invariant and successfully performed scene segmentation to identify safety-related assets.

From the obtained results, the One-shot method demonstrated promising results and successfully identified all the chosen safety-related assets in S3DIS Area-6 with AIR>75%. Despite our limited training datasets, when assessed on new buildings, the One-shot method effectively identified assets with some visual variations compared to those used for training, like fire switches with AIR>80% (S3DIS Area-5 and HPS Scan 5). However, it failed to identify entirely new-looking assets than the ones the network is familiar with, like fire extinguishers achieving a zero AIR for S3DIS Area-5 and HPS scans, which is a setback for reusing the method for a building with different-looking assets. However, the dissimilarities among the assets in buildings are unavoidable. But, through domain adaptation (transfer learning) performed on the HPS scans, we demonstrate that the method effectively updates its learnings by adapting to new representations of fire extinguishers when trained on them, later achieving 100% AIR. Therefore, the user can follow this approach when applying our method to a building with new-looking assets. However, a more efficient solution for future works would be to diversify the varieties of representations for assets in the training datasets.

The main setback in the previous works using DL networks to identify safety assets was that they failed to correctly identify small assets like fire alarms (Hossain et al., 2021). Our research proves that it is possible to identify small-sized assets like fire switches and exit signs using the DL scene segmentation approach (KP-FCNN), achieving a 100% AIR in some cases. Additionally, we establish that one can identify safety-related assets within a building using the point clouds obtained from simpler lidar devices, like iPhone. To the best of our knowledge, the proposed workflow and datasets are the first DL-based methods and labeled datasets effectively implemented for safety-related asset management using 3D point clouds. Lastly, we aim to make our network adaptation and labeled datasets open-source, hoping to help further research using 3D point clouds for safety-related asset management.

7.2. Answers to the Research Questions

1) **How should the chosen DL network be modified for identifying safety-related assets in an indoor scene?**
   KP-FCNN network required suitable choices of network parameters like choosing the input features and sampling strategies for identifying safety-related assets, especially small-sized ones. The chosen parameters with justifications are described in Sections 4.3.1 to 4.3.3. We designed the One-shot and Stage-wise methods (Section 4.3) to utilize KP-FCNN to identify assets.

2) **What is a suitable strategy to handle the class imbalance problem for safety-related assets and to effectively assist their feature learning for DL?**
   A hybrid method with data and algorithm level solutions that modify training data to add more instances of assets and update network weights based on class handles the class imbalance problem in the point cloud data (Section 4.3.3). The random picking method (Section 2.3) and class weights (Section 4.3.3) strategies in the KP-FCNN supports efficient feature learning for safety-related assets.
3) **How can prior knowledge of safety-related assets be used to support their identification?**
   We designed Stage-wise Method (Section 4.3) to utilize the prior knowledge of safety-related asset location to reduce the search space and select a region of interest like ceiling and wall where the chosen assets are present to improve asset identification. Section 5.3 illustrates the improvement in the results of the Stage-wise Method compared to identification rates when scene segmentation is performed for all the 13 semantic classes at once (One-shot Method).
4) **How accurately does the designed methodology identify the safety-related assets?**
   The designed method demonstrates that it can identify all the chosen assets. The S3DIS Area-6 archives AIR>75% for all assets. Section 6.1 discusses the overall asset identification with all datasets.
5) **How well does the method generalize on an unfamiliar point cloud dataset?**
   The method generalizes and identifies assets with visual variations compared to those used for training, like fire switches with AIR>80% (S3DIS Area-5 and HPS Scan 5). However, it fails to identify entirely new-looking assets achieving a zero AIR (fire extinguishers in t S3DIS Area-5 and HPS dataset). Section 5.1 shows the network's generalization results, which later are discussed in Section 6.1.1.
6) **Which evaluation metrics are most suitable for asset management with the obtained results?**
   The AIR metric defined in Section 4.4.1 gives a practical assessment of the designed methodology by providing total counts of correctly identified safety-related assets and the overall identification rate.
7) **What is the acceptable point cloud resolution to identify safety-related assets with the designed method?**
   Table 5.9 shows that point clouds up to point spacing 0.02m are suitable for performing scene segmentation to identify the safety-related assets (Section 5.2.1). Though evaluated on the S3DIS dataset, the designed method proved invariant to point cloud densities and sizes. Hence, this suitable point spacing principle can also be extended to the lidar point clouds of HPS and iPhone scans as well.
8) **Is the designed methodology robust to data generated from different 3D sensor systems?**
   The designed workflow proved compatible with a depth camera, MLS, and smartphone lidar datasets. Experiments 1-4 verified this with datasets from different train-test data split strategies.

7.3. Recommendations

For the future works, the suggested recommendations are as follows:

1. Class imbalances for safety-related assets in 3D point clouds of buildings affect their feature learning in DL networks (discussed in Section 1.3). In this context, it would be interesting to explore attention-based mechanisms that help decide regions of interest to focus on and support effective feature learning by emphasizing useful features (Vaswani et al., 2017).
2. Since the Stage-wise method showed an improvement in asset identification rates, future works can develop on this method by utilizing detection methods in Stage-2 to efficiently search and identify only safety-related assets with a separate workflow for ceiling and wall associated assets. Also, including other assets like smoke detectors, fire sprinklers, and temperature controllers is possible.
3. Instead of manually labeling the point clouds to create more datasets in this line of research, the designed workflow can be used as an initial labeling step and then post-process to correct the incorrectly labeled data using clustering mechanisms like Hossain et al. (2021).
4. Further expanding this research's results into 3D BIMs and databases would be a valuable line of application for real-world modeling and indoor mapping. Users like first responders would benefit from more meaningful 3D visualizations of the location of assets in indoor spaces.

LIST OF REFERENCES

Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S., 1993. An Improved Algorithm for Neural Network Classification of Imbalanced Training Sets. IEEE Trans. Neural Networks 4, 962–969. https://doi.org/10.1109/72.286891

Anjanappa, G., 2022. Using KPConv for indoor scene and safety-related asset segmentation. [WWW Document]. URL https://github.com/HuguesTHOMAS/KPConv-PyTorch/issues/147 (accessed 5.25.22).

Apple, 2020. iPhone 12 [WWW Document]. URL https://www.apple.com/iphone-12/key-features/ (accessed 4.28.22).

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. arXiv. https://doi.org/10.48550/arxiv.1702.01105

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces, in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, NV, USA, pp. 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Balamurugan, A., Zakhor, A., 2019. Online Learning for Indoor Asset Detection, in International Workshop on Machine Learning for Signal Processing, MLSP. IEEE Computer Society, pp. 1–6. https://doi.org/10.1109/MLSP.2019.8918849

Bello, S.A., Yu, S., Wang, C., Adam, J.M., Li, J., 2020. Review: Deep Learning on 3D Point Clouds. Remote Sens. 12, 1729. https://doi.org/10.3390/rs12111729

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Chen, C., 2019. OGC Indoor Mapping and Navigation Pilot Engineering Report. Retrieved from http://www.opengis.net/doc/PER/IndoorPilotER

Díaz-Vilariño, L., Tran, H., Frías, E., Balado, J., Khoshelham, K., 2022. 3D Mapping Of Indoor And Outdoor Environments Using Apple Smart Devices. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLIII-B4-2, 303–308. https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-303-2022

Engel, N., Belagiannis, V., Dietmayer, K., 2021. Point Transformer. IEEE Access 9, 134826–134840. https://doi.org/10.1109/ACCESS.2021.3116304

Esri, 2019. Indoor Mapping [WWW Document]. ArcGIS Indoors. URL https://www.esri.com/arcgis-blog/products/arcgis-indoors/mapping/what-is-indoor-mapping/ (accessed 5.11.22).

Esri, n.d. Vertical Asset Management Program - Raleigh Water Implements [WWW Document]. URL https://www.esri.com/en-us/lg/industry/water/raleigh-water-case-study (accessed 5.11.22).

Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., Wang, F.-Y., 2021. SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation, Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Nashville, TN, USA, pp. 14499–14508. https://doi.org/10.1109/CVPR46437.2021.01427

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning, MIT Press. MIT Press. Retrieved from http://deeplearning.net/

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 43, 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2019. Deep learning for 3D point clouds: A survey. arXiv. https://doi.org/10.1109/tpami.2020.3005434

Guzov, V., Mir, A., Sattler, T., Pons-Moll, G., 2021a. Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-localization in Large Scenes from Body-Mounted Sensors, Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Nashville, TN, USA, pp. 4316–4327. https://doi.org/10.1109/CVPR46437.2021.00430

Guzov, V., Mir, A., Sattler, T., Pons-Moll, G., 2021b. Supplementary - Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-localization in Large Scenes from Body-Mounted Sensors. 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 4316–4327. https://doi.org/10.1109/CVPR46437.2021.00430

Hossain, M., Ma, T., Watson, T., Simmers, B., Khan, J., Jacobs, E., Wang, L., 2021. Building Indoor Point Cloud Datasets with Object Annotation for Public Safety, Proceedings of the 10th International Conference on Smart Cities and Green ICT Systems. SCITEPRESS - Science and Technology Publications, pp. 45–56. https://doi.org/10.5220/0010454400450056

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds, Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA, pp. 11105–11114. https://doi.org/10.1109/CVPR42600.2020.01112

Hugues, T., 2020. Kernel Point Convolution implemented in PyTorch [WWW Document]. GitHub. URL https://github.com/HuguesTHOMAS/KPConv-PyTorch (accessed 5.25.22).

ISO, 2018. Asset Management : Achieving the UN Sustainable Development Goals.

Jaritz, M., Gu, J., Su, H., 2019. Multi-View PointNet for 3D Scene Understanding, International Conference on Computer Vision Workshop (ICCVW). IEEE, Seoul, Korea (South), pp. 3995–4003. https://doi.org/10.1109/ICCVW.2019.00494

Johnson, J.M., Khoshgoftaar, T.M., 2019. Survey on deep learning with class imbalance. J. Big Data 6, 27. https://doi.org/10.1186/s40537-019-0192-5

Kostoeva, R., Upadhyay, R., Sapar, Y., Zakhor, A., 2019. Indoor 3D interactive asset detection using a smartphone, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, pp. 811–817. https://doi.org/10.5194/isprs-archives-XLII-2-W13-811-2019

Landrieu, L., Boussaha, M., 2019. Point Cloud Oversegmentation With Graph-Structured Deep Metric Learning, Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, CA, USA, pp. 7432–7441. https://doi.org/10.1109/CVPR.2019.00762

Landrieu, L., Simonovsky, M., 2018. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs, Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, USA, pp. 4558–4567. https://doi.org/10.1109/CVPR.2018.00479

Lehtola, V. V., Kaartinen, H., Nüchter, A., Kaijaluoto, R., Kukko, A., Litkey, P., Honkavaara, E., Rosnell, T., Vaaja, M.T., Virtanen, J.-P., Kurkela, M., Issaoui, A. El, Zhu, L., Jaakkola, A., Hyyppä, J., 2017. Comparison of the Selected State-Of-The-Art 3D Indoor Scanning and Point Cloud Generation Methods. Remote Sens. 9, 796. https://doi.org/10.3390/RS9080796

Leica Geosystems, n.d. Leica Pegasus: Backpack [WWW Document]. URL https://leica-geosystems.com/en-us/products/mobile-mapping-systems (accessed 4.29.22).

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on X-transformed points, Advances in Neural Information Processing Systems. arXiv, pp. 820–830. https://doi.org/10.48550/arxiv.1801.07791

Liu, F., Li, S., Zhang, L., Zhou, C., Ye, R., Wang, Y., Lu, J., 2017. 3DCNN-DQN-RNN: A Deep Reinforcement Learning Framework for Semantic Parsing of Large-Scale 3D Point Clouds, Proceedings of the International Conference on Computer Vision. IEEE, Venice, Italy, pp. 5679–5688. https://doi.org/10.1109/ICCV.2017.605

Liu, W., Sun, J., Li, W., Hu, T., Wang, P., 2019. Deep Learning on Point Clouds and Its Application: A Survey. Sensors 19, 4188. https://doi.org/10.3390/s19194188

Luetzenburg, G., Kroon, A., Bjørk, A.A., 2021. Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences. Sci. Rep. 11, 1–9. https://doi.org/10.1038/s41598-021-01763-9

Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., Pajarola, R., 2014. Object detection and classification from large-scale cluttered indoor scans. Comput. Graph. Forum 33, 11–21.

https://doi.org/10.1111/CGF.12286

Matterport, n.d. Matterport: 3D models of interior spaces. [WWW Document]. URL https://matterport.com/ (accessed 4.22.22).

NAPSG, 2020. Best Practices Guide to Indoor Mapping, Tracking, and Navigation » NAPSG Foundation. Retrieved from https://www.napsgfoundation.org/resources/open-for-comment-i-axis-best-practices-guide-to-indoor-mapping-tracking-and-navigation/

NavVis, 2022. NavVis: Scalable Reality Capture with NavVis M6 [WWW Document]. URL https://www.navvis.com/m6 (accessed 4.22.22).

Nikoohemat, S., Diakité, A.A., Zlatanova, S., Vosselman, G., 2020. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. Autom. Constr. 113, 103109. https://doi.org/10.1016/j.autcon.2020.103109

Nikoohemat, S., Peter, M., Elberink, S.O., Vosselman, G., 2018. Semantic interpretation of mobile laser scanner point clouds in Indoor Scenes using trajectories. Remote Sens. 10, 1754–1777. https://doi.org/10.3390/rs10111754

Papers With Code, 2022. S3DIS Benchmark (Semantic Segmentation) [WWW Document]. URL https://paperswithcode.com/sota/semantic-segmentation-on-s3dis (accessed 5.5.22).

Qi, C.R., Su, H., Kaichun, M., Guibas, L.J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, USA, pp. 77–85. https://doi.org/10.1109/CVPR.2017.16

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Adv. Neural Inf. Process. Syst. 2017-Decem, 5100–5109.

Robert, D., Vallet, B., Landrieu, L., 2022. Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. arXiv. https://doi.org/10.48550/arxiv.2204.07548

Rukhovich, D., Vorontsova, A., Konushin, A., 2021. FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection. arXiv. https://doi.org/10.48550/arxiv.2112.00322

Soilán, M., Sánchez-Rodríguez, A., Del Río-Barral, P., Perez-Collazo, C., Arias, P., Riveiro, B., 2019. Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring. Infrastructures 2019, Vol. 4, Page 58 4, 58. https://doi.org/10.3390/INFRASTRUCTURES4040058

Song, S., Lichtenberg, S.P., Xiao, J., 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Boston, MA, USA, pp. 567–576. https://doi.org/10.1109/CVPR.2015.7298655

Su, F., Zhu, H., Chen, T., Li, L., Yang, F., Peng, H., Tang, L., Zuo, X., Liang, Y., Ying, S., 2021. An anchor-based graph method for detecting and classifying indoor objects from cluttered 3D point clouds. ISPRS J. Photogramm. Remote Sens. 172, 114–131. https://doi.org/10.1016/J.ISPRSJPRS.2020.12.007

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. SEGCloud: Semantic Segmentation of 3D Point Clouds, International Conference on 3D Vision (3DV). IEEE, Qingdao, China, pp. 537–547. https://doi.org/10.1109/3DV.2017.00067

Teixeira, H., Magalhães, A., Ramalho, A., Pina, M. de F., Gonçalves, H., 2021. Indoor Environments and Geographical Information Systems: A Systematic Literature Review. SAGE Open 11, 215824402110503. https://doi.org/10.1177/21582440211050379

Thomas, H., 2019. Learning new representations for 3D point cloud semantic segmentation. Université Paris sciences et lettres, Paris. Retrieved from https://pastel.archives-ouvertes.fr/tel-02458455

Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds, Proceedings of the International Conference on Computer Vision. Institute of Electrical and Electronics Engineers Inc., pp. 6410–6419. https://doi.org/10.1109/ICCV.2019.00651

United Nations, 2021. Managing Infrastructure Assets for Sustainable Development. United Nations, New

York. https://doi.org/10.18356/9789210051880

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, Advances in Neural Information Processing Systems. Neural information processing systems foundation, pp. 5999–6009. https://doi.org/10.48550/arxiv.1706.03762

Vázquez, F., 2017. Deep Learning made easy with Deep Cognition [WWW Document]. Becom. Hum. Artif. Intell. Mag. URL https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351 (accessed 5.15.22).

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, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Bioinformatics). Springer, Cham, pp. 93–103. https://doi.org/10.1007/978-3-030-87156-7_8

Vosselman, G. (George), Maas, H.-G., 2010. Airborne and terrestrial laser scanning. Whittles Publishing. https://doi.org/10.1080/17538947.2011.553487

Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X., Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. ISPRS J. Photogramm. Remote Sens. 143, 150–166. https://doi.org/10.1016/j.isprsjprs.2018.03.025

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic Graph CNN for Learning on Point Clouds. ACM Trans. Graph. 38, 1–12. https://doi.org/10.1145/3326362

Warsop, T., Singh, S., 2010. A survey of object recognition methods for automatic asset detection in high-definition video. 2010 IEEE 9th Int. Conf. Cybern. Intell. Syst. CIS 2010 1–6. https://doi.org/10.1109/UKRICIS.2010.5898117

Xiao, J., Owens, A., Torralba, A., 2013. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels, International Conference on Computer Vision. IEEE, Sydney, NSW, Australia, pp. 1625–1632. https://doi.org/10.1109/ICCV.2013.458

Xie, X., Lu, Q., Parlikad, A.K., Schooling, J.M., 2020. Digital Twin Enabled Asset Anomaly Detection for Building Facility Management. IFAC-PapersOnLine 53, 380–385. https://doi.org/10.1016/J.IFACOL.2020.11.061

Xu, H., Xu, J., Xu, W., 2019. Survey of 3D modeling using depth cameras. Virtual Real. Intell. Hardw. 1, 483–499. https://doi.org/10.1016/J.VRIH.2019.09.003

Yi, L., Kim, V.G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L., 2016. A scalable active framework for region annotation in 3D shape collections. ACM Trans. Graph. 35, 1–12. https://doi.org/10.1145/2980179.2980238

APPENDIX

Appendix 1 – Scene segmentation results for all thirteen semantic classes

Table 1: Per class IoU scores in % for all thirteen semantic classes. Experiments 1-4 use the One-shot Method; Experiment-1* uses the Stage-wise method.

Exp	Ceil.	Floor	Wall	Furn.	Clut.	Light	Fire Swi.	Fire Ext	Vent.	Exit Sign	Stairs	Door	Wind.
1	86.2	97.8	84.9	76	43.5	40.1	52	0	74	7.6	-	56.9	60.2
2(a) Scan 5	92.7	93.4	66.7	77.5	9.5	56.8	69.8	0	8.1	5	-	7.3	30.9
2(a) Scan 6	90.8	97.5	78.4	54.3	26.3	27	34.2	6.7	-	40.9	0	25.9	16.9
3	94.4	98.3	92.6	85.7	64.3	81.3	62.1	80	84.8	89.2	76.6	89.5	85.8
4 Scan 5	95.3	94.8	82.3	95.8	74.3	68.9	73.8	72.1	36.6	69.9	-	59.2	83.1
4 Scan 6	88	96.7	77.3	65.7	38.3	28.3	42.9	84.9	-	53	4.4	51	34.7
1*	85.1	97.6	88.7	79.4	46.8	41.1	42.5	1.1	77.6	13.1	-	68.9	56.1

Table 2: Per class IoU scores in % for Experiment-3 with subsampled point clouds for the S3DIS dataset using the One-shot Method

Exp 3 With Point spacing (In m)	Ceil.	Floor	Wall	Furn.	Clut.	Light	Fire Swi.	Fire Ext	Vent.	Exit Sign	Stairs	Door	Wind.
0.01	94.4	98.2	93.6	86.1	64.7	73.8	60.5	88.4	86	84.8	79.7	89.3	86.9
0.02	93.6	98.2	93.1	86	65	70.4	65.8	81.5	85.5	85.6	81.7	89.8	85.4

Appendix 2: Confusion Matrices

The columns represent the actual class in the confusion matrix, and the rows represent the predicted class. In the primary matrix with class labels as plot ticks, each cell shows the number and percentage of points of each actual class corresponding to each predicted class. The diagonal cells with black text represent correctly classified points. The other cells with red text represent the wrongly classified points.

The last column shows the number of points for each predicted class in black, the **Precision** rates in green, and the remaining percentages in red. The last row shows the number of points for each class in black, the **Recall** rates in green, and the remaining percentages in red. The bottom-most diagonal cell in dark green shows the total number of points in black, the **Overall Accuracy** percentage in green, and the remaining percentage in red.

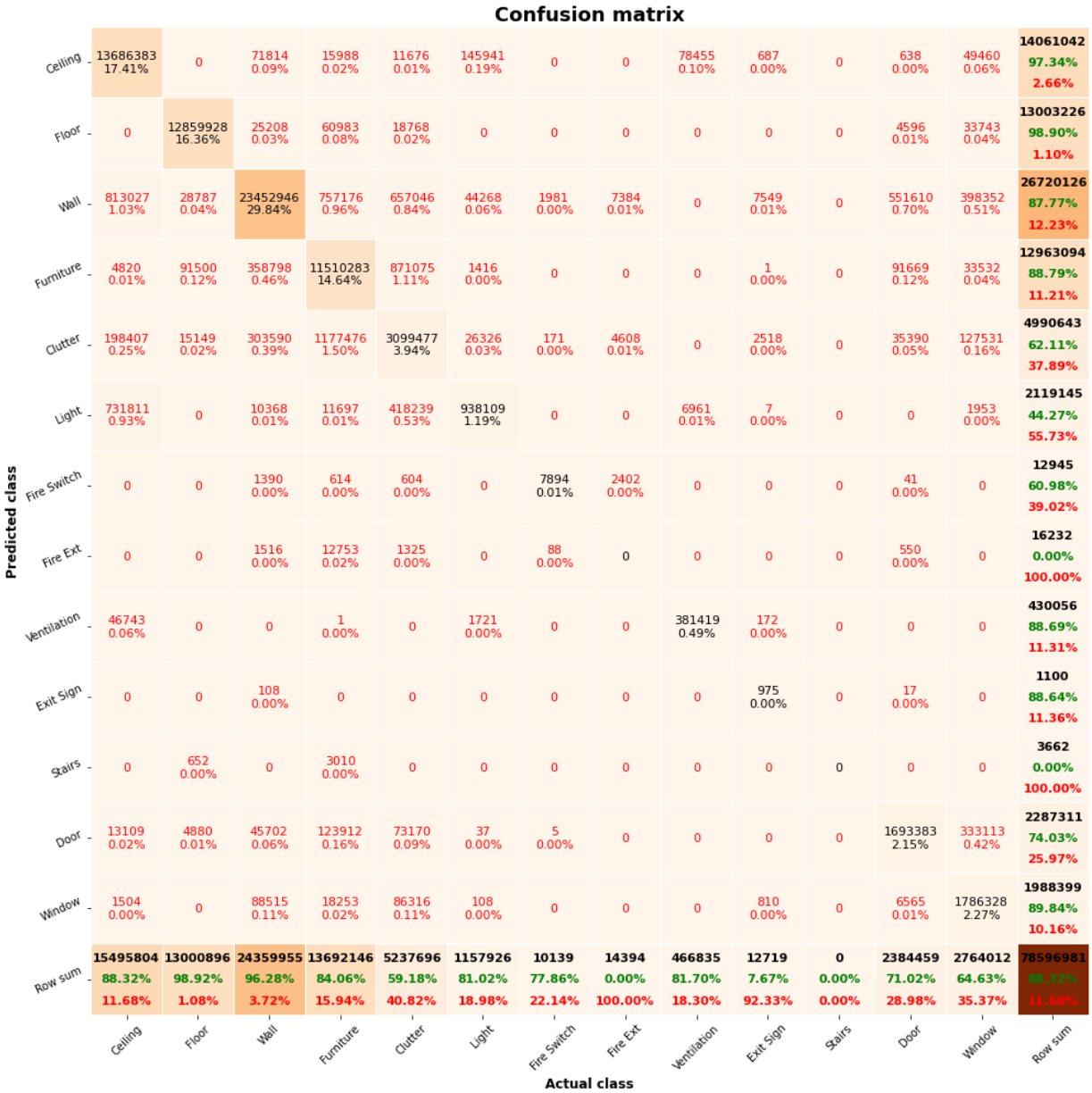


Figure 1: Confusion matrix for Experiment-1 S3DIS test area-5.

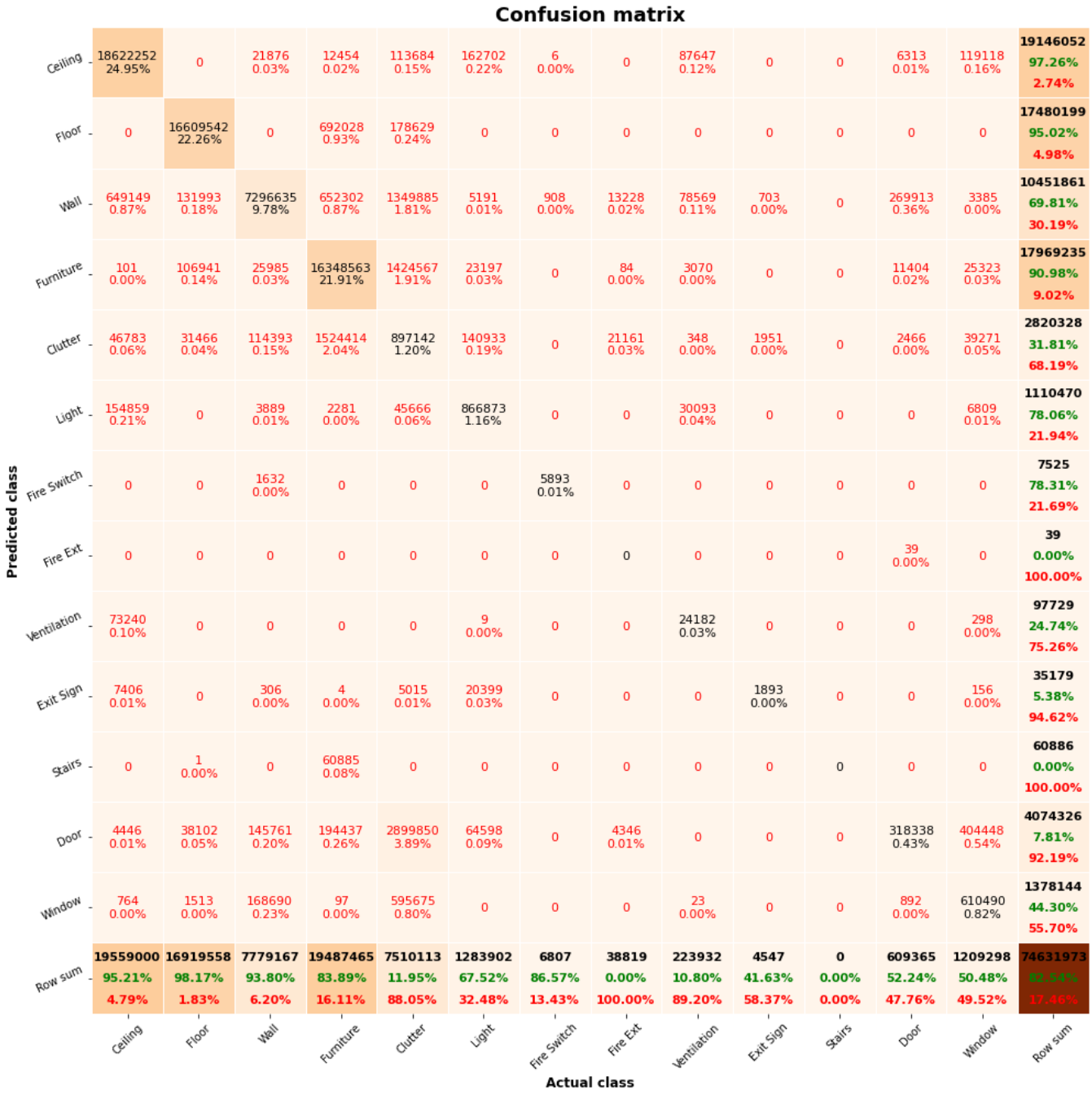


Figure 2: Confusion matrix for Experiment-2a HPS test scan-5.

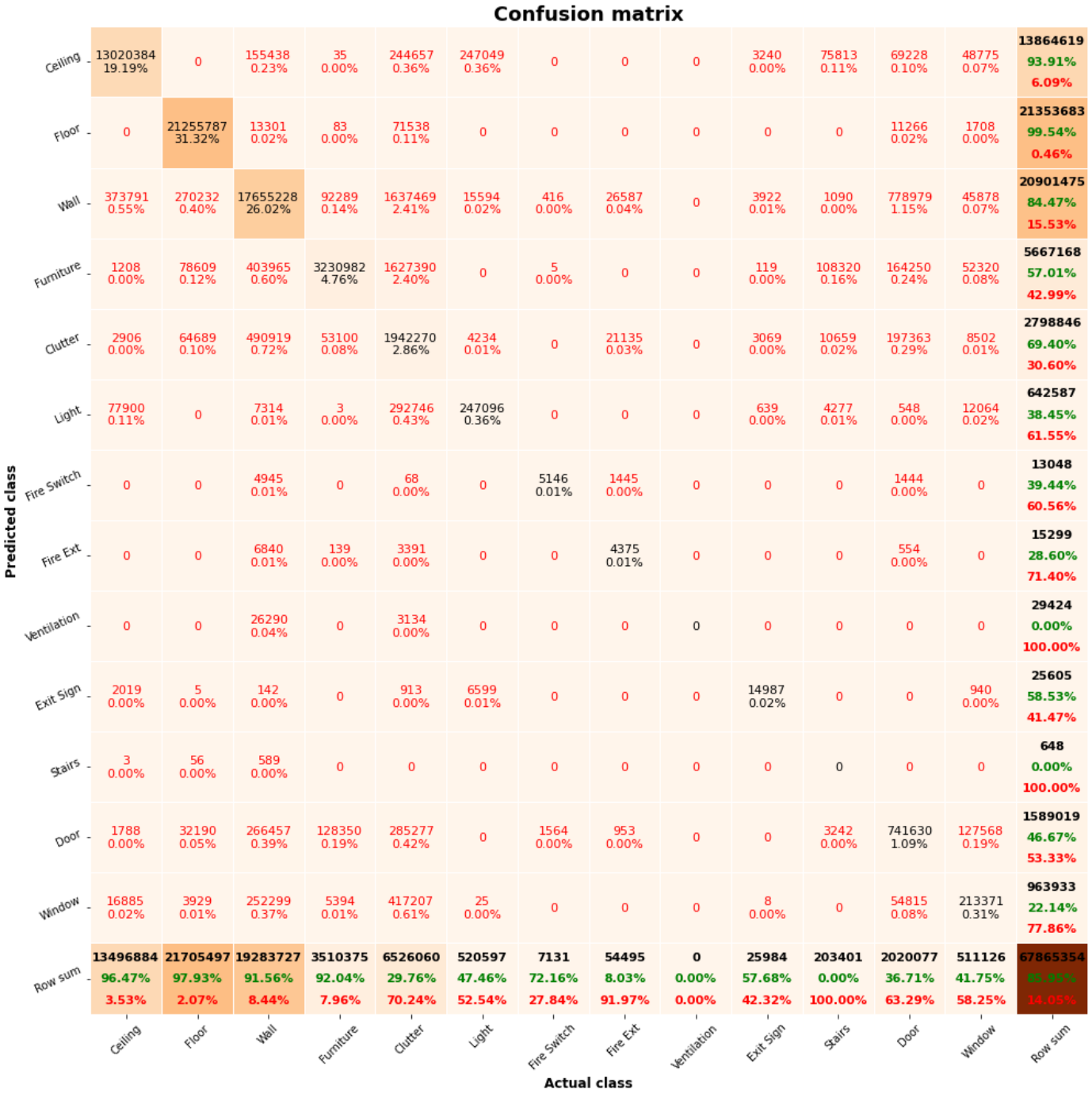


Figure 3: Confusion matrix for Experiment-2(a) HPS test scan 6.

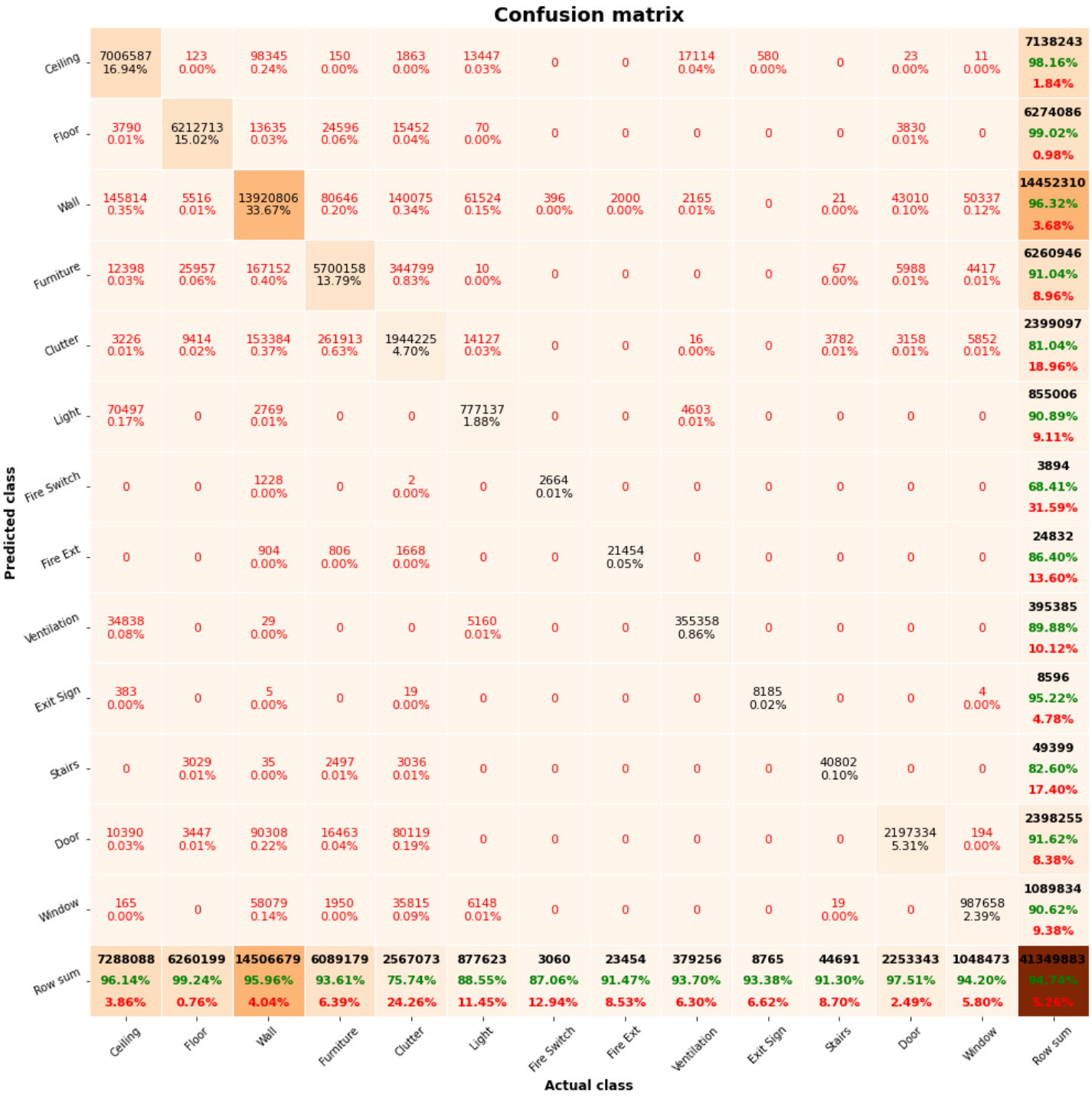


Figure 4: Confusion matrix for Experiment-3 S3DIS test area-6.

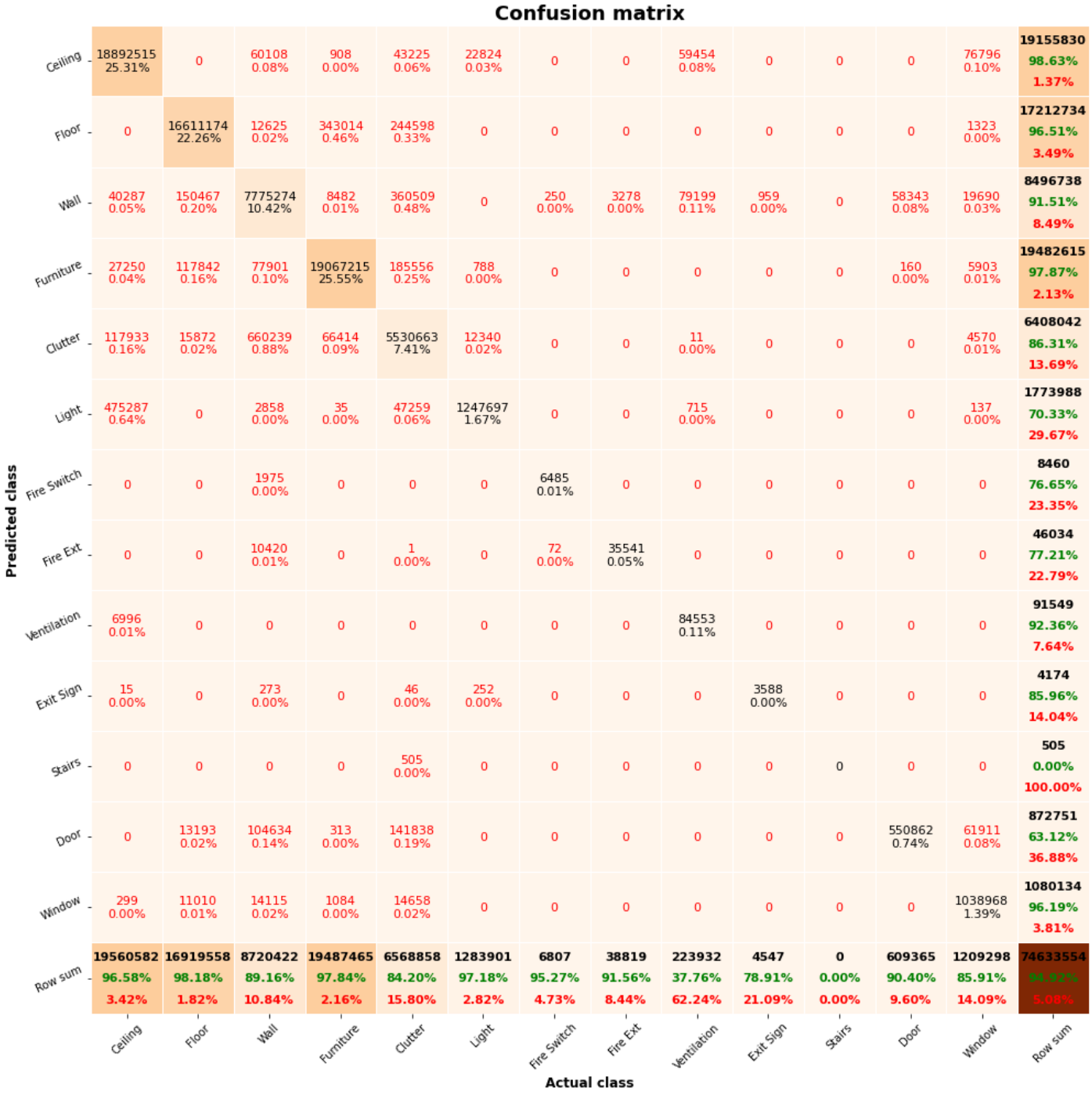


Figure 5: Confusion matrix for Experiment-4 HPS test scan 5.

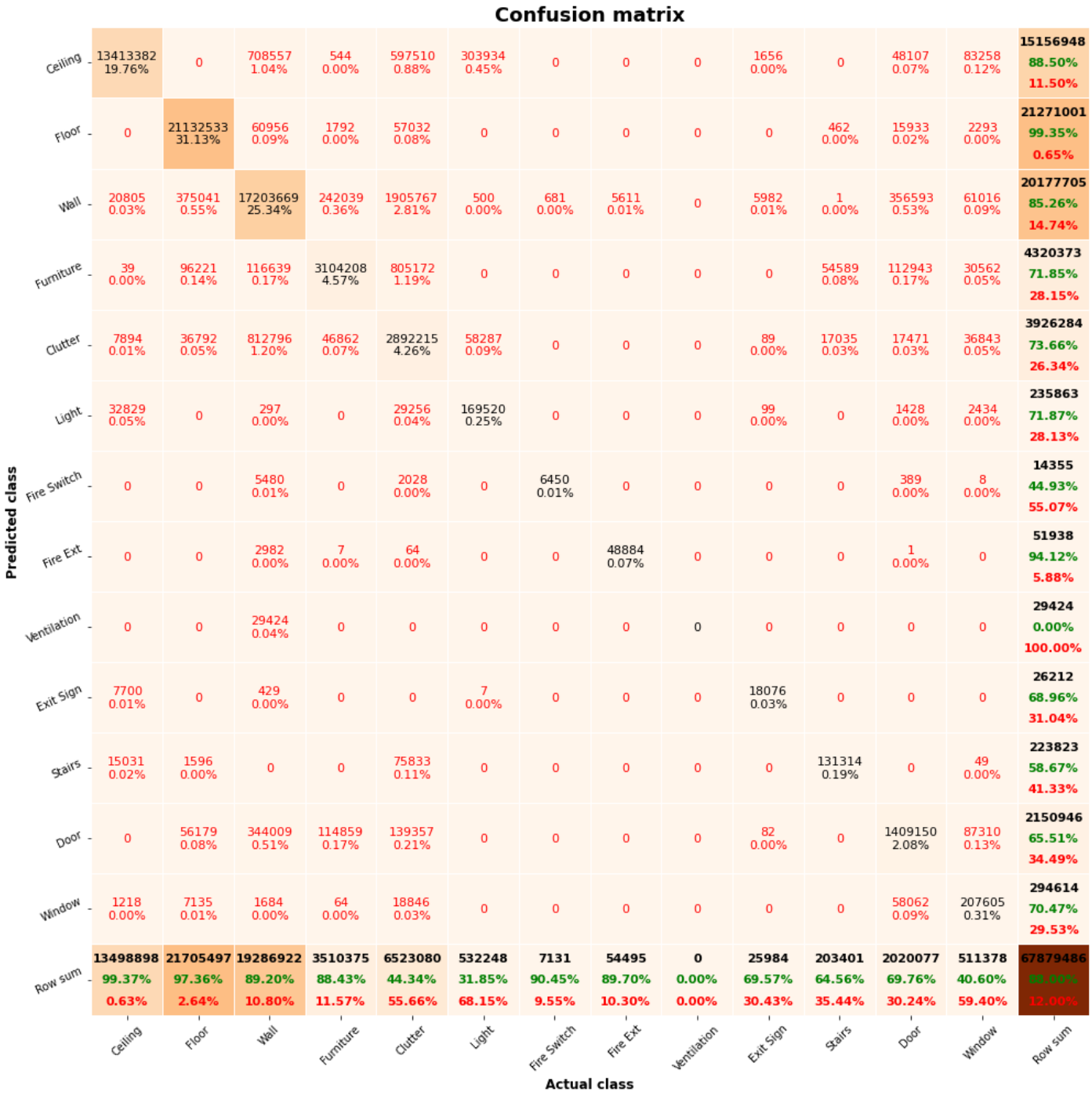


Figure 6: Confusion matrix for Experiment-4 HPS test scan 6.